



HAL
open science

Analyse d'images : Filtrage et segmentation

Philippe Bolon, Jean-Marc Chassery, Jean-Pierre Cocquerez, Didier Demigny,
Christine Graffigne, Annick Montanvert, Sylvie Philipp, Rachid Zéboudj,
Josiane Zerubia, Henri Maître

► **To cite this version:**

Jean-Pierre Cocquerez, Sylvie Philipp (Dir.). Analyse d'images : Filtrage et segmentation. Jean-Pierre Cocquerez; Sylvie Philipp. MASSON, 1995, Enseignement de la physique, 2-225-84923-4. hal-00706168

HAL Id: hal-00706168

<https://hal.science/hal-00706168v1>

Submitted on 23 Jun 2017

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.

Analyse d'images : Filtrage et segmentation

Ouvrage collectif publié avec le soutien du
Groupement de Recherche 134 : Traitement Du Signal et
Images (CNRS)

Coordonné par J-P. COCQUEREZ et S. PHILIPP

Auteurs :

Ph. Bolon, J-M. Chassery, J-P. Cocquerez,
D. Demigny, C. Graffigne, A. Montanvert,
S. Philipp, R. Zéboudj, J. Zerubia

Préface d'Henri Maître

Ouvrage publié avec l'aide du Ministère des affaires étrangères,
direction de la coopération scientifique et technique

AVERTISSEMENT

**Le livre publié en 1995 chez MASSON (EAN13 : 9782225849237)
est épuisé.**

**Cette version pdf est une version élaborée à partir de la version
préliminaire transmise à l'éditeur. La mise en page est légèrement
différente de celle du livre.**

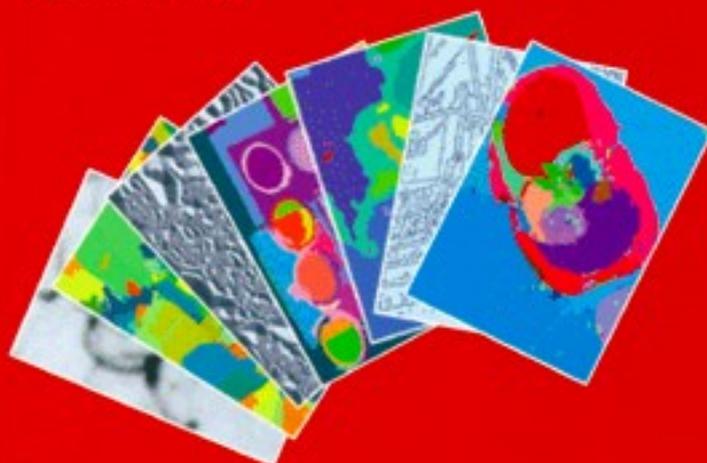
Malheureusement quelques figures de l'annexe C ont été perdues.

ENSEIGNEMENT DE LA PHYSIQUE

Analyse d'images : filtrage et segmentation

Ouvrage collectif coordonné par :
J.-P. COCQUEREZ et S. PHILIPP

Préface de H. MAÎTRE



MASSON 

• **Coordination générale**

Jean-Pierre COCQUEREZ
Sylvie PHILIPP

ETIS-ENSEA
6 avenue du Ponceau
95014 Cergy-Pontoise Cédex
cocquerez@ensea.fr
philipp@ensea.fr

• **Auteurs**

Philippe BOLON
(chapitres IV et X)

LAMII/CESALP
Université de Savoie
BP 406
74016 Annecy

J-M. CHASSERY
(chapitres I, VIII, X, XII,
annexe A)

TIMC-IMAG
Institut Albert Bonniot
Université Joseph Fourier
Domaine de la Merci
38706 La Tronche Cédex

J-P. COCQUEREZ
(chapitres I, III, V, VI, XII,
annexes D, E)

ETIS-ENSEA
6 avenue du Ponceau
95014 Cergy-Pontoise Cédex

D. DEMIGNY
(chapitre V, annexes E, F)

ETIS-ENSEA
6 avenue du Ponceau
95014 Cergy-Pontoise Cédex

C. GRAFFIGNE
(chapitres VII, XI, annexe B)

Université René Descartes
UFR de Mathématiques et d'Informatique
45 rue des Saints Pères
75270 Paris Cédex 06

A. MONTANVERT
(chapitres III, XII, annexe A)

LIP
ENS Lyon
46 allée d'Italie
69364 Lyon Cédex

S. PHILIPP
(chapitres II, III, V, IX, X,
annexes A, C)

ETIS-ENSEA
6 avenue du Ponceau
95014 Cergy-Pontoise Cédex

R. ZEBOU DJ
(chapitres X, XIII)

ETIS-ENSEA
6 avenue du Ponceau
95014 Cergy-Pontoise Cédex

J. ZERUBIA
(chapitres VII, XI, annexe B)

INRIA Sophia-Antipolis
2004 route des lucioles
06560 Valbonne Cédex

• **Contributions**

P. GAUSSI ER ETIS-ENSEA (chapitre VI)

I. MAGNIN Laboratoire CREATIS INSA Lyon (chapitre III)

P. MIGNOT INRIA-Rocquencourt (chapitre X)

F. PRETEUX INT Evry (annexe C)

• Remerciements

Un ouvrage comportant autant d'essais n'a pu se faire sans les conseils et la participation de nombreux collègues de différents laboratoires, doctorants, ingénieurs-système, sans oublier les personnes ayant accepté de relire certaines parties du texte. Nous remercions chaleureusement :

E. Bertin, P. Bertolino, B. Chalmond, D. Coquin, R. Deriche, A. De Cesare, P. Duvaut, Y. Elomary, M. Jordan, C. Mathieu, F. Ployette, R. Karah Fallah, Y. Lechevallier, M. Leclerc, V. Lefevre, J. Levy-Vehel, B. Rabin, A. Raji, JP. Reissmann.

Enfin, tout ce travail aurait été impossible sans le soutien du GDR 134 que nous remercions, avec une mention particulière pour Odile Macchi, sa directrice qui a su nous encourager dans les moments difficiles.

Préface

La segmentation des images est indissociablement attachée à l'histoire du traitement de l'image. Dès les années 60 et les premiers travaux d'analyse, d'amélioration, de compression et de reconnaissance des formes sur des signaux bidimensionnels, apparaissent des contributions à ce chapitre important. Leur objectif est encore mal déclaré, mais on peut déjà discerner dans ces premiers écrits l'espoir de décomposer des scènes plus complexes en des éléments individuellement identifiables, ou plus aisément traitables que l'image entière. Les deux approches « par contours » ou « par régions » sont très tôt en compétition, chacune réclamant des vertus particulières. Les travaux fondamentaux, expérimentaux ou appliqués se succèdent sans arrêt jusqu'à nos jours et constituent aujourd'hui le corpus littéraire le plus important du traitement des images. Au milieu des années 80 déjà, une bibliographie attentive du domaine proposait plus de 1500 références au lecteur, qui se gardait bien de les lire d'ailleurs, car un petit nombre seulement d'entre elles apportait une information véritablement établie et généralisable.

Pourquoi ce foisonnement de travaux ? Bien sûr par l'importance du sujet. L'image reste aujourd'hui encore un sujet difficile, dont l'interprétation est très malaisée, et sa décomposition en sous-problèmes plus simples relève naturellement de la technique scientifique la plus élémentaire. Mais l'intérêt du thème n'explique pas tout, sa difficulté est en grande partie cause de la profusion des travaux qui lui sont dédiés, et cela pour deux raisons : tout d'abord parce que le problème est mal posé, ensuite parce qu'il est généralement trivialement résolu par un observateur humain quelconque. Ce deuxième point est particulièrement irritant : qu'un enfant reconnaisse en un instant son chien dans un mauvais cliché d'amateur, qu'il retrouve le chasseur caché dans un dessin, ou qu'il nomme les uns après les autres tous ses camarades sur la photo de classe, en voilà assez pour vexer tous les laboratoires du monde entier. Et l'on rejoint alors le premier problème : qu'appelons-nous détecter, reconnaître ou même voir ? On le sait bien sûr, mais on ne peut aujourd'hui en donner scientifiquement que des définitions axiomatiques manipulant des êtres abstraits du langage scientifique : des pixels, ou des ensembles de pixels partageant des propriétés ou vérifiant des prédicats, et les objets de notre quête (le chien ou le chasseur) ont bien du mal à nous donner « leurs » propriétés, ou « leurs » prédicats distinctifs.

L'ouvrage qui nous est proposé s'attaque courageusement au problème. C'est un ouvrage collectif aussi, peut-être parce que le nombre fait le courage ! Qui aurait conçu de le mener à bien seul ? Il a pris naissance dans l'une de ces structures du CNRS où il est permis de réfléchir à plusieurs sur un problème difficile, de comparer ses expériences et ses connaissances et d'en faire profiter ainsi la communauté. Merci au GdR 134 Traitement du Signal et Images pour cette œuvre de qualité qui explore sans parti-pris toutes les mémoires des traiteurs d'image et

en exprime les algorithmes les plus variés. Merci aux rédacteurs téméraires qui ont pris le risque de cette aventure malgré les conseils des Cassandre (j'en étais !), merci aux programmeurs obscurs, aux testeurs inconnus, aux expérimentateurs ignorés qui ont élaboré les parcelles de ce savoir.

Cet ouvrage est officiellement consacré, nous le disions tout à l'heure, à la segmentation des images. Cet objet est trop modeste car il traite chemin faisant de la majorité des grands problèmes du traitement des images : de la représentation des signaux, de leur description, des outils théoriques de leur traitement (morphologie mathématique, champs de Markov, graphes d'adjacence, classification et bien d'autres que j'oublie). C'est un traité qui sera précieux à plusieurs niveaux : pour l'étudiant qui y trouvera les bases indispensables à sa formation, pour le chercheur et l'enseignant qui compléteront leur culture et leurs références, pour le thématique en télédétection, en sciences biomédicales ou dans d'autres domaines utilisateurs d'images, comme source d'inspiration pour automatiser ses traitements.

Henri Maître

Professeur à l'ENST Paris

Responsable du département Image

Tables des matières

• Coordination générale	V
• Auteurs	V
• Contributions	V
• Remerciements	VI
PREFACE	VII
NOTATIONS	XVIII
Ensemble	XVIII
Filtrage, convolution, transformée de Fourier	XVIII
Fonction	XVIII
Image	XVIII
Opérateurs morphologiques	XIX
Région	XIX
Signal	XIX
Statistique d'ordre	XIX
Variable aléatoire, champ, probabilité.	XIX
Vecteur	XX
AVANT-PROPOS	XXI
IMAGE ET SEGMENTATION	1
ATTRIBUTS DE REGION	6
1 TEXTURE	7
2 ATTRIBUTS STOCHASTIQUES D'UNE IMAGE	8
3 MOMENTS ET MOYENNES D'ESPACE D'UNE REGION	9
4 MATRICES DE COOCCURRENCE	11
4.1 Matrices de cooccurrence et indices de texture du second ordre	11
4.2 Périodes calculées à partir des cooccurrences	16
4.3 Matrices de cooccurrence généralisées	17
5 LONGUEURS DE PLAGE (OU ISOSEGMENTS)	19
6 SPECTRE	20
6.1 Densité spectrale de puissance	21
6.2 Périodes calculées à partir du spectre	22
7 CONTRASTE ASSOCIE A UNE REGION	23
8 ATTRIBUTS FRACTALS	25
8.1 Éléments de théorie fractale et multifractale	25
8.1.1 Dimension fractale	25
8.1.2 Dimension fractale généralisée	25
8.1.3 Spectre multifractal	26
8.1.4 Lien entre dimension fractale généralisée et spectre multifractal	26
8.1.5 Lacunarité	27
8.2 Calcul d'attributs fractals	27

8.2.1 Calcul de la dimension fractale: la méthode des boîtes	27
8.2.2 Calculs du spectre multifractal	30
8.2.3 Calcul des dimensions fractales généralisées	31
8.2.4 Lacunarité	31
9 ATTRIBUTS SURFACIQUES	32
9.1 Courbures	32
9.2 Quadriques	33
10 ATTRIBUTS GEOMETRIQUES	34
11 ATTRIBUTS INTER-REGIONS	36
12 CONCLUSION	37
<u>STRUCTURES DE DONNEES ET ALGORITHMES ASSOCIES</u>	38
1 INTRODUCTION	38
2 PARTITIONS ELEMENTAIRES D'UNE IMAGE ET STRUCTURES ASSOCIEES	39
2.1 Partitions élémentaires	39
2.1.1 Pavage, maillage d'une image	39
2.1.2 Voisinages et métriques associées	40
2.2 Partitions irrégulières	42
2.2.1 Graphes d'adjacence de régions	42
2.2.2 Diagramme de Voronoï	44
3 STRUCTURES HIERARCHIQUES POUR LA DECOMPOSITION D'UNE IMAGE	47
3.1 Arbre quaternaire ou « <i>quadtree</i> »	47
3.1.1 Principe de base	47
3.1.2 Le <i>quadtree</i> linéaire	48
3.1.3 Discussion	50
3.2 Les pyramides : modèles rigides et généralisations	51
3.2.1 La structure pyramidale	51
3.2.2 Pyramides rigides et pyramides stochastiques adaptatives	51
3.2.3 Discussion	51
4 CODAGE DE CONTOURS	52
4.1 Code de Freeman	52
4.2 Codage de la frontière d'un objet en 8-connexité	54
4.3 Approximations polygonales	55
4.3.1 Approximation par découpage récursif	57
4.3.2 Approximation itérative	59
5 ETIQUETAGE EN COMPOSANTES CONNEXES	61
<u>PRETRAITEMENTS</u>	64
1 MODIFICATION D'HISTOGRAMME	64
1.1 Expansion de dynamique	65
1.2 Egalisation d'histogramme	65
1.3 Spécification d'histogramme	68
2 REDUCTION DE BRUIT	69
2.1 Modèles de bruit d'image	70
2.2 Filtres linéaires stationnaires	71
2.2.1 Cas continu	71
2.2.2 Exemples de filtres réducteurs de bruit	72
2.2.3 Cas discret	74
2.2.4 Problèmes liés à l'implantation des opérateurs	76

2.3 Filtres non-linéaires stationnaires	78
2.3.1 Filtres d'ordre	78
2.3.2 Filtres homomorphiques	84
2.3.3 Filtrage morphologique	85
2.4 Filtres adaptatifs	87
2.4.1 Moyennes adaptatives	87
2.4.2 Filtre de rang adaptatif	88
2.4.3 Moyenne tronquée adaptative	89
2.4.4 Filtrage par fractionnement de la fenêtre d'analyse	93
2.4.5 Remarques	93
2.5 Bruit multiplicatif	94
2.5.1 Algorithme de Lee	95
2.5.2 Filtrage homomorphique	95
2.6 Commentaires	96
3 REHAUSSEMENT DE CONTRASTE	96
3.1 Méthodes linéaires	97
3.1.1 Méthodes basées sur le laplacien	97
3.1.2 Méthodes inverses	100
3.2 Filtrage homomorphique	100
3.3 Filtrage d'ordre adaptatif	101
3.4 Méthodes morphologiques	103
3.5 Observations	105
4 CONCLUSION	105

APPROCHE FRONTIERE: METHODES DERIVATIVES, SURFACIQUES,

<u>MORPHOLOGIQUES</u>	106
1 METHODES DERIVATIVES	107
1.1 Généralités	107
1.2 Principes généraux d'utilisation des méthodes dérivatives	110
1.3 Approches dérivatives appliquées aux images numériques	111
1.4 Opérateurs dérivatifs du premier ordre	112
1.4.1 Opérateurs de Prewitt et de Sobel	112
1.4.2 Opérateurs de gradients directionnels de Kirsh	114
1.4.3 Opérateurs MDIF et NAGDIF	115
1.5 Opérateurs dérivatifs du deuxième ordre	117
1.5.1 Opérateur laplacien sur voisinage réduit	117
1.5.2 Opérateur de Marr et Hildreth	117
1.5.3 Opérateur de Huertas-Médioni	118
1.6 Approches par filtrage optimal	119
1.6.1 Approche de Canny	119
1.6.2 Opérateur monodimensionnel de Deriche	123
1.6.3 Opérateurs bidimensionnels de dérivation et de lissage de Deriche	125
1.6.4 Opérateurs monodimensionnels de Shen	128
1.6.5 Opérateurs bidimensionnels de Shen	131
1.7 Comparaison des performances théoriques de quelques opérateurs de dérivation	132
1.7.1 Critères continus de comparaison	134
1.7.2 Critères discrets de comparaison	135
2 MODELE SURFACIQUE	139
2.1 Opérateur de Hueckel	140
2.2 Opérateur de Haralick	140
2.2.1 Rappel sur l'approximation polynômiale	140
2.2.2 Polynômes monodimensionnels discrets de Tchebycheff	141

<u>2.2.3 Approximation polynômiale bidimensionnelle</u>	141
<u>2.2.4 Exemple de calcul de masques pour l'opérateur d'Haralick</u>	142
<u>2.2.5 Estimation de la direction du gradient</u>	143
<u>3 MORPHOLOGIE MATHÉMATIQUE</u>	144
<u>3.1 Gradient morphologique</u>	144
<u>3.2 Ligne de partage des eaux</u>	145
<u>4 LOCALISATION DES CONTOURS ET SEUILLAGE</u>	147
<u>4.1 Extraction des maximums locaux de la norme du gradient</u>	147
<u>4.2 Seuillage par hystérésis</u>	148
<u>4.3 Suivi de ligne de crête</u>	148
<u>5 CONCLUSION</u>	150
<u>FERMETURE DE CONTOURS, CONTOURS VIRTUELS</u>.....	151
<u>1 FERMETURE DES CONTOURS</u>	151
<u>1.1 Recherche du meilleur chemin entre 2 extrémités</u>	151
<u>1.2 Recherche du meilleur chemin à partir d'une extrémité</u>	153
<u>1.2.1 Identification des extrémités</u>	153
<u>1.2.2 Algorithme de fermeture</u>	154
<u>2 APPROCHE NEUROMIMÉTIQUE DE LA VISION</u>	156
<u>2.1 Modèles de réseaux de neurones pour la vision de bas-niveau</u>	156
<u>2.1.1 Système d'analyse d'images par réseau de neurones</u>	156
<u>2.1.2 Masques orientés pour l'extraction de contours</u>	158
<u>2.1.3 Premier niveau de compétition</u>	160
<u>2.1.4 Deuxième niveau de compétition</u>	162
<u>2.1.5 Coopération orientée</u>	165
<u>2.1.6 Rétroaction</u>	166
<u>2.2 Réalisation d'un système d'extraction et de fermeture</u>	<u>des contours avec</u>
<u>rétroaction</u>	167
<u>2.2.1 Description de l'implantation</u>	167
<u>2.2.2 Conclusion</u>	169
<u>3 EXTRACTION DE POINTS CARACTÉRISTIQUES</u>	169
<u>3.1 Utilisation de la diffusion pour l'extraction</u>	<u>de points caractéristiques</u>
<u>3.2 Détection des points caractéristiques grâce</u>	<u>à des cellules Center-Off</u>
<u>3.2 Détection des points caractéristiques grâce</u>	173
<u>4 CONCLUSION</u>	174
<u>APPROCHE FRONTIÈRE : MÉTHODES MARKOVIENNES</u>.....	175
<u>1 INTRODUCTION</u>	175
<u>2 RESTAURATION ET DÉTECTION DE BORDS</u>	176
<u>2.1 Premier modèle</u>	177
<u>2.2 Deuxième modèle</u>	178
<u>2.3 Troisième modèle</u>	180
<u>3 DEUX ALGORITHMES DETERMINISTES</u>	180
<u>3.1 GNC (Graduated Non Convexity)</u>	180
<u>3.2 Recuit par champs moyens (MFA)</u>	184

<u>APPROCHE FRONTIERE : METHODES VARIATIONNELLES</u>	189
<u>1 DETECTION DE CONTOURS : APPROCHES GLOBALES</u>	189
<u>2 MODELE DE CONTOURS ACTIFS (OU SNAKES)</u>	191
<u>2.1 Présentation du modèle du contour actif</u>	192
<u>2.2 Résolution par une approche de type modèle stationnaire</u>	195
<u>2.3 Aspects numériques</u>	196
<u>2.4 Modèle évolutif et formulation variationnelle</u>	199
<u>2.5 Approche numérique et évolution de la courbe</u>	201
<u>3 MODELE DISCRET DE LA BULLE</u>	202
<u>4 CONCLUSION</u>	206
<u>COMPARAISON DES EXTRACTEURS DE CONTOUR</u>	208
<u>1 DESCRIPTION DES METHODES D'EXTRACTION DE CONTOUR TESTEES</u>	208
<u>2 COMPARAISON SUR IMAGES DE SYNTHESE</u>	210
<u>2.1 Génération des images de test</u>	210
<u>2.2 Définitions des critères de comparaison</u>	212
<u>2.3 Paramètres de réglage des différents opérateurs</u>	215
<u>2.4 Analyse des résultats</u>	217
<u>3 RESULTATS SUR LES IMAGES REELLES</u>	225
<u>3.1 Description des images</u>	225
<u>3.2 Paramètres de réglage des différents opérateurs</u>	226
<u>3.3 Analyse des résultats</u>	227
<u>3.4 Conclusion</u>	229
<u>METHODES DE CLASSIFICATION</u>	236
<u>1 METHODES MONODIMENSIONNELLES</u>	237
<u>1.1 Seuillage global</u>	238
<u>1.1.1 Recherche des seuils inter-modes</u>	238
<u>1.1.2 Recherche des modes</u>	238
<u>1.2 Méthodes locales itératives et adaptatives</u>	244
<u>1.2.1 Méthode itérative de classification bayésienne</u>	244
<u>1.2.2 Méthode de relaxation-gradient</u>	247
<u>1.3 Méthodes de seuillage dynamique intégrant des informations de localisation</u>	249
<u>1.3.1 Méthode de Nakagawa</u>	249
<u>1.3.2 Utilisation du laplacien</u>	253
<u>1.3.3 Méthode de Hertz : coopération contour/multiseuillage</u>	254
<u>1.4 Discussion</u>	255
<u>1.5 Conclusion</u>	255
<u>2 METHODES MULTIDIMENSIONNELLES</u>	256
<u>2.1 Classification non supervisée</u>	257
<u>2.1.1 Nuées dynamiques</u>	258
<u>2.1.2 Classification ascendante hiérarchique</u>	260
<u>2.2 Classification supervisée</u>	263
<u>2.2.1 Construction d'une fonction discriminante</u>	263
<u>2.2.2 Construction d'un sous-espace discriminant</u>	268
<u>2.3 Méthodes de sélection des attributs</u>	273
<u>2.3.1 Définition des critères de sélection</u>	273
<u>2.3.2 Procédure de sélection des attributs</u>	275
<u>2.4 Procédure de validation</u>	275

<u>3 CONCLUSION</u>	276
<u>APPROCHE REGION : METHODES MARKOVIENNES</u>	277
<u>1 REGULARISATION</u>	277
<u>1.1 Introduction</u>	277
<u>1.2 Un algorithme supervisé de segmentation d'images texturées</u>	278
<u>1.2.1 Modèle de texture</u>	278
<u>1.2.2 Loi jointe</u>	279
<u>1.2.3 Estimateur du MAP</u>	281
<u>1.3 Segmentation et détection de contours sur des images texturées</u>	281
<u>1.3.1 Modèle de contours</u>	281
<u>1.3.2 Modèle de segmentation</u>	283
<u>2 RELAXATION</u>	285
<u>2.1 Introduction</u>	285
<u>2.2 Champ de Markov caché et segmentation non-supervisée</u>	285
<u>2.3 Un algorithme non supervisé de segmentation</u>	288
<u>2.3.1 Formation automatique de germes de base</u>	289
<u>2.3.2 Croissance de régions de bas niveau</u>	290
<u>2.3.3 Croissance de régions de haut niveau</u>	290
<u>2.4 Un algorithme semi-supervisé de segmentation d'images texturées</u>	292
<u>2.4.1 Modèle</u>	292
<u>2.4.2 Définition de Ω</u>	293
<u>2.4.3 Energie U_1</u>	295
<u>2.4.4 Système de voisinages</u>	295
<u>2.4.5 Caractéristiques de textures et seuils</u>	296
<u>2.5 Un algorithme supervisé de segmentation d'images non-texturées</u>	297
<u>2.5.1 Modèle</u>	297
<u>2.5.2 Choix du paramètre β</u>	298
<u>2.5.3 Fonction énergie</u>	298
<u>2.5.4 Relaxation</u>	298
<u>APPROCHE REGION : METHODES STRUCTURALES</u>	300
<u>1 INTRODUCTION</u>	300
<u>2 AGREGATION DE PIXELS</u>	301
<u>2.1 Algorithme général</u>	301
<u>2.2 Détail de la croissance de régions</u>	302
<u>2.3 Convergence</u>	304
<u>2.4 Commentaires</u>	304
<u>3 SEGMENTATION PAR CORRELATION D'HISTOGRAMMES LOCAUX</u>	304
<u>3.1 Généralités</u>	304
<u>3.2 Principe</u>	307
<u>3.3 Croissance de régions</u>	308
<u>3.3.1 La relaxation</u>	308
<u>3.3.2 La propagation</u>	311
<u>3.3.3 Choix des seuils</u>	314
<u>4 DIVISION ET FUSION GUIDEES PAR LES STRUCTURES</u>	315
<u>4.1 Présentation générale</u>	315
<u>4.1.1 Algorithme général de segmentation par division et fusion</u>	315
<u>4.1.2 Remarques sur la convergence et la complexité</u>	316
<u>4.2 Fusion de régions dans un graphe</u>	317
<u>4.3 Méthode de fusion multicritère</u>	319

4.4 Partitionnement de Voronoï	320
4.5 Arbre quaternaire (« quadtree »)	322
4.6 Approches pyramidales	326
4.6.1 Pyramides de graphes	327
4.6.2 Compléments sur la stratégie de fusion	330
<u>COMPARAISON DES RESULTATS : APPROCHE REGION</u>	331
1 <u>DEFINITION DES CRITERES DE COMPARAISON</u>	331
1.1 <u>Une distance entre deux images de régions</u>	331
1.1.1 <u>Mesure de Vinet</u>	332
1.1.2 <u>Une distance entre deux images de régions</u>	333
1.2 <u>Utilisation du contraste inter-région</u>	334
2 <u>COMPARAISON SUR DES IMAGES DE SYNTHESE</u>	334
2.1 <u>Fabrication des images de synthèse</u>	334
2.2 <u>Méthodes testées</u>	336
2.3 <u>Paramètres de réglage des méthodes</u>	337
2.4 <u>Analyse des résultats</u>	338
2.4.1 <u>Images bruitées (analyse quantitative)</u>	344
2.4.2 <u>Images bruitées (analyse visuelle)</u>	346
2.4.3 <u>Images de texture</u>	347
3 <u>COMPARAISON SUR DES IMAGES REELLES</u>	347
3.1 <u>Méthodes testées</u>	347
3.2 <u>Paramètres des méthodes</u>	347
3.3 <u>Résultats sur les images de la banque du GDR 134</u>	350
3.4 <u>Comparaison</u>	362
3.5 <u>Conclusion</u>	363
<u>EN GUISE DE CONCLUSION</u>	364
<u>ELEMENTS DE THEORIE DES GRAPHS DIAGRAMME DE VORONOÏ</u>	365
1 <u>DEFINITIONS GENERALES DE LA THEORIE DES GRAPHS</u>	365
1.1 <u>Notions de graphe</u>	365
1.2 <u>Chaîne, connexité</u>	366
1.3 <u>Graphe valué</u>	367
2 <u>DIFFERENTES APPROCHES POUR LA CONSTRUCTION</u> <u>DU DIAGRAMME DE</u>	367
<u>VORONOÏ</u>	
2.1 <u>Approches algorithmiques et distance euclidienne</u>	367
2.1.1 <u>Approche Divide and Conquer</u>	368
2.1.2 <u>Méthode incrémentale</u>	368
2.2 <u>Approches algorithmiques et distance discrète</u>	370
<u>CHAMPS MARKOVIENS ET SEGMENTATION</u>	372
1 <u>DEFINITIONS</u>	372
1.1 <u>Champs markoviens</u>	372
1.2 <u>Mesure de Gibbs et champs markoviens</u>	374
1.3 <u>Expression des probabilités conditionnelles</u>	375

2 APPROCHE MARKOVIENNE DE LA SEGMENTATION	376
3 ESTIMATION DE LA CONFIGURATION λ	377
4 ESTIMATION DES PARAMETRES	378
5 ALGORITHMES DE SIMULATION	379
5.1 Echantillonneur de Gibbs	380
5.2 Algorithme de Métropolis	380
6 ALGORITHMES D'OPTIMISATION	381
6.1 Algorithme ICM (Iterated Conditional Modes)	381
6.2 Recuit stochastique simulé	382
ELEMENTS DE MORPHOLOGIE MATHEMATIQUE	384
1 MORPHOLOGIE MATHEMATIQUE APPLIQUEE AUX IMAGES BINAIRES	384
1.1 Erosion et dilatation morphologiques	385
1.2 Ouverture et fermeture morphologiques	386
1.3 Amincissement et épaissement	387
2 MORPHOLOGIE MATHEMATIQUE APPLIQUEE AUX IMAGES EN NIVEAUX	390
DE GRIS	
2.1 Erosion morphologique	390
2.2 Dilatation morphologique	393
2.3 Ouverture, fermeture morphologiques	395
2.4 Amincissement et épaissement	398
RAPPELS DE TRAITEMENT DU SIGNAL	399
1 TRANSFORMEE DE FOURIER	399
1.1 Définition	399
1.2 Translation	399
1.3 Dérivation	400
1.4 Théorème de convolution	400
1.5 Transformée de Fourier d'une distribution	400
1.5.1 Transformée de Fourier de la distribution de Dirac	401
1.5.2 Transformée de Fourier d'un signal sinusoïdal	401
1.6 Développement en série de Fourier de la fonction peigne	401
2 ECHANTILLONNAGE	402
2.1 Formule de Poisson	402
2.2 Spectre d'une fonction échantillonnée	403
2.3 Théorème de l'échantillonnage, formule de Shannon	403
3 PROCESSUS ALEATOIRES	404
3.1 Statistiques des processus aléatoires	405
3.2 Relation d'incertitude	405
3.3 Processus stationnaires	406
4 FILTRE	406
4.1 Filtrage d'un signal aléatoire	407
4.2 Filtre adapté	407
4.3 Filtre dérivateur	409
IMPLANTATION RECURSIVE DES OPERATEURS DE DERICHE ET SHEN	411
1 RAPPELS SUR LA TRANSFORMEE EN Z	411
1.1 Définition	411

1.2 Quelques propriétés de la transformée en Z	411
2 IMPLANTATION DES OPERATEURS DE DERICHE ET SHEN	412
2.1 Implantation des opérateurs monodimensionnels de Deriche	412
2.1.1 Dérivation	412
2.1.2 Lissage	414
2.2 Implantation de l'opérateur gradient bidimensionnel de Deriche	415
2.3 Implantation des opérateurs monodimensionnels de Shen-Castan	415
2.3.1 Opérateur de lissage	415
2.3.2 Opérateur de dérivation	416
2.4 Implantation des opérateurs bidimensionnels de Shen-Castan	417
2.4.1 Lissage	417
2.4.2 Opérateurs de dérivation première	417
<u>DISCRETISATION DES CRITERES DE CANNY</u>	419
1 CRITERE DE BONNE DETECTION	420
2 PROBABILITE DE LOCALISATION EXACTE DU MAXIMUM	420
3 CRITERE DE FAIBLE MULTIPLICITE DES REPONSES (CD3)	422
4 CRITERE DE BONNE LOCALISATION	423
<u>BIBLIOGRAPHIE</u>	427
<u>Sigles des revues ou congrès les plus couramment cités</u>	427
<u>Chapitre II</u>	427
<u>Chapitre III</u>	430
<u>Chapitre IV</u>	431
<u>Chapitre V</u>	434
<u>Chapitre VI</u>	435
<u>Chapitre VII</u>	437
<u>Chapitre VIII</u>	438
<u>Chapitre IX</u>	440
<u>Chapitre X</u>	440
<u>Chapitre XI</u>	442
<u>Chapitre XII</u>	444
<u>Chapitre XIII</u>	445
<u>Annexe A</u>	446
<u>Annexe B</u>	446
<u>Annexe C</u>	447

Notations

Les notations sont présentées par rubriques classées dans l'ordre alphabétique.

Ensemble

Un ensemble est désigné par une lettre majuscule en caractère gras italique.

Exemple :

Un ensemble \mathcal{S} de sites s

Filtrage, convolution, transformée de Fourier

$$\text{TF } A(u) = \hat{A}(u) = \int_{-\infty}^{\infty} A(x) \exp(-2j\pi ux) dx$$

$$\text{TF}\{A * h(x)\} = \hat{A}(u) \hat{h}(u)$$

Fonction

arc cosinus : acos

exponentielle : exp

logarithme népérien : ln

tangente: tan

Image

Première lettre de l'alphabet en majuscule, caractère gras.

Exemples :

$\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$

\mathbf{A} peut être considérée comme :

- un vecteur bidimensionnel
- un champ aléatoire

Dimensions d'une image, site, indice de ligne :

L'image est définie sur une maille dont les sommets constituent un ensemble \mathcal{S} de sites s :

$\mathcal{S} = \{s = (i,j); 1 \leq i \leq M, 1 \leq j \leq N\}$ pour une maille carrée de taille $M \times N$. Le premier indice est l'indice de ligne.

Le nombre de niveaux d'une image monochrome est noté L .

A_s représente la grandeur associée au site s . Cette notation est équivalente à $A[i,j]$.

Opérateurs morphologiques

Erreur ! : érosion.

\oplus : dilatation.

Région

Une région est un ensemble de sites. La région i est donc notée R_i , elle comprend $\text{card}(R_i)$ sites.

Signal

Exemple de signaux continus et discrets:

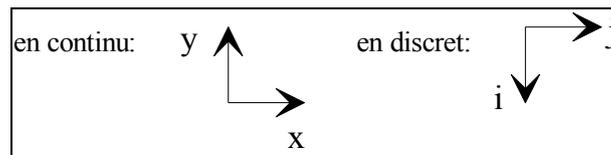
$A(x)$ est un signal monodimensionnel continu

$A(x, y)$ est un signal bidimensionnel continu.

$A[i]$ est un signal monodimensionnel discret.

$A[i, j]$ est un signal bidimensionnel discret.

+Convention d'orientation des axes :



Statistique d'ordre

$a_{(i)}$ est la $i^{\text{ème}}$ valeur d'un vecteur \mathbf{A} après classement.

Variable aléatoire, champ, probabilité.

L'image \mathbf{A} peut être considérée comme un champ aléatoire dont une réalisation est notée par \mathbf{a} , c'est-à-dire par la lettre minuscule correspondante.

A_s est une variable aléatoire au site s .

a_s est une réalisation de A_s

Fonction de répartition :

$F(a) = P(A \leq a)$.

Probabilité conditionnelle:

$$P(\Lambda | \mathbf{a}) = \frac{P(\mathbf{a} | \Lambda)P(\Lambda)}{P(\mathbf{a})} = \frac{P(\mathbf{A} = \mathbf{a}, \Lambda)}{P(\mathbf{a})}$$

Espérance :

$$E\{A(x)\} = \int_{-\infty}^{\infty} a \cdot f(a, x) da$$

avec les relations suivantes entre densité de probabilité et fonction de répartition :

$$f(a, x) = \frac{\partial F(a, x)}{\partial a} \text{ et } F(a, x) = P\{A(x) \leq a\}$$

Variance :

$$\text{var } A = E\{(A - E\{A\})^2\}$$

Vecteur

Un vecteur variable est noté par une lettre majuscule en caractère gras. Un vecteur constant est noté par une lettre minuscule en caractère gras. Le transposé d'un vecteur **u** se note : **^tu**.

Avant-propos

Aujourd'hui, c'est pour moi une très grande joie de participer au baptême de ce livre « *Analyse d'images : Filtrage et segmentation* » que nous offre une famille de neuf chercheurs et enseignants de grande réputation dans la communauté française du traitement et de l'analyse d'images. Cet ouvrage illustre pleinement le travail commun dans un GDR puisqu'il présente différentes approches testées et comparées sur une base de données communes.

La décision d'une création réellement collective est toujours un sérieux défi à la loi universelle de l'entropie maximum. De la sorte, le succès de l'entreprise est en même temps un fleuron pour les auteurs et un signe d'espérance pour tous leurs collègues. Peut-on calculer en kilo joules le ΔW , quantité supplémentaire de travail que Philippe BOLON, Jean-Marc CHASSERY, Jean-Pierre COCQUEREZ, Didier DEMIGNY, Christine GRAFFIGNE, Annick MONTANVERT, Sylvie PHILIPP, Rachid ZEBODJ et Josiane ZERUBIA ont dépensée pendant leurs trois années de collaboration, pour développer avec autant de cohérence leurs interférences croisées, de telle sorte que le tout soit infiniment plus riche que la somme des neuf parties ?

Sur la question de la segmentation des images, cet ouvrage assoit véritablement un état de l'art qui est unique, au niveau mondial. Le problème est analysé à fond, sous tous ses angles, intégrant différents types de modèles, allant de la perception aux modèles variationnels et aux approches non linéaires, rien n'y manque. Argumentant ces approches, une vision globale, unique elle aussi, apparaît dans les deux chapitres de comparaison. Ce livre rendra donc les plus précieux services aux étudiants, aux chercheurs et aux ingénieurs. C'est pourquoi je puis parier sans prendre de risque qu'il deviendra immédiatement un texte de référence.

Je profite de la tribune qui m'est donnée ici pour formuler mon vif souhait que cet ouvrage fasse des émules au sein du Groupement de Recherches « Traitement du Signal et Images » qui l'a aidé à naître, et qu'il se trouve ainsi le premier d'une série d'ouvrages collectifs français en signal et en image.

Odile MACCHI

Directrice du GDR « Traitement du Signal et Images »

Membre de l'Académie des Sciences

CHAPITRE I

Image et segmentation

Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel. Son élaboration résulte de la volonté de proposer une entité observable par l'œil humain. Ceci explique d'une part son aspect planaire et d'autre part le fait que l'information élémentaire associée à chaque point de l'image soit transcrite en niveau de gris ou en couleur.

Cette information élémentaire est, en général, une grandeur physique liée à la nature du capteur. Une caméra CCD mesure la quantité de lumière émise par l'objet, un récepteur de rayons X mesure la quantité de rayonnement transmise par l'organe (en imagerie médicale) ou la pièce (en contrôle d'inspection).

L'image est donc une collection d'informations qui, dans un premier temps, se présentait sur un support photographique qui permettait le traitement en différé d'un phénomène fugace, une analyse fine des phénomènes enregistrés et bien sûr l'archivage et l'illustration. Le *traitement d'images* est né de l'idée et de la nécessité de remplacer l'observateur humain par la machine. L'image ou les signaux provenant des capteurs ont alors été numérisés pour pouvoir être traités par l'ordinateur. Dans un deuxième temps, l'image a été codée et mémorisée sur différents supports (magnétique, magnéto-optique, *etc.*). La vision intervenant dans un grand nombre d'activités humaines, le champ des applications du traitement d'images est très vaste.

A partir d'une image numérique, il convient d'*extraire* les informations pertinentes en regard de l'application concernée, les *traiter* puis les *interpréter*. Le terme générique d'*analyse d'images* désigne l'ensemble de ces opérations.

Il est important à ce niveau de préciser que l'analyse d'images est allée au delà du remplacement de l'observateur humain. Des solutions novatrices ont été apportées pour traiter des problèmes correspondant à des situations auxquelles l'observateur humain n'avait jamais été confronté. C'est le cas notamment des images « non visibles » telles que les images acoustiques, ultrasonores et les images radar. Avec le développement de capteurs comme les scanners et les tomographes, de « nouvelles » images ont été élaborées dans le cadre, par exemple, de la médecine et du contrôle non-destructif, ouvrant ainsi considérablement le champ d'application du traitement d'images.

En analyse d'images, on distingue les *traitements de bas-niveau* et les *traitements de haut-niveau*. Cette distinction est liée au contenu sémantique des entités traitées et extraites de l'image. Les traitements de bas-niveau opèrent, en

général, sur les grandeurs calculées à partir des valeurs attachées à chaque point de l'image sans faire nécessairement la liaison avec la réalité qu'elles représentent. Par exemple, la détection de contours est un traitement de bas-niveau qui est effectué sans « comprendre » l'image, le contour détecté peut très bien ne pas correspondre à un bord d'objet dans la réalité et ne présenter aucun intérêt (création d'artefacts liés à un éclairage non homogène, par exemple). Les traitements de bas-niveau opèrent plutôt sur des données de nature *numérique*. A l'opposé, les traitements de haut-niveau s'appliquent à des entités de nature *symbolique* associées à une représentation de la réalité extraite de l'image ; ils sont relatifs à l'interprétation et à la compréhension de l'image exprimées avec des mots du vocabulaire de l'application. Par exemple, des zones d'une image aérienne peuvent être caractérisées par leur forme (rectangulaire, linéique,...), être étiquetées avec les termes : bâtiment, route, bosquet, ombres, *etc.* Des relations entre ces zones sont exploitées pour comprendre la scène étudiée, par exemple : une route ne peut pas être incluse dans un bosquet.

La *segmentation* est un traitement de bas-niveau qui consiste à créer une partition de l'image \mathbf{A} en sous-ensembles \mathbf{R}_i , appelés *régions*, tels que:

$$\forall i \mathbf{R}_i \neq \emptyset$$

$$\forall i, j ; i \neq j \mathbf{R}_i \cap \mathbf{R}_j = \emptyset$$

$$\mathbf{A} = \bigcup_i \mathbf{R}_i$$

Une *région* est un ensemble connexe de points image (*pixels*¹) ayant des propriétés communes (intensité, texture, ...) qui les différencient des pixels des régions voisines. Les connaissances utilisées sont le plus souvent du domaine de l'image numérique et du traitement du signal, donc sémantiquement assez pauvres.

Il n'y a pas de méthode unique de segmentation d'une image, le choix d'une technique est lié :

- à la nature de l'image :
 - éclairage non homogène, reflets,
 - présence de bruit, de zones texturées,
 - contours flous, en partie occultés,
- aux opérations situées en aval de la segmentation :
 - localisation, mesure, calcul 3D,
 - reconnaissance des formes, interprétation,
 - diagnostic, contrôle qualité,
- aux primitives à extraire :

¹pixel provient de l'expression **picture element**

- contours, segments de droite, angles,...
- régions, formes,
- textures,
- aux contraintes d'exploitation :
 - complexité algorithmique, fonctionnement en temps réel,
 - taille de la mémoire disponible en machine.

Du fait de cette diversité, il est difficile de définir, de manière absolue, une « bonne » segmentation. La segmentation n'est pas une fin en soi, sa qualité est fonction des résultats obtenus par les traitements situés en aval qui utilisent les primitives extraites.

La segmentation fait référence aux notions de différence et de similarité comme les perçoit le système visuel humain et ceci donne naissance à deux approches couramment qualifiées d'approche « *frontière* » et d'approche « *région* » :

- La notion de « *frontière* » est associée à une variation d'intensité ou à une discontinuité entre les propriétés de deux ensembles connexes de points. L'approche frontière regroupe les techniques de détection de contours. Ces méthodes ne conduisent pas directement à une segmentation de l'image telle qu'on l'a définie précédemment, car les contours obtenus sont rarement connexes, il faut donc procéder à une fermeture des contours si l'on souhaite une partition complète de l'image. En effet, après fermeture de contours, la dualité contours-régions apparaît nettement. Les régions sont définies topologiquement comme l'intérieur d'une ligne fermée.
- La notion de « *région* » fait référence à des groupements de points ayant des propriétés communes. Les méthodes de l'approche région aboutissent directement à une partition de l'image, chaque pixel étant affecté à une région unique.

Suivant les méthodes employées pour la traiter, l'image numérique peut être considérée comme :

- un signal bidimensionnel à support et à valeurs bornés que l'on notera :

$A[i, j]$ avec $[i, j] \in \mathbb{N}^2$ et $0 \leq i \leq M-1$; $0 \leq j \leq N-1$. L'image résulte de l'échantillonnage du signal continu $A(x, y)$. On désigne par $s = [i, j]$ un *site* de coordonnées $[i, j]$ dont la valeur sera notée $A[s]$ ou $A[i, j]$. Les sites constituent un ensemble \mathcal{S} organisé par un maillage (*cf.* chapitre III). Pour s donné, on appelle pixel le couple $(s, A[s])$.
- un processus stochastique $A[s]$.
- un vecteur aléatoire $\mathbf{A} = (A_1, \dots, A_s, \dots)$ ou A_s est une variable aléatoire associée au site s .

- une surface $(i, j, A[i, j])$ de l'espace \mathbf{N}^3

Chacune de ces représentations est en général associée à un modèle analytique, continu ou discret, stochastique, géométrique...

Dans la plupart des cas, la segmentation est plus aisée si elle est précédée de prétraitements. Les prétraitements les plus utilisés sont les suivants :

- les transformations ponctuelles de luminance,
- la réduction du bruit ou lissage,
- le rehaussement de contraste.

Le choix et le paramétrage de ces opérateurs de prétraitement dépendent évidemment de la nature de l'image ; ils sont parfois intimement imbriqués dans la technique de segmentation elle-même : c'est le cas, par exemple, de certains détecteurs de contours qui combinent lissage et dérivation (*cf.* chapitre V).

L'objectif de cet ouvrage est de présenter de manière *didactique* les principaux modes de prétraitements et les méthodes de segmentation les plus courantes et récentes. La plupart des traitements présentés ont été implantés par les différents auteurs. Ce travail en commun nous a permis d'*unifier les notations*, ce qui n'est pas une petite tâche compte tenu de la diversité des origines des travaux présentés. Une autre *caractéristique importante de l'ouvrage* est que les différentes méthodes ont été appliquées sur une même série d'images qui font partie de la banque de test du Groupement de Recherche TDSI « Traitement du Signal et Images » du CNRS qui a soutenu ce travail.

Comme nous l'avons indiqué précédemment, il est difficile de définir une segmentation optimale. Cependant, il est possible de comparer les méthodes quand elles sont appliquées sur des mêmes images réelles ou de synthèse construites pour tester le comportement des différents opérateurs de segmentation. Un autre point important est que ce livre comporte *deux chapitres consacrés à la comparaison des méthodes*. Ils ont pour objet, nous l'espérons, de guider le traiteur d'images dans ses choix, au travers de résultats quantitatifs et qualitatifs.

Suite à la présentation qui vient d'être faite, les différents chapitres de l'ouvrage sont organisés de la manière suivante :

- Le chapitre IV est consacré aux prétraitements.
- L'approche frontière est traitée dans les chapitres V (méthodes dérivatives, surfaciques et morphologiques), VI (méthodes d'inspiration neurobiologique), VII (méthodes variationnelles), VIII (méthodes markoviennes) avec le chapitre IX qui est une comparaison des résultats pour les opérateurs de cette approche.
- L'approche région est présentée dans les chapitres X (méthodes de classification préalables à une segmentation), XI (méthodes markoviennes) et

XII (méthodes structurales). Dans le même esprit que précédemment, le chapitre XIII comprend une comparaison des méthodes de cette approche.

Les chapitres II et III sont particuliers, ils ne traitent pas directement de la segmentation ; ils décrivent d'une part les grandeurs utilisées par les méthodes de segmentation et d'autre part les structures de données indispensables pour mettre en œuvre les différentes méthodes :

- Le chapitre II (attributs) est consacré à la présentation des principales grandeurs « calculables » sur un groupe de pixels et exploitables pour la segmentation mais aussi pour la classification et la reconnaissance des formes.
- L'image numérique est « volumineuse », son traitement pour être efficace implique une bonne structuration des données. Le chapitre III présente donc les structures de données associées au traitement numérique des images et quelques algorithmes de base pour les créer ou les manipuler.

Enfin plusieurs annexes comportent des rappels en traitement du signal, des introductions à la morphologie mathématique et aux champs de Markov, des études sur l'implémentation de certains opérateurs ainsi que des développements théoriques particuliers.

CHAPITRE II

Attributs de région

Dans ce chapitre, sont exposés les attributs pouvant caractériser un ensemble de pixels connexes d'une image, désigné sous le terme de *région*.

Ils sont utilisés en segmentation :

- par les méthodes markoviennes ou structurales présentées chapitres XI et XII,
- dans les méthodes de classification multidimensionnelle (*cf.* chapitre X), pour créer des ensembles de pixels « semblables », en recourant ou non à un corpus d'apprentissage.

Ce chapitre déborde largement du cadre de la segmentation, puisque les attributs présentés sont utilisés à diverses fins. Ils peuvent servir en tant que tels pour :

- décrire une région (paramètres de forme),
- effectuer des mesures (de taille, de radiométrie, de texture, *etc.*).

Ils peuvent également être employés :

- en post-traitement pour fusionner des régions sur des critères de similarité, de taille,
- pour classifier une région par rapport à des classes apprises,
- dans une phase d'interprétation d'image pour reconnaître un objet ou une partie d'objet,
- *etc.*

Que ces attributs soient calculés avant ou après la segmentation, ils caractérisent une région (soit en cours de construction, soit déjà figée). Ils sont utilisés dans une approche de l'analyse d'images qui exploite une certaine « homogénéité » de groupements connexes de pixels. Cette approche est duale de la recherche des contours qui s'intéresse aux « hétérogénéités » dans l'image. Cette notion d'homogénéité est souvent associée à la notion de « texture » qui est d'abord présentée, avant de faire le panorama des attributs proprement dits.

La nature des attributs dépend de la modélisation sous-jacente utilisée pour l'image. Ainsi, les attributs présentés dans les paragraphes 2, 3, 4 et 5 correspondent à des statistiques (moments d'espace, histogramme, « cooccurrence », ...) calculées sur l'image qui est alors considérée comme la réalisation d'un processus stochastique. Le paragraphe 6 est relatif à l'analyse spectrale de l'image. Le paragraphe 7 qui expose la notion de contraste constitue

une transition vers les paragraphes 8 et 9 consacrés respectivement à une introduction à la théorie fractale et aux attributs surfaciques (l'image est alors une surface $(i, j, A[i, j])$). Les attributs géométriques du paragraphe 10 sont relatifs à la forme d'une région. Le dernier paragraphe définit quelques attributs inter-régions.

Les notations suivantes sont utilisées :

R désigne une *région* de l'image qui peut être un voisinage d'un pixel, une fenêtre rectangulaire, ou plus généralement, un sous-ensemble de forme quelconque de pixels connexes de l'image.

K est le nombre de pixels de cette région.

Par abus de langage, on appellera niveau de gris la grandeur $A[s]$ attachée au site s . Les attributs se calculent en général à partir des niveaux de gris des pixels de la région, que l'on supposera appartenir à un ensemble $\Omega = \{0, 1, \dots, \omega_i, \dots, L-1\}$.

1 Texture

La texture [RAO90] se décrit en termes linguistiques tels que la rugosité, le contraste, la finesse, la régularité, *etc.*, que les chercheurs se sont efforcés de caractériser depuis plusieurs années. C'est ainsi que l'on trouvera dans les paragraphes suivants plusieurs définitions du contraste ou de l'homogénéité par exemple, qui toutes tendent à donner une définition mathématique du même phénomène visuel.

Une texture présente, à une échelle donnée, le même aspect quelle que soit la zone observée (*cf.* figure 2.4). Dans ces conditions, on considère l'image comme la réalisation d'un processus stochastique stationnaire : c'est-à-dire que les statistiques calculées sont invariantes en translation.

On rencontre deux types de définition de la texture :

- La première est *déterministe* et fait référence à une répétition spatiale d'un motif de base dans différentes directions. Cette approche structurelle correspond à une vision *macroscopique* des textures. C'est le cas, par exemple, des tissus ou de la peau de reptiles (la primitive dans ce cas est l'écaille). Ainsi, B. Julesz, dès 1962, considère la texture comme l'agencement d'un motif de base qu'il appelle « texton » [JULE62].
- La deuxième est *probabiliste* et cherche à caractériser l'aspect anarchique et homogène qui ne comprend ni de motif localisable, ni de fréquence de répétition principale (cas de l'herbe). Elle correspond à une vision *microscopique*.

Gagalowicz propose une synthèse des deux approches en considérant la texture comme « une structure spatiale constituée de l'organisation de *primitives* (ou motifs de base) ayant chacune un aspect aléatoire » [GAGA83]. C'est donc « une structure hiérarchique à deux niveaux » (*cf.* figure 2.1)[VOLE87].

Certains des attributs que nous présentons dans cette partie sont des statistiques d'ensemble qui doivent, en toute rigueur, être calculés sur une infinité de

réalisations du processus stochastique (donc une infinité d'images *cf.* annexe D). On suppose en général que le processus est stationnaire et *ergodique* : une seule réalisation (une seule image) permet alors de calculer les statistiques spatiales sur une région de l'image. Mais, il ne faut pas oublier que ces statistiques spatiales ne sont égales aux moyennes et moments d'ensemble que lorsque le nombre d'éléments de la région concernée tend vers l'infini.

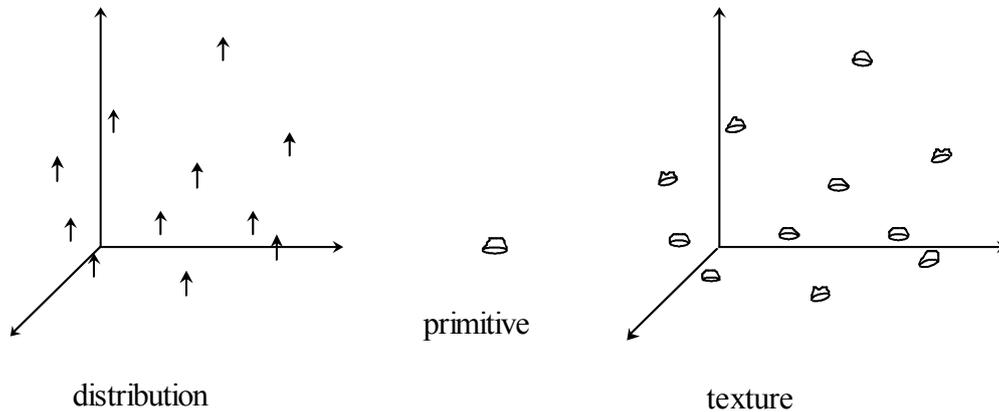


Figure 2.1 : Modélisation de la texture avec une distribution dans \mathbf{R}^3 et une primitive de \mathbf{R}^3

2 Attributs stochastiques d'une image

Dans ce paragraphe, on considère l'image comme un processus aléatoire discret $A[s]$ à valeur dans \mathbf{R} . Si s est fixé, $A[s]$ est alors une variable aléatoire pour laquelle on peut définir différentes grandeurs (statistiques d'ensemble) qui permettent de caractériser le processus $A[s]$ à partir de différentes réalisations :

moyenne :

Pour un signal continu $A(x)$, l'espérance ou moyenne d'ensemble est définie par :

$$\mu(x) = E\{A(x)\} = \int_{-\infty}^{+\infty} a \cdot f(a, x) da \quad (2-1)$$

où f est la densité de probabilité de A (*cf.* annexe D).

Pour un signal discret, la moyenne est définie par :

$$\mu[s] = E\{A[s]\} = \sum_i p_i \cdot a_i \quad (2-2)$$

où a_i est une réalisation de la variable $A[s]$ et p_i est la probabilité $P(A[s]=a_i)$

moment d'ordre k :

$$E\{A^k[s]\} = \sum_i p_i \cdot a_i^k \quad (2-3)$$

moment centré d'ordre k :

$$\mu_k(s) = E\{(A[s] - \mu[s])^k\} \quad (2-4)$$

autocorrélation :

$$R[s_1, s_2] = E\{A[s_1] \cdot A[s_2]\} \quad (2-5)$$

3 Moments et moyennes d'espace d'une région

Pour une région R de K pixels, on définit les différents moments d'espace, non centrés et centrés :

moment d'espace du premier ordre ou moyenne :

$$m_1 = \frac{1}{K} \sum_{s \in R} A[s] \quad (2-6)$$

moment d'espace centré du second ordre ou variance :

$$m_2 = \frac{1}{K} \sum_{s \in R} (A[s] - m_1)^2 \quad (2-7)$$

L'écart-type, noté σ , est la racine carrée de la variance.

moment d'espace centré d'ordre k :

$$m_k = \frac{1}{K} \sum_{s \in R} (A[s] - m_1)^k \quad (2-8)$$

moment d'espace composé du second ordre ou autocovariance :

Pour une translation \mathbf{t} du plan, le moment d'espace du second ordre est donné par la formule :

$$M_2(\mathbf{t}) = \frac{1}{N_c} \sum_{s \in R, s+\mathbf{t} \in R} (A[s] - m_1)(A[s + \mathbf{t}] - m_1) \quad (2-9)$$

où N_c est le nombre de couples $(s, s+\mathbf{t})$ de la région R .

L'autocovariance est périodique quand la texture est périodique.

moment d'espace composé d'ordre k :

Pour $k-1$ translations dans le plan, $\mathbf{t}_1, \dots, \mathbf{t}_{k-1}$,

$$M_k(\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{k-1}) = \frac{1}{N_k} \sum_{\substack{s \in R \\ s+\mathbf{t} \in R}} (A[s] - m_1)(A[s + \mathbf{t}_1] - m_1) \dots (A[s + \mathbf{t}_{k-1}] - m_1)$$

où N_k est le nombre de k -uplets $(s, s+\mathbf{t}_1, \dots, s+\mathbf{t}_{k-1})$ de la région R .

Le processus est *stationnaire du premier ordre*, si $E\{A[s]\}$ est indépendant du site. Si, de plus, il est *stationnaire du deuxième ordre*, l'autocorrélation ne dépend que de la translation \mathbf{t} entre deux sites :

$$R[s_1, s_2] = E\{A[s_1].A[s_2]\} = R[\mathbf{t}] = E\{A[s].A[s+\mathbf{t}]\} \quad (2-10)$$

Quand les processus sont stationnaires, on peut exploiter les moments d'espace, en observant une réalisation sur un ensemble de sites, au lieu d'envisager différentes réalisations en différents sites. De plus, si le processus est ergodique d'ordre 2, quand \mathbf{R} tend vers le plan tout entier \mathbf{N}^2 , on a :

$$m = \lim_{\mathbf{R} \rightarrow \mathbf{N}^2} \frac{1}{K} \sum_{s \in \mathbf{R}} A[s] = E\{A[s]\} \quad \text{et} \quad (2-11)$$

$$M_2(\mathbf{t}) = \lim_{\mathbf{R} \rightarrow \mathbf{N}^2} \frac{1}{N_c} \sum_{s \in \mathbf{R}, s+\mathbf{t} \in \mathbf{R}} A[s].A[s+\mathbf{t}] = E\{A[s].A[s+\mathbf{t}]\} \quad (2-12)$$

L'étude des seuls moments ne suffit pas pour discriminer des textures. En effet, A. Gagalowicz a pu synthétiser des textures différenciables à l'œil et possédant des moments identiques jusqu'à l'ordre 4, voire 5 [GAGA83].

histogramme normalisé ou moyenne d'espace du premier ordre :

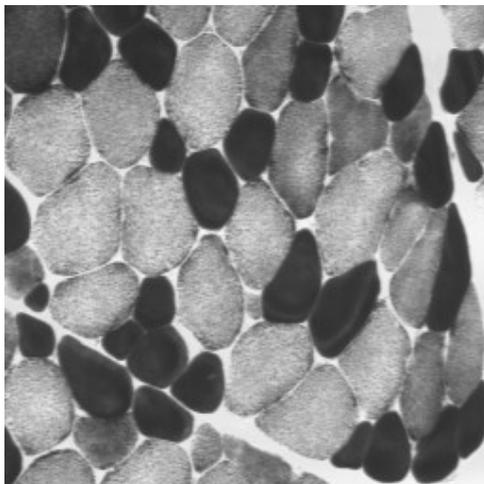
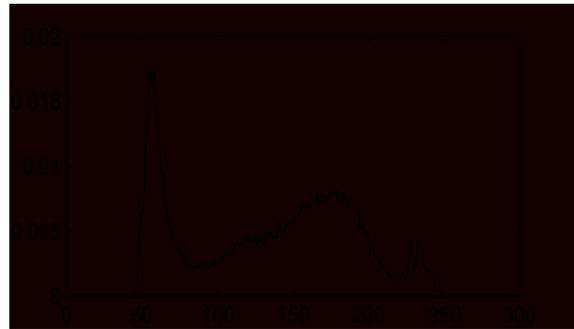


image MUSCLE



histogramme calculé sur toute l'image

Figure 2.2 : Exemple d'histogramme normalisé comportant plusieurs modes

Pour $\omega \in \Omega = \{0, \dots, L-1\}$, on définit :

$$h(\omega) = \frac{1}{K} \sum_{s \in \mathbf{R}} \delta(A[s] - \omega) \quad \text{où} \quad \delta(A[s] - \omega) = \begin{cases} 1 & \text{si } A[s] = \omega \\ 0 & \text{sinon} \end{cases} \quad (2-13)$$

Pour un grand nombre de réalisations, on estime $P(A(s) = \omega)$ par $h(\omega)$. h n'est autre que la densité empirique de probabilité de A .

histogramme des cooccurrences ou moyenne d'espace du second ordre :

$$h_{\mathbf{t}}(\omega_i, \omega_j) = \frac{1}{N_c} \sum_{s \in R} \delta(A[s] - \omega_i) \cdot \delta(A[s + \mathbf{t}] - \omega_j) \quad (2-14)$$

Il permet d'estimer $P(A[s] = \omega_i, A[s + \mathbf{t}] = \omega_j)$ qui est la probabilité conjointe ou *cooccurrence* d'un pixel de niveau de gris ω_i et d'un pixel de niveau de gris ω_j , séparés par la translation \mathbf{t} .

B. Julesz [JULE62] s'est intéressé le premier aux statistiques du second ordre pour la discrimination des textures. Il a proposé la conjecture suivante : « le système visuel ne peut discriminer deux textures si celles-ci possèdent les mêmes distributions de probabilités du premier et du second ordre ». Il a néanmoins trouvé des contre-exemples à cette conjecture avec des textures macroscopiques (contenant des éléments déterministes ou « textons »).

A. Gagalowicz [GAGA83] a constaté sur des expériences psychovisuelles, que l'ensemble des attributs discriminants d'une texture homogène est supérieur à l'ensemble formé par les moments des deux premiers ordres et inférieure à l'ensemble des moyennes d'espace du second ordre. Ces dernières sont rassemblées dans des matrices appelées *matrices de cooccurrence* et étudiées dans le paragraphe 4.

Cet ensemble de paramètres statistiques, qui contient toute l'information texturale, a l'inconvénient d'être difficilement manipulable (problèmes d'occupation de la mémoire, de temps de calcul). C'est pourquoi on en extrait différents attributs, appelés indices de texture du second ordre et présentés au paragraphe 4.1.

moyenne d'espace d'ordre k :

Il s'agit d'une généralisation de la définition précédente qui permet d'estimer :

$$P(A[s] = \omega_1, A[s + \mathbf{t}_1] = \omega_2, \dots, A[s + \mathbf{t}_{k-1}] = \omega_k) \quad (2-15)$$

pour $k-1$ translations

4 Matrices de cooccurrence

4.1 Matrices de cooccurrence et indices de texture du second ordre

Les matrices de cooccurrence contiennent les moyennes d'espace du second ordre.

Pour une translation \mathbf{t} , la matrice de cooccurrence $\mathbf{MC}_{\mathbf{t}}$ d'une région R est définie pour tout couple de niveaux de gris (a, b) par :

$$\mathbf{MC}_{\mathbf{t}}(a, b) = \text{card}\{(s, s + \mathbf{t}) \in R^2 | A[s] = a, A[s + \mathbf{t}] = b\} \quad (2-16)$$

$MC_t(a,b)$ est donc le nombre de couples de sites $(s,s+t)$ de la région considérée, séparés par le vecteur de translation t et tel que s a pour niveau de gris a et $s+t$ pour niveau de gris b (cf. figure 2.3).

Pour une image quantifiée sur L niveaux de gris, la matrice MC_t est une matrice $L \times L$. En pratique, on réduit généralement cet ensemble de niveaux de gris à 8 ou 16 valeurs [PHIL88].

1	2	1	3	4
2	3	1	2	4
3	3	2	1	1

région de 15 sites

a	b	0	1	2	3	4	5	6	7
0		0	0	0	0	0	0	0	0
1		0	1	2	0	0	0	0	0
2		0	1	0	2	0	0	0	0
3		0	0	1	1	0	0	0	0
4		0	1	0	0	1	0	0	0
5		0	0	0	0	0	0	0	0
6		0	0	0	0	0	0	0	0
7		0	0	0	0	0	0	0	0

matrice de cooccurrence

Figure 2.3 : Exemple de matrice de cooccurrence calculée sur une région de 15 pixels, quantifiés sur 8 niveaux de gris et avec la translation $t = (0,1)$ (un pixel vers le bas)

La figure 2.4 présente les matrices de cooccurrence calculées sur trois textures naturelles (LAINE, TISSU et HERBE) issues de l'album de Brodatz [BROD66]. Ce sont des images 256×256 . Les matrices sont calculées sur une fenêtre 20×20 placée au centre de l'image. Les niveaux de gris sont ramenés à 16 valeurs. Les trois matrices sont nettement différentes : si on examine la position des valeurs non nulles des matrices, on observe que les niveaux de gris de HERBE sont en moyenne plus faibles que ceux de TISSU et, que ceux de LAINE ont une plus faible dynamique. Les valeurs non nulles sont plus étalées dans l'espace (a,b) et plus éloignées de la diagonale pour HERBE que pour les autres textures, ce qui correspond à un plus grand contraste de l'image. Ces notions seront formalisées par les indices présentés ci-dessous.

Les matrices de cooccurrence contiennent une masse d'information trop importante et difficilement manipulable dans son intégralité (256 valeurs dont beaucoup de zéros dans les exemples précédents). Quatorze indices prenant en compte l'ensemble de la matrice MC_t ont été définis par Haralick [HARA73] [HARA79]. Ils correspondent à des caractères descriptifs des textures (contraste, homogénéité, etc.). Nous exposons ici les indices les plus fréquemment utilisés ; ils sont présentés sous forme normalisée, c'est-à-dire qu'ils prennent des valeurs comprises entre 0 et 1 [PHIL88].

homogénéité :

$$\frac{1}{N_c^2} \sum_a \sum_b (MC_t(a, b))^2 \quad (2-17)$$

Cet indice est d'autant plus élevé que l'on retrouve souvent le même couple de pixels, ce qui est le cas quand le niveau de gris est uniforme ou quand il y a périodicité dans le sens de la translation.

contraste :

$$\frac{1}{N_c(L-1)^2} \sum_{k=0}^{L-1} k^2 \sum_{|a-b|=k} MC_t(a, b) \quad (2-18)$$

Chaque terme de la matrice MC_t est pondéré par sa distance à la diagonale. On obtient un indice correspondant à la notion usuelle du contraste, il est élevé quand les termes éloignés de la diagonale de la matrice sont élevés, c'est-à-dire quand on passe souvent d'un pixel très clair à un pixel très foncé ou inversement.

entropie :

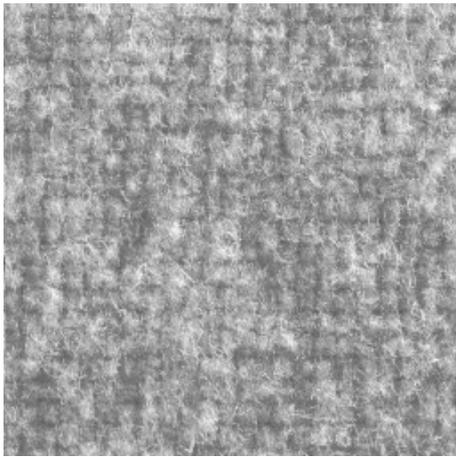
$$1 - \frac{1}{N_c \cdot \ln(N_c)} \sum_a \sum_b MC_t(a, b) \cdot \ln(MC_t(a, b)) \cdot \mathbf{1}_{MC_t(a, b)} \quad (2-19)$$

avec $\mathbf{1}_{MC_t(a, b)} = \begin{cases} 1 & \text{si } MC_t(a, b) \neq 0 \\ 0 & \text{sinon} \end{cases}$

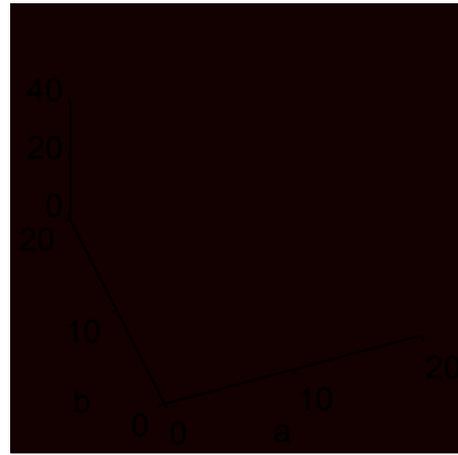
L'entropie est faible si on a souvent le même couple de pixels, forte si chaque couple est peu représenté. Elle fournit un indicateur du désordre que peut présenter une texture.

corrélation :

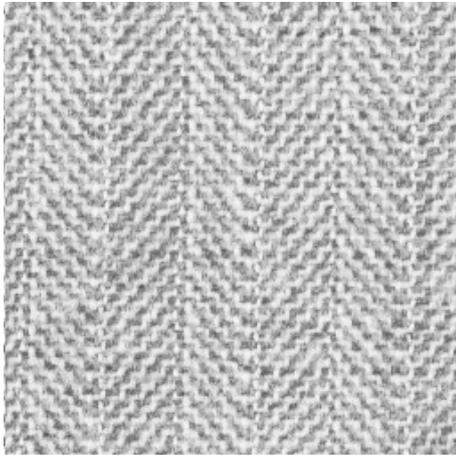
$$\frac{1}{N_c \cdot \sigma_x \cdot \sigma_y} \left| \sum_a \sum_b (a - m_x)(b - m_y) MC_t(a, b) \right| \quad (2-20)$$



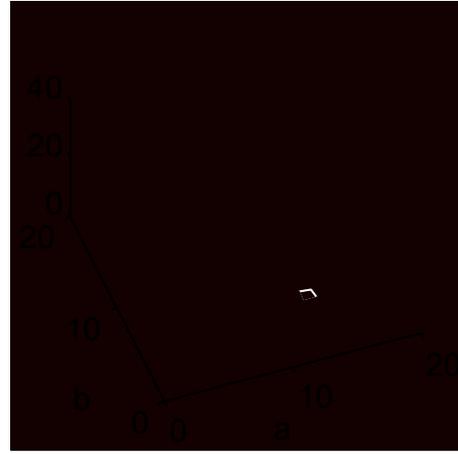
LAINÉ (Brodatz D19)



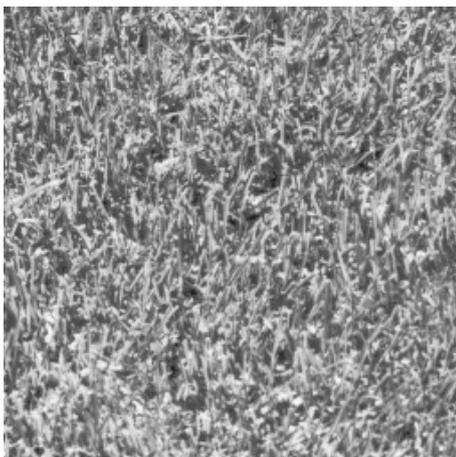
matrice de cooccurrence de LAINÉ



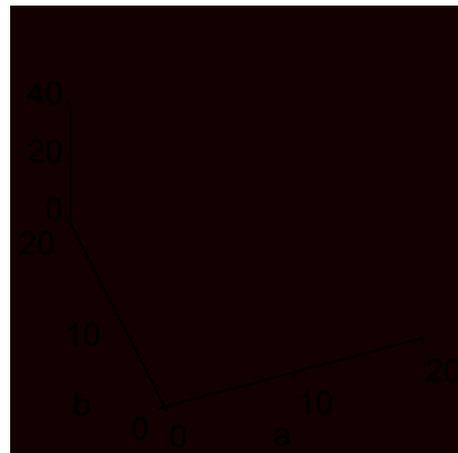
TISSU (Brodatz D17)



matrice de cooccurrence de TISSU



HERBE (Brodatz D9)



matrice de cooccurrence de HERBE

Figure 2.4 : Matrices de cooccurrence de trois textures naturelles calculées sur une fenêtre 20×20 positionnée au centre de l'image, pour une translation de 1 pixel vers la droite ; les pixels sont quantifiés sur 16 niveaux

où :

$$m_x = \frac{1}{N_c} \sum_a \sum_b a MC_t(a, b) \quad (2-21)$$

est la moyenne pondérée des lignes de MC_t ,

$$m_y = \frac{1}{N_c} \sum_a \sum_b b MC_t(a, b) \quad (2-22)$$

est la moyenne pondérée des colonnes,

$$\sigma_x^2 = \frac{1}{N_c} \sum_a \sum_b (a - m_x)^2 MC_t(a, b) \quad (2-23)$$

est la variance des lignes,

$$\sigma_y^2 = \frac{1}{N_c} \sum_a \sum_b (b - m_y)^2 MC_t(a, b) \quad (2-24)$$

est la variance des colonnes.

L'attribut décrit par (2-21) rend compte de la corrélation entre les lignes et les colonnes de la matrice de cooccurrence, c'est-à-dire entre le niveau de gris du pixel d'origine et celui d'arrivée de la translation.

homogénéité locale :

$$\frac{1}{N_c} \sum_a \sum_b \frac{1}{1 + (a - b)^2} MC_t(a, b) \quad (2-25)$$

directivité :

$$\frac{1}{N_c} \sum_a MC_t(a, a) \quad (2-26)$$

La directivité est importante s'il y a des pixels de même niveau de gris séparés par la translation t .

uniformité :

$$\frac{1}{N_c^2} \sum_a MC_t^2(a, a) \quad (2-27)$$

Elle est d'autant plus importante qu'un seul niveau de gris apparaît souvent dans la direction de la translation.

Ces indices sont corrélés, mais ils permettent de réduire notablement l'information contenue dans la matrice de cooccurrence. R.W. Connors et C.A. Harlow [CONN80b] ont prouvé de façon théorique que les 5 premiers indices présentés (l'homogénéité, le contraste, l'entropie, la corrélation et l'homogénéité locale) permettaient une meilleure discrimination des textures que

d'autres attributs tels que longueur des isosegments, mesure de gradient et spectre de puissance.

4.2 Périodes calculées à partir des cooccurrences

Certaines images, notamment les images de texture, présentent un aspect périodique. La période est un attribut important qui peut être estimé à partir du spectre. Elle peut aussi servir à l'extraction des primitives et de leur règle de placement dans la texture. La période élémentaire est toujours un parallélogramme (cf. figure 2.5). Ce parallélogramme n'est pas unique mais on peut déterminer celui qui est défini par les côtés les plus courts [CONN80a]. On notera \mathbf{f} et \mathbf{g} les vecteurs directeurs des côtés de ce parallélogramme.

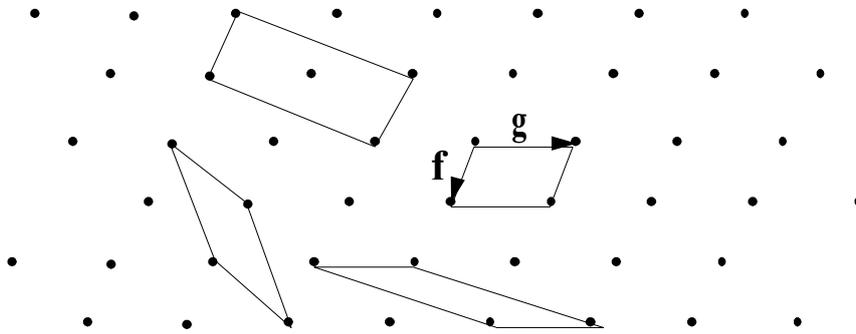


Figure 2.5 : Exemples de périodes élémentaires d'une texture périodique : la plus petite période est définie par les vecteurs \mathbf{f} et \mathbf{g}

Pour l'étude des textures périodiques, R.W. Connors et C.A. Harlow [CONN80a] ont proposé d'utiliser l'un des indices de texture du second ordre pour déterminer la période élémentaire et les règles de placement.

Si l'on considère une texture périodique dont une période est \mathbf{f} et que l'on ne s'intéresse qu'aux translations \mathbf{t} de même direction que \mathbf{f} , la matrice de cooccurrence vérifie les trois propriétés suivantes :

- $\mathbf{MC}_{\mathbf{t}}$ est une matrice diagonale
- $\mathbf{MC}_{\mathbf{t}}$ a des éléments non diagonaux différents de zéro pour tout \mathbf{t} tel que $\|\mathbf{t}\| < \|\mathbf{f}\|$ (norme euclidienne).
- $\mathbf{MG}_{m\mathbf{f}+\mathbf{t}}$ pour tout entier m ; c'est-à-dire que les matrices de cooccurrence ont la même périodicité que la texture.

R. W. Connors et C. A. Harlow ont montré que tout indice, calculé à partir des matrices de cooccurrence et dépendant du carré de la différence des niveaux de gris est un bon indicateur de la périodicité des matrices. Par exemple, le contraste défini en (2-18), est un indice périodique de même période que la texture.

4.3 Matrices de cooccurrence généralisées

Les matrices de cooccurrence généralisées [DAVI79] [DAVI81] ont pour but de décrire l'arrangement spatial de primitives d'image telles que les contours ou les lignes. Elles généralisent la notion de matrice de cooccurrence à l'étude des distributions spatiales de primitives et non plus seulement aux distributions des niveaux de gris.

Elles nécessitent de se donner :

- une primitive P
- un attribut p de cette primitive
- un prédicat spatial **Pred** liant deux primitives et pouvant prendre les deux valeurs vrai et faux.

La *matrice de cooccurrence généralisée* est définie pour un attribut p et un prédicat spatial **Pred**. Soit i et j deux valeurs de l'attribut p , le terme $G(i,j)$ de la matrice de cooccurrence généralisée s'exprime comme suit :

$$G(i,j) = \frac{\text{card}\{(P_1, P_2) | p(P_1) = i, p(P_2) = j, \mathbf{Pred}(P_1, P_2) = \text{vrai}\}}{\text{card}\{(P_1, P_2) | \mathbf{Pred}(P_1, P_2) = \text{vrai}\}} \quad (2-28)$$

Cela correspond à la proportion de couples de primitives (P_1, P_2) , vérifiant le prédicat spatial **Pred** et pour lesquelles l'attribut vaut respectivement i et j .

Par exemple, les primitives peuvent être les pixels, les pixels de contour, les portions de contour :

- la primitive est le pixel.

Si l'attribut est le niveau de gris du pixel et si le prédicat est vrai quand les deux pixels sont séparés par une translation \mathbf{t} fixée, on retrouve la matrice de cooccurrence.

- la primitive est un pixel de contour.

L'attribut peut être l'orientation du contour, son contraste ou simplement la position du pixel dans l'image.

Davis a proposé différents prédicats spatiaux avec comme attribut l'orientation du contour :

Pred₁(P_1, P_2, k) est vrai si la distance entre deux pixels P_1, P_2 de contour est inférieure ou égale à k ,

Pred₂(P_i, P_j, k) est vrai pour les primitives P_i et P_j si P_j appartient au secteur d'angle θ limité par k et ayant pour axe le vecteur \mathbf{f}_i tangent au contour en P_i (zone grisée sur la figure 2.6a),

Pred₃(P_i, P_j, k) est identique à **Pred**₂(P_i, P_j, k) mais pour la direction perpendiculaire à \mathbf{f}_i (cf. figure 2.6b).

$\mathbf{Pred}_2(P_i, P_j, k)$ et $\mathbf{Pred}_3(P_i, P_j, k)$ servent à déterminer l'élongation et la largeur des éléments de texture.



Figure 2.6 : Exemples de positions admissibles pour P_j avec comme attribut l'orientation

- la primitive est une portion de contour :

L'attribut peut être la position de l'une des extrémités, l'orientation ou la longueur du contour.

Comme pour les matrices de cooccurrence, Davis *et al.* ont défini des attributs en vue de classifier les textures. Ils sont calculés pour un attribut p et un prédicat spatial \mathbf{Pred} :

homogénéité :

$$\sum_i \sum_j G(i, j)^2 \quad (2-29)$$

Cet indice est faible quand tous les éléments de G sont égaux.

contraste :

$$\sum_i \sum_j d(i, j) \cdot G(i, j) \quad (2-30)$$

où $d(i, j)$ est une mesure de dissemblance qui dépend de l'attribut p utilisé ; par exemple, si p est l'orientation, on peut prendre $d(i, j) = |\sin(i, j)|$

entropie :

$$\sum_i \sum_j G(i, j) \ln(G(i, j)) \mathbf{1}_{G(i, j)} \quad (2-31)$$

$$\text{où: } \mathbf{1}_{G(i, j)} = \begin{cases} 1 & \text{si } G(i, j) \neq 0 \\ 0 & \text{sinon} \end{cases}$$

corrélation:

$$\sum_i \sum_j \frac{ijG(i,j) - m_x^i m_y^j}{\sigma_x^i \cdot \sigma_y^j} \quad (2-32)$$

où m_x^i et m_y^j sont les moyennes respectives de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne et σ_x^i et σ_y^j sont les écarts-types respectifs de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne.

5 Longueurs de plage (ou isosegments)

Une *plage* de niveaux de gris (ou un *isosegment*) est un ensemble de pixels consécutifs, dans une direction donnée, ayant le même niveau de gris [GALL75]. La longueur d'une plage (« run length ») est le nombre de pixels dans cet ensemble. À chaque direction, on peut associer une matrice de longueurs de plages $\mathbf{P} = (p_\theta(i, j))$; l'élément $p_\theta(i, j)$ de cette matrice représente le nombre de plages de longueur j (dans la direction θ) constituées de pixels de niveau de gris i .

Dans une région, on peut extraire différents attributs de cette matrice de L lignes et n_θ colonnes (L est le nombre de niveaux de gris dans l'image et n_θ est la longueur de la corde maximale de direction θ dans la région) :

nombre de longueurs de plage :

$$SLP = \sum_{i=0}^{L-1} \sum_{j=1}^{n_\theta} p_\theta(i, j) \quad (2-33)$$

proportion de petites plages :

$$RF_1 = \frac{1}{SLP} \sum_{i=0}^{L-1} \sum_{j=1}^{n_\theta} \frac{p_\theta(i, j)}{j^2} \quad (2-34)$$

proportion de grandes plages :

$$RF_2 = \frac{1}{SLP} \sum_{i=0}^{L-1} \sum_{j=1}^{n_\theta} j^2 p_\theta(i, j) \quad (2-35)$$

hétérogénéité des niveaux de gris :

$$RF_3 = \frac{1}{SLP} \sum_{i=0}^{L-1} \left(\sum_{j=1}^{n_\theta} p_\theta(i, j) \right)^2 \quad (2-36)$$

RF_3 mesure la dispersion des plages entre les niveaux de gris.

hétérogénéité des longueurs de plage :

$$RF_4 = \frac{1}{SLP} \sum_{j=1}^{n_\theta} \left(\sum_{i=0}^{L-1} p_\theta(i, j) \right)^2 \quad (2-37)$$

RF₄ mesure la dispersion des plages entre les longueurs.

pourcentage de plages :

$$RF_5 = SLP / K \quad (K \text{ est le nombre total de pixels dans la région}) \quad (2-38)$$

6 Spectre

La transformée de Fourier permet de passer d'une représentation de l'image dans le domaine spatial (coordonnées m,n) à une représentation dans le domaine fréquentiel (coordonnées u,v).

La transformée de Fourier d'une séquence 2D correspondant au signal discret A[m,n] défini sur Z² s'exprime sous la forme :

$$F(u, v) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} A[m, n] \cdot e^{-2j\pi(mu+nv)} \quad (2-39)$$

Cette fonction de période 2π en u et v est la superposition des transformées de Fourier décalées du signal continu A(x,y) [GONZ92].

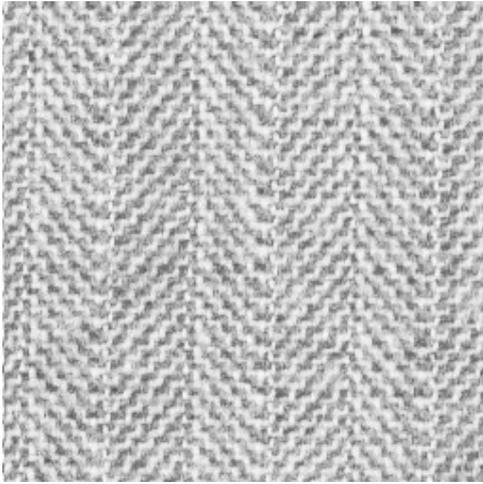
En pratique, une image A[s] = A[m,n], avec m et n entiers, 0 ≤ m ≤ M-1 et 0 ≤ n ≤ N-1 est un signal à support borné. Sa transformée de Fourier discrète est alors donnée par :

$$F[u, v] = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A[m, n] \cdot e^{-2j\pi \left(\frac{um}{M} + \frac{vn}{N} \right)} \quad (2-40)$$

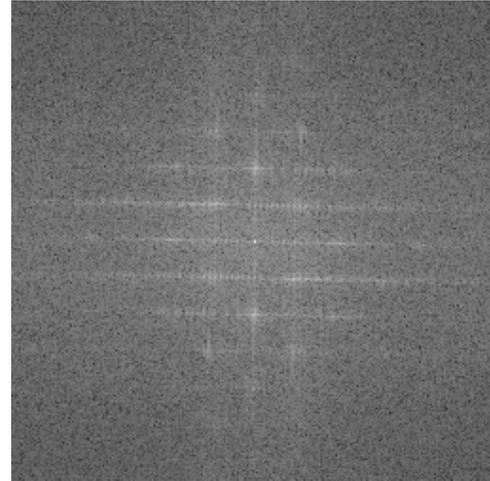
avec 0 ≤ u ≤ M-1 et 0 ≤ v ≤ N-1

Pour éviter les problèmes délicats d'interprétation de la phase, on exploite souvent le module de la transformée de Fourier ou *spectre de Fourier*

La figure 2.7 présente une texture de tissu, issue de [BROD66]. Le spectre est symétrique par rapport à l'origine. La dynamique du spectre d'une image est, en général, élevée et pour une bonne lisibilité du spectre, on affiche habituellement ln(|F(u, v)|). Le lignage presque vertical (resp. horizontal) sur l'image conduit aux lignes presque horizontales (resp. verticales) du spectre. Les périodicités de l'image se traduisent par des périodicités du spectre, qui sont beaucoup plus facile à extraire que dans le plan spatial (*cf.* § 6.2).



Texture périodique TISSU [Brodatz D17]



Spectre de TISSU centré à l'origine

Figure 2.7 : Spectre de Fourier d'une image 256×256

Le spectre est très riche en information. On peut en extraire les composantes fréquentielles les plus énergétiques de l'image. Il peut être exploité aussi pour calculer des filtres passe-bande, *etc.* Le paragraphe 6.2 présente une application à la recherche de motifs périodiques.

6.1 Densité spectrale de puissance

La *densité spectrale de puissance*, notée $S[u,v]$ est la transformée de Fourier discrète de la fonction d'autocorrélation (Eq 2-5) :

$$S[u,v] = \sum_{p=-\infty}^{+\infty} \sum_{q=-\infty}^{+\infty} R[p,q] e^{-2j\pi(pu+qv)} \quad (2-41)$$

Elle existe pour des signaux de puissance finie, on peut la calculer directement à partir de la transformée de Fourier. Pour un signal ergodique $A[m,n]$, on démontre que :

$$S[u,v] = \lim_{M,N \rightarrow +\infty} E \left\{ \frac{1}{(2M+1)(2N+1)} \left| \sum_{m=-M}^{+M} \sum_{n=-N}^{+N} A[m,n] e^{-2j\pi(mu+nv)} \right|^2 \right\}$$

En pratique, on ne peut pas calculer le spectre à l'aide de cette expression, en raison des dimensions finies de l'image. On exploite alors le *périodogramme* comme estimateur spectral. Son expression est donnée par :

$$\hat{P}[u,v] = \frac{1}{(2M+1)(2N+1)} \left| \sum_{m=-M}^{+M} \sum_{n=-N}^{+N} W[m,n] A[m,n] e^{-2j\pi(mu+nv)} \right|^2 \quad (2-42)$$

où $W[m,n]$ est une fonction déterministe de pondération (par exemple une fenêtre de Hanning), introduite pour réduire la puissance des lobes secondaires dus au fenêtrage rectangulaire imposé par les dimensions finies de l'image (*cf.* chapitre 13 de [PAPO91]).

Une méthode indirecte du calcul du spectre de puissance utilise la fonction d'autocorrélation, qui peut être calculée :

- sous forme de moyenne statistique, si le signal est stationnaire :

$$R[p,q] = E\{A[m,n]A[m+p,n+q]\} \quad (2-43)$$

- et sous forme de moyenne spatiale, si le signal est ergodique :

$$R_{p,q} = \lim_{M, N \rightarrow +\infty} \frac{1}{(2M+1)(2N+1)} \sum_{m=-M}^{+M} \sum_{n=-N}^{+N} A_{m,n} \cdot A_{m+p,n+q} \quad (2-44)$$

6.2 Périodes calculées à partir du spectre

Matsuyama *et al.* [MATS83] proposent une méthode pour déterminer les deux vecteurs \mathbf{f} et \mathbf{g} engendrant la période minimale (*cf.* § 4.2) à partir des périodes calculées dans le plan de Fourier.

Si la texture est exactement périodique, toute l'énergie du spectre est concentrée sur les fréquences correspondant aux périodes définies par : $\mathbf{h} = m\mathbf{f} + n\mathbf{g}$, $m, n \in \mathbb{Z}$. Même dans le cas de textures périodiques bruitées ou présentant des irrégularités dans la répartition des primitives (cas de textures naturelles telles que des images de tissu par exemple), l'énergie du spectre aux fréquences $m\mathbf{f} + n\mathbf{g}$ demeure un critère discriminant.

Pour une image de dimension $M \times M$, les relations entre les périodes spatiales \mathbf{f} et \mathbf{g} et les périodes spectrales u et v sont :

$$|u| = \frac{M}{|\mathbf{f}| \cos \theta} \quad \text{et} \quad |v| = \frac{M}{|\mathbf{g}| \sin \theta} \quad (2-45)$$

$$\arg u = \arg \mathbf{g} + \frac{\pi}{2} \pmod{\pi} \quad \text{et} \quad \arg v = \arg \mathbf{f} + \frac{\pi}{2} \pmod{\pi} \quad (2-46)$$

$$\text{avec } \theta = \arg(\mathbf{g}) - \arg(\mathbf{f}) \quad (2-47)$$

Il y a donc un décalage de $\pi/2$ entre les directions des périodes dans le plan-image et dans le plan des fréquences. Les périodes sont calculées plus précisément dans le plan de Fourier, par extraction des maximums locaux du spectre et suppression des harmoniques. Ces formules permettent de déduire les périodes \mathbf{f} et \mathbf{g} de l'image.

Coordonnées polaires

Le spectre estimé par la méthode du périodogramme peut être exprimé en coordonnées cylindriques [GONZ92] :

$$\hat{P}(\rho, \theta) = \frac{1}{M \cdot N} \left| \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} W[m,n] A[m,n] e^{-2i\rho(m \cos \theta + n \sin \theta)} \right|^2 \quad (2-48)$$

Pour chaque direction θ , on définit la fonction de ρ :

$$P_{\theta}(\rho) = \hat{P}(\rho, \theta) \quad (2-49)$$

Les maximums de cette fonction indiquent avec précision les périodes dans une direction donnée θ (avec toujours un écart de $\pi/2$ entre les directions dans le plan des fréquences et dans le plan-image).

De même, pour chaque ρ , on introduit la fonction de variable θ :

$$P_{\rho}(\theta) = \hat{P}(\rho, \theta) \quad (2-50)$$

Pour une distance au centre ρ donnée, les maximums de $P_{\rho}(\theta)$ indiquent les directions des périodes.

La fonction $F(\rho) = \sum_{0 \leq \theta < \pi} P_{\theta}(\rho)$ donne de manière plus globale les normes des périodes.

La fonction $G(\theta) = \sum_{\rho=1}^D P_{\rho}(\theta)$, où D est le rayon d'un cercle centré à l'origine,

indique les directions des différentes périodicités de la texture (correspondant aux maximums).

Il faut bien entendu prendre garde aux problèmes de discrétisation posés par la représentation en coordonnées polaires.

7 Contraste associé à une région

Si les régions que l'on cherche à extraire doivent correspondre à des zones homogènes en niveau de gris (c'est le cas si on ne travaille pas sur des images fortement texturées) et présentant des transitions de niveaux de gris assez élevées vis-à-vis des régions voisines (adjacentes), on peut utiliser la notion de contraste telle qu'elle a été utilisée par Zéboudj [ZEBO88] puis reprise dans [JOUR89].

Le contraste associé à deux sites s et t , de niveaux de gris respectifs $A[s]$ et $A[t]$, est défini suivant la nature de l'image :

- pour des images de nature logarithmique (c'est-à-dire telles que le niveau de gris du pixel résulte d'une compression logarithmique de la grandeur physique fournie par le système de numérisation), on a :

$$c(s, t) = \begin{cases} \frac{|A[s] - A[t]|}{\max(A[s], A[t])} & \text{si } \max(A[s], A[t]) \neq 0 \\ 0 & \text{sinon} \end{cases} \quad (2-51)$$

- pour des images de nature non logarithmique, il vaut :

$$c(s, t) = \begin{cases} \frac{|A[s] - A[t]|}{L - 1} & \text{si } L - 1 \neq 0 \\ 0 & \text{sinon} \end{cases} \quad (2-52)$$

On peut calculer sur une région R différents contrastes :

Le *contraste intérieure* est la moyenne des contrastes locaux des points de R :

$$C_{\text{int}}(\mathbf{R}) = \frac{1}{K} \sum_{s \in R} \max(c(s, t), c \in V(s) \cap R) \quad (2-53)$$

où K est le nombre de pixels de la région R , et $V(s)$ est un voisinage du site s , par exemple un voisinage 3×3 .

Le *contraste extérieur* est la moyenne des contrastes locaux à la frontière de R . On appelle frontière de R l'ensemble des points de R connexes à une autre région que R [CHAS91].

$$C_{\text{front}}(\mathbf{R}) = \frac{1}{K_f} \sum_{s \in F} \max(c(s, t), t \in V(s), t \notin R) \quad (2-54)$$

où F est la frontière de R et K_f est son cardinal. $V(s)$ est défini comme précédemment.

Remarque : si $C_{\text{int}}(\mathbf{R})$ SYMBOL 179 \f "Symbol" $C_{\text{front}}(\mathbf{R})$ la région R n'est pas homogène.

On peut également utiliser les contrastes intérieur et extérieur pour définir le *contraste d'une région* R par :

$$C(\mathbf{R}) = \begin{cases} 1 - \frac{C_{\text{int}}(\mathbf{R})}{C_{\text{front}}(\mathbf{R})} & \text{si } C_{\text{int}}(\mathbf{R}) < C_{\text{front}}(\mathbf{R}) \\ 0 & \text{sinon} \end{cases} \quad (2-55)$$

Le *contraste associé à l'image des étiquettes* A (cf. chapitre III §2.2.1) est défini par :

$$C(A) = \frac{1}{\sum K_i} \sum_{i=1}^{N_r} K_i \cdot C(\mathbf{R}_i) \quad (2-56)$$

où K_i est le cardinal de la région \mathbf{R}_i et N_r est le nombre de régions de l'image.

On pondère par la surface de la région de façon à ne pas donner le même poids à des régions de surfaces très différentes, les petites régions devant avoir un poids plus faible. De plus, pour ne pas être trop sensible au bruit, si $C_{\text{int}}(\mathbf{R})$ est nul, on ne pose pas $C(\mathbf{R}) = 1$, mais on tient compte de la valeur de $C_{\text{front}}(\mathbf{R})$.

Ce paramètre sera utilisé pour comparer les résultats de segmentation dans le chapitre XIII.

8 Attributs fractals

Ce paragraphe contient d'abord quelques notions de base de la théorie multifractale. Les attributs issus de cette théorie pour caractériser une texture ou une région sont ensuite présentés, avec leur calcul pratique.

Remarque : il est important de noter qu'il n'est pas nécessaire que les objets soient fractals pour être caractérisés par des attributs fractals.

8.1 Éléments de théorie fractale et multifractale

Pour plus de détails, on pourra se reporter à [HENT83], [MAND82], [BARN88], [LEVY94].

Soit μ une mesure de probabilité Borélienne sur $[0,1]$, c'est-à-dire vérifiant les deux propriétés suivantes :

$$\mu(\emptyset) = 0 \text{ et}$$

$$\mu\left(\bigcup_k E_k\right) = \sum_k \mu(E_k) \text{ si } E_k \text{ sont des sous-ensembles disjoints de } [0,1]$$

Le *support* de la mesure μ est l'ensemble des points sur lesquels la mesure est définie. Soit v_n une suite croissante d'entiers positifs. On définit une suite d'intervalles $I_{i,n}$ de $[0,1]$:

$$I_{i,n} = \left[\frac{i}{v_n}, \frac{i+1}{v_n} \right], i = 0, \dots, v_n - 1. \quad (2-57)$$

8.1.1 Dimension fractale

On définit usuellement la dimension fractale (ou *dimension de Hausdorff* d'un ensemble F (notée D) de la façon suivante :

$$\forall \delta > 0, H_s^\delta(F) = \inf \left\{ \sum_{i=1}^{+\infty} |U_i|^s ; \{U_i\} \text{ est une } \delta\text{-couverture de } F \right\}$$

où $|U|$ est le diamètre de U et une δ -couverture est un recouvrement dénombrable par des ensembles de diamètre inférieur ou égal à δ . Alors :

$$D = \inf \left\{ s ; \lim_{\delta \rightarrow 0} H_s^\delta = 0 \right\} = \sup \left\{ s ; \lim_{\delta \rightarrow 0} H_s^\delta = +\infty \right\} \quad (2-58)$$

Cette définition permet de mesurer finement la dimension d'un ensemble et de mettre en évidence pour certains types d'ensembles des dimensions fractionnaires (*i.e.* non entières).

8.1.2 Dimension fractale généralisée

Si on veut caractériser non seulement la géométrie d'un ensemble, mais la répartition d'une mesure (par exemple, les niveaux de gris) définie sur celui-ci, on considère les quantités suivantes :

$$\tau_n(q) = \frac{1}{\ln v_n} \ln \sum_{\substack{0 \leq i \leq v_n \\ \mu(I_{i,n}) \neq 0}} \mu(I_{i,n})^q \quad (2-59)$$

On dit que μ possède un comportement multifractal si $\lim_{n \rightarrow +\infty} \tau_n(q) = \tau(q)$ existe pour q appartenant à un intervalle non vide de \mathbb{R} . $\tau(q)$ caractérise le comportement global de la mesure quand la taille des intervalles tend vers zéro, et est relié à une notion de dimension généralisée. En effet, si l'on définit :

$$D_q = \frac{1}{q-1} \tau(q), \quad q \neq 1 \quad \text{et} \quad D_1 = \lim_{q \rightarrow 1} \frac{1}{q-1} \tau(q) \quad (2-60)$$

alors D_0 est la dimension fractale du support de μ , D_1 est la dimension d'information, D_2 la dimension de corrélation, *etc.*

8.1.3 Spectre multifractal

On peut effectuer une autre description multifractale :

Soit $I_n(x)$ l'intervalle $I_{i,n}$ contenant x . On définit :

$$E_\alpha = \left\{ x \in [0,1[\mid \lim_{n \rightarrow +\infty} \frac{\ln \mu(I_n(x))}{\ln v_n} = \alpha \right\} \quad (2-61)$$

Les indices α caractérisent l'invariance d'échelle locale de la mesure ; si α existe au point x , alors on a $\mu(I_n(x)) \approx v_n^{-\alpha(x)}$ quand $n \rightarrow +\infty$, v_n étant la taille linéaire de l'intervalle autour de x sur lequel on évalue μ . α est appelé *exposant de Hölder* au point x . E_α peut alors être interprété comme le sous-ensemble des points ayant même comportement d'échelle, décrit par α . Pour obtenir une description multifractale de μ , on calcule d'abord l'ensemble des exposants α possibles, puis on évalue la « taille » du sous-ensemble E_α de $[0, 1[$ associé à α , en calculant sa *dimension de Hausdorff*, notée $f(\alpha)$. Cette description en terme de $(\alpha, f(\alpha))$ est ainsi à la fois locale (via α) et globale (via $f(\alpha)$). Elle est désignée sous le nom de *spectre multifractal* de μ .

8.1.4 Lien entre dimension fractale généralisée et spectre multifractal

Un problème central de la théorie multifractale est de relier les deux descriptions $(\alpha, f(\alpha))$ et $(q, \tau(q))$. Ceci a d'importantes applications. En effet, $\tau(q)$ est généralement beaucoup plus facile à calculer sur des données expérimentales que $(\alpha, f(\alpha))$: $\tau(q)$ est obtenu par un moyennage sur un grand nombre d'intervalles suivi par un passage à la limite. α est plus sensible au bruit, puisqu'il est calculé indépendamment en chaque point. En ce qui concerne $f(\alpha)$, son calcul implique l'évaluation d'une dimension de Hausdorff, tâche généralement difficile. Sous des hypothèses assez générales, on peut prouver que [BROW92] :

$$f(\alpha) \leq \inf_q \{q\alpha - \tau(q)\} \quad (2-62)$$

Pour certaines classes spéciales de mesures, qui incluent les mesures multinômiales, on a une égalité $f(\alpha) = \inf_q (q\alpha - \tau(q))$. C'est-à-dire que la dimension de Hausdorff de E_α est obtenue en effectuant une transformée de Legendre de $\tau(q)$. Dans le cas de mesures multinômiales, $D(\alpha)$ a une forme en cloche. Cette forme est aussi observée pour un grand nombre de phénomènes naturels. Cependant, elle n'est en aucun cas une propriété générale des spectres multifractals, puisque l'on peut prouver que toute fonction réglée peut être le spectre d'une mesure multifractale.

8.1.5 Lacunarité

La lacunarité est un paramètre fractal du second ordre qui permet de mesurer la géométrie d'une région R relativement au reste de l'image S :

$$\Lambda = \int_S \left(\frac{\iint_{B(s,r)} \mathbf{1}_R(r,\theta) dr d\theta}{m} - 1 \right)^2 ds \quad (2-63)$$

où :

- $B(s,r)$ est le disque de centre s et de rayon r ,
- m est le nombre moyen de points de R dans un disque de rayon r ,
- $\mathbf{1}_R$ est la fonction indicatrice sur R .

8.2 Calcul d'attributs fractals

Soit R une région dans laquelle on calcule les attributs. Dans tout ce qui suit, on supposera que la mesure utilisée est la somme des niveaux de gris de la région. On notera $A[j]$ le niveau de gris du pixel j .

8.2.1 Calcul de la dimension fractale: la méthode des boîtes

Pentland [PENT84] a montré que pour des textures homogènes, la surface d'un objet 3-D est fractale si et seulement si son image est fractale, et que la dimension fractale reste la même. La dimension fractale d'une image est comprise entre 2 et 3 ; plus la texture est lisse (resp. rugueuse), plus la dimension fractale est proche de 2 (resp. 3).

La méthode de Voss [VOSS86] [KELL89] [SARK92] a pour but d'estimer le nombre moyen, noté $N(r)$, de boîtes cubiques de côté r fixé, nécessaires pour recouvrir l'image, considérée comme une surface dans l'espace R^3 . Pour cela, on estime $P(m,r)$, la probabilité qu'une boîte de taille r , centrée sur un point arbitraire de la surface, contienne m points de l'ensemble. On a donc :

$$\forall r, \sum_{m=1}^{N_p} P(m,r) = 1 \quad (2-64)$$

où N_p est le nombre de points possibles dans le cube.

Si $N(r,m)$ est le nombre moyen de boîtes contenant m points et K le nombre de sites sur lesquels se fait le calcul (taille de la région) alors :

$$m \cdot N(r,m) = K \cdot P(m,r) \quad (2-65)$$

L'estimation du nombre moyen de boîtes disjointes nécessaires pour recouvrir la surface est :

$$N(r) = \sum_{m=1}^{N_p} N(m,r) = K \sum_{m=1}^{N_p} \frac{P(m,r)}{m} \quad (2-66)$$

L'estimation aux moindres carrés de la pente du nuage de points $(\ln(r), -\ln N(r))$, obtenu avec des boîtes de taille r croissante, donne l'estimation de la dimension fractale. L'algorithme 2.1 [KELL89] présente ce calcul.

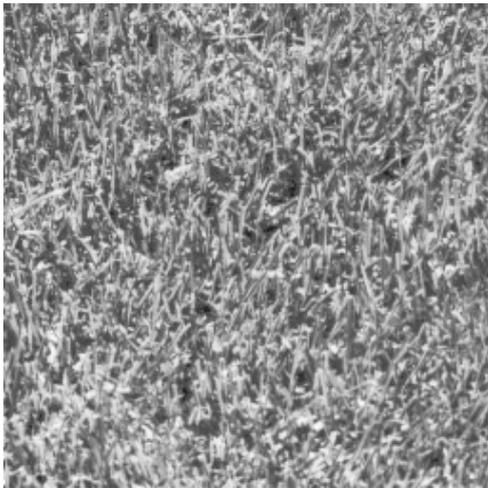
```

Initialisation :
POUR r = 1 à rmax et m = 1 à r3 FAIRE
    P(m,r) = 0
POUR tout site s de l'image FAIRE
    DEBUT
        POUR tout r = 1 à rmax FAIRE
            DEBUT
                centrer un cube de côté r sur le pixel s, A s
                compter le nombre m de pixels de l'image qui appartiennent à ce cube
                incrémenter P(m,r) de 1
            FIN
        FIN
    FIN
POUR tout r = 1 à rmax FAIRE
    N(r) =  $\sum_{m=1}^{N_p} \frac{P(m,r)}{m}$ 
Estimer par la méthode des moindres carrés la pente D de la courbe  $(\ln(r), -\ln N(r))$ 

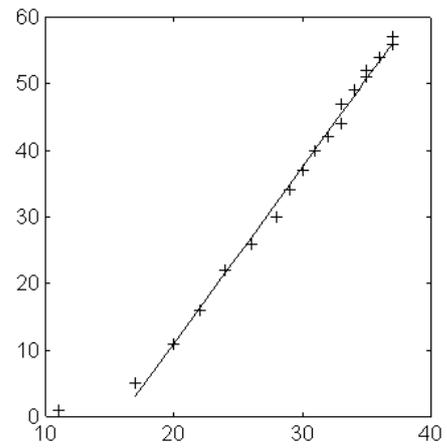
```

Algorithme 2.1 : Estimation de la dimension fractale par la méthode des boîtes

Cette méthode, appliquée sur la texture HERBE de Brodatz [BROD66] avec r variant de 3 à 41 (par pas de 2) fournit la courbe présentée figure 2.8. La dimension fractale estimée par cette méthode est de 2,66, si on ne prend pas en compte le premier point.



Texture HERBE (256 x 256) [Brodatz D9]



Position des points $(\ln(r), -\ln N(r))$ et droite de régression linéaire

Figure 2.8 : Estimation de la dimension fractale par la méthode des boîtes

D'autres méthodes [GANG86] [PENT86] [SARK92] ont été proposées pour estimer la dimension fractale, elles donnent des résultats sensiblement différents sur les mêmes textures, bien que classant *grosso modo* les textures dans le même ordre [SARK92].

Sarkar et Chaudhuri [SARK92] calculent $N(r)$ de la façon suivante. On découpe la fenêtre F de taille $n \times n$ en un ensemble de fenêtres f sans recouvrement de taille $d \times d$, où d est un entier tel que $1 < d \leq n/2$. Sur chacune de ces fenêtres, on calcule le nombre $n_r(f)$ de boîtes de taille $d \times d \times d$ nécessaires pour contenir la portion de la surface correspondant à f . $n_r(f)$ n'est autre que l'étendue augmentée de 1 de l'image A dans la fenêtre f :

$$n_r(f) = \max_f \{A[s]\} - \min_f \{A[s]\} + 1 \quad (2-67)$$

On estime $N(r)$ en sommant sur toutes les fenêtres f :

$$N(r) = \sum_f n_r(f) \quad (2-68)$$

En faisant croître d , c'est-à-dire le facteur d'échelle r , on estime $N(r)$ pour différentes échelles. On estime D comme précédemment dans un repère $(\ln(1/r), \ln(N(r)))$.

8.2.2 Calculs du spectre multifractal

Le calcul du spectre se fait en deux étapes :

- Pour chaque site s , on calcule $\alpha(s)$ dans un voisinage de taille k autour de ce site. On obtient ainsi un nuage de points $\left(-r \ln 2, \ln \sum_{j \in \nu_r(s)} A[j] \right)$, où $\nu_r(s)$ est le voisinage de taille r autour de s , avec r variant de 1 à k . Il est important de garder k petit (pas plus de 7) afin de conserver l'aspect local de α . L'estimation aux moindres carrés de la pente du nuage de points donne l'estimation de $\alpha(s)$.
- Dans la sous-image F , l'ensemble des $\alpha(s)$ est discrétisé en intervalles $I_\alpha = [\alpha - \varepsilon, \alpha + \varepsilon[$. $E_\alpha = \{s \in F, \alpha(s) \in I_\alpha\}$ est donc l'ensemble des points ayant un exposant de Hölder α proche.

On obtient $D(\alpha)$ en calculant la dimension fractale de E_α (par la méthode de Voss).

Le spectre $(\alpha, f(\alpha))$ contient un grand nombre d'informations. Les α donnent une mesure locale des singularités, $f(\alpha)$ mesure le comportement fractal des points de même α . Il a généralement une forme de cloche.

On peut extraire du spectre plusieurs attributs. Voici les plus intuitifs. Tout d'abord sur α :

- α_{\min} : minimum de singularité.
- α_{\max} : maximum de singularité.
- α^* : singularité la plus fréquente (α tel que $f(\alpha)$ soit maximum).

Puis sur le spectre lui-même :

- $f(\alpha^*)$: dimension fractale des singularités les plus fréquentes.
- aire de la courbe ($\alpha, f(\alpha)$).

8.2.3 Calcul des dimensions fractales généralisées

Dans la sous-image \mathbf{F} , on considère un recouvrement du support de la mesure (l'ensemble des sites) par des boîtes $\mathbf{B}_r(s)$ de taille 2^{-r} . Soit $p_r(s) = \sum_{j \in \mathbf{B}_r(s)} A_j$ la mesure portée par la boîte $\mathbf{B}_r(s)$. $\tau(q)$ est obtenue par l'estimation aux moindres carrés de la pente du nuage de points $\left(-r \ln 2, \ln \sum_{s \in \mathbf{F}} p_r(s)^q \right)$.

D_q peut être vue comme le moment d'ordre q de la mesure μ . Quand q est grand positivement (resp. négativement), D_q mesure les zones de l'image où la masse est très concentrée (resp. diffuse).

8.2.4 Lacunarité

La dimension fractale ne suffit en général pas à caractériser une texture, même si celle-ci est fractale [MAND68]. Il faut souvent lui ajouter l'attribut de lacunarité [LEVY90], qui est faible quand la texture est fine et forte pour une texture grossière.

Elle peut se calculer de deux manières :

- soit en appliquant la formule (2-63), en faisant une binarisation préalable de l'image par seuillage,

$$\Lambda(r) = \sum_{s \in \mathcal{S}} \left(\frac{\sum_{i \in R \cap \mathbf{B}(s,r)} A[i]}{m} - 1 \right)^2$$

où m est la moyenne des niveaux de gris de l'image.

- soit à partir des probabilités $P(m,r)$ (cf. § 8.2.1) [VOSS86] [KELL89] :

$$\Lambda(r) = \frac{M^2(r) - (M(r))^2}{(M(r))^2} \quad (2-69)$$

$$\text{avec } M(r) = \sum_{m=1}^{N_p} m \cdot P(m,r) \text{ et } M^2(r) = \sum_{m=1}^{N_p} m^2 \cdot P(m,r) \quad (2-70)$$

Conclusion : En pratique, les attributs fractals permettent souvent de bien caractériser les textures. Cependant, ils ne peuvent se calculer que sur des fenêtres ou des régions de taille importante et leur temps de calcul est élevé.

9 Attributs surfaciques

9.1 Courbures

Si l'on modélise l'image comme une surface dans l'espace de dimension 3, on peut étudier localement cette surface en utilisant la géométrie différentielle.

Dans le cas de surfaces « lisses », c'est-à-dire telles que toutes les dérivées partielles secondes existent et sont continues, on utilise les deux formes fondamentales L_1 et L_2 pour calculer des attributs de courbure.

Si un point de la surface est défini par sa position $U = (x, y, z)$ où x et y sont les coordonnées dans l'image et z est le niveau de gris au site (x, y) , la *première forme fondamentale* est :

$$L_1 = dU \cdot dU = E' dx^2 + 2F' dx dy + G' dy^2 \quad (2-71)$$

$$\text{avec } E' = 1 + \left(\frac{\partial z}{\partial x} \right)^2, \quad F' = \frac{\partial z}{\partial x} \frac{\partial z}{\partial y}, \quad G' = 1 + \left(\frac{\partial z}{\partial y} \right)^2 \quad (2-72)$$

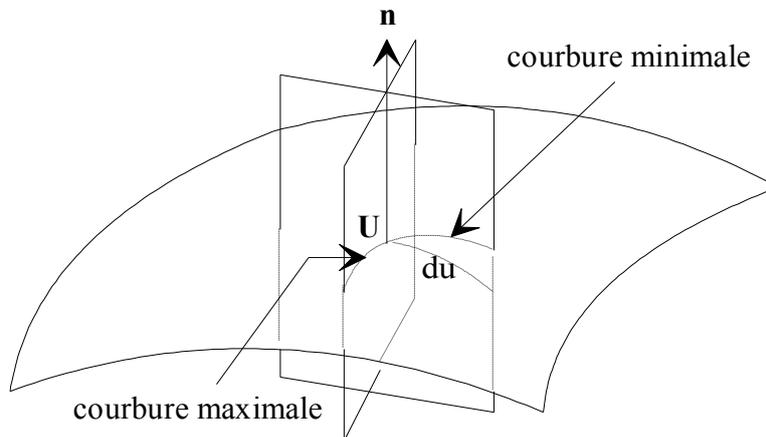


Figure 2.9 : Surface lisse et directions des courbures maximale et minimale en un point U

L_1 représente l'élément de longueur d'arc du des courbes tracées sur la surface et passant par le point (x,y,z) . Il ne dépend pas des translations et rotations effectuées sur la surface ; c'est donc un attribut intrinsèque de la surface.

La *seconde forme fondamentale* représente la courbure des courbes tracées sur la surface (cf. figure 2.9). Si \mathbf{n} est le vecteur normal à la surface, elle est définie par :

$$L_2 = \mathbf{dU} \cdot \mathbf{dn} = e' dx^2 + 2f' dx dy + g'dy^2 \quad (2-73)$$

$$\text{avec } e' = \frac{1}{\Delta'} \frac{\partial^2 z}{\partial x^2}, \quad f' = \frac{1}{\Delta'} \frac{\partial^2 z}{\partial x \partial y}, \quad g' = \frac{1}{\Delta'} \frac{\partial^2 z}{\partial y^2} \quad (2-74)$$

$$\text{et } \Delta' = \sqrt{E' G' - F'^2} \quad (2-75)$$

L_2 est un attribut extrinsèque de la surface, il dépend de la position de celle-ci dans l'espace.

La fonction $k = L_2/L_1$ est la courbure normale en un point U de la surface. Les extrémums de cette fonction en un point définissent les *courbures principales* k_1 et k_2 .

À partir de k_1 et k_2 , on définit :

- la *courbure gaussienne* :

$$K_1 = k_1 \cdot k_2 \quad (2-76)$$

- la *courbure moyenne* :

$$K_2 = (k_1 + k_2) / 2 \quad (2-77)$$

Les courbures principales sont associées à des directions ; selon leur signe, la surface a l'une des six formes : bosse, creux, vallée, plateau, selle, crête [BESL86]. Par contre, selon les signes des courbures moyenne et gaussienne, la surface peut prendre huit formes, les formes en selle étant elles-mêmes de 3 types différents.

Les courbures moyenne et gaussienne sont invariantes en rotation et en translation et la courbure gaussienne est de plus, stable par changement d'échelle.

À partir des courbures principales, Peet et Sahota [PEET85] proposent d'autres attributs de courbure tels que le maximum des valeurs absolues de k_1 et k_2 , ...

9.2 Quadriques

On peut étudier plus précisément la surface si on l'approxime localement par une surface polynômiale du second degré (quadrique).

L'équation générale d'une quadrique est :

$$q(x,y,z) = ax^2+by^2+cz^2+2dxy+2eyz+2fzx+2gx+2hy+2iz+j=0 \quad (2-78)$$

Levin [LEVI79] donne une classification des quadriques selon les valeurs des dix coefficients.

Si on suppose que l'un des axes d'inertie est dans la direction des niveaux de gris (Oz), l'équation (2-78) s'écrit dans le repère des axes d'inertie :

$$q'(x,y,z) = \lambda x^2 + \mu y^2 + cz^2 + g'x + h'y + 2iz + j = 0 \quad (2-79)$$

où λ et μ sont les valeurs propres associées aux deux directions différentes de (Oz).

La plupart des auteurs [YOKO89] [PEET85] [PHIL94] se limitent aux seules formes biquadratiques, parce qu'une image est une fonction $z = f(x, y)$ et donc à tout site (x, y) , il correspond au plus un z . Dans ce cas, l'équation (2-78) devient :

$$p(x,y) = ax^2 + by^2 + 2dxy + 2gx + 2hy + j \quad (2-80)$$

Le calcul des 6 coefficients a , b , d , g , h et j de la forme biquadratique approximant au mieux la surface dans un voisinage autour du pixel (x, y) se fait par convolution de l'image avec les 6 masques de la taille voulue de ce voisinage.

À titre d'exemple, voici les masques sur des voisinages de taille 3×3 et 5×5 . **A** est le masque pour calculer le coefficient a , **D** pour le coefficient d , **G** pour le coefficient g , **J** pour le coefficient j . b se calcule avec la matrice transposée de **A** et h avec la matrice transposée de **G**.

voisinage 3×3 :

$$\mathbf{A} = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix} \quad \mathbf{D} = \frac{1}{8} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{G} = \frac{1}{12} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{J} = \frac{1}{9} \begin{bmatrix} -1 & 2 & -1 \\ 2 & 5 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

voisinage 5×5 :

$$\mathbf{A} = \frac{1}{70} \begin{bmatrix} 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \\ 2 & -1 & -2 & -1 & 2 \end{bmatrix} \quad \mathbf{D} = \frac{1}{200} \begin{bmatrix} 4 & 2 & 0 & -2 & -4 \\ 2 & 1 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -2 & 0 & 2 & 4 \end{bmatrix}$$

$$\mathbf{G} = \frac{1}{100} \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix} \quad \mathbf{J} = \frac{1}{175} \begin{bmatrix} -13 & 2 & 7 & 2 & -13 \\ 2 & 17 & 22 & 17 & 2 \\ 7 & 22 & 27 & 22 & 7 \\ 2 & 17 & 22 & 17 & 2 \\ -13 & 2 & 7 & 2 & -13 \end{bmatrix}$$

Les masques de taille 7×7 et 9×9 sont donnés dans [PHIL94].

10 Attributs géométriques

R étant une région (connexe) de l'image dont les pixels ont pour coordonnées (x_i, y_j) , on peut disposer des attributs suivants :

- aire $S(\mathbf{R}) = K$ ou nombre de pixels de la région

- *périmètre* $P(\mathbf{R})$; il est en général calculé comme la somme des distances entre pixels consécutifs du contour (somme de 1 et de $\sqrt{2}$). Le résultat est sensiblement différent s'il est calculé à partir du contour intérieur ou extérieur de la région.

- *centre de gravité* G , de coordonnées :

$$x_G = \frac{1}{K} \sum_{i=1}^K x_i \text{ et } y_G = \frac{1}{K} \sum_{i=1}^K y_i \quad (2-81)$$

- *diamètre* de la région :

$$D = \max \{d(s,t) \mid s,t \in \mathbf{R}\} = \max \{d(s,t) \mid s,t \in F(\mathbf{R})\} \quad (2-82)$$

$F(\mathbf{R})$ désignant la frontière de \mathbf{R} et $d(s, t)$ la distance entre les sites s et t (par exemple, la distance euclidienne).

- la *compacité* (ou facteur de circularité) :

$$\frac{4\pi S(\mathbf{R})}{P^2(\mathbf{R})} \quad (2-83)$$

Il est proche de 1 si la forme de \mathbf{R} avoisine celle d'un disque et proche de 0 si le contour est très découpé ou si la forme est très allongée.

- *direction principale d'inertie*, premier vecteur propre de la matrice d'inertie

$$\begin{bmatrix} a & c \\ c & b \end{bmatrix} \text{ avec :}$$

$$a = \frac{1}{K} \sum_{i=1}^K (x_i - x_G)^2 = \frac{1}{K} \sum_{i=1}^K x_i^2 - x_G^2 \quad (2-84)$$

$$b = \frac{1}{K} \sum_{i=1}^K (y_i - y_G)^2 = \frac{1}{K} \sum_{i=1}^K y_i^2 - y_G^2 \quad (2-85)$$

$$c = \frac{1}{K} \sum_{i=1}^K (x_i - x_G)(y_i - y_G) = \frac{1}{K} \sum_{i=1}^K x_i y_i - x_G y_G \quad (2-86)$$

La direction principale de la région \mathbf{R} est alors définie par son angle α avec l'axe des x :

$$\text{tg}2\alpha = \frac{2c}{a-b} \quad (2-87)$$

- *rectangle d'encadrement* de \mathbf{R} :

rectangle dont les côtés sont parallèles aux vecteurs propres de la matrice centrale d'inertie de \mathbf{R}

- *parallélogramme d'encadrement* de R

À partir du polygone approximant les contours de R , on calcule les deux directions cumulées des côtés les plus fréquentes, c'est-à-dire celles pour lesquelles le produit vectoriel est maximum. Le parallélogramme d'encadrement est le parallélogramme circonscrit à R dont les côtés sont parallèles à ces deux directions [PHIL93].

- *élongation* ou rapport entre la longueur et la largeur du rectangle d'encadrement de R .
- *rapport entre la surface de la région et celle de son enveloppe convexe*
- *nombre de connexité* ou nombre d'Euler $N(R)$:

Pour une région connexe, le nombre d'Euler est égal à 1 diminué du nombre de trous de R . Plus généralement pour une image binaire, il est égal au nombre de composantes connexes diminué du nombre total de trous. Le nombre de connexité se calcule aisément par une transformation morphologique binaire (*cf.* annexe C) avec deux éléments structurants en maille hexagonale et trois en maille carrée.

Pour une image binaire A (le support de la région R par exemple) :

en maille hexagonale, $N(A) = \text{aire}(A \otimes S_1) - \text{aire}(A \otimes S_2)$ avec :

$$S_1 = \begin{pmatrix} 0 & 0 \\ & 1 \end{pmatrix} \text{ et } S_2 = \begin{pmatrix} 0 & \\ 1 & 1 \end{pmatrix}$$

en maille carrée, $N(A) = \text{aire}(A \otimes S_3) + \text{aire}(A \otimes S_4) - \text{aire}(A \otimes S_5)$ avec :

$$S_3 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, S_4 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ et } S_5 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

11 Attributs inter-régions

On peut étudier les relations entre une région R et les régions qui lui sont adjacentes pour classifier. Pour cela on ajoute à la liste des attributs d'une région R le nombre de régions adjacentes et le nombre de régions incluses. Comme nous le verrons dans le chapitre III, les différentes régions d'une image peuvent être représentées par un graphe dans lequel une arête traduit une adjacence entre deux régions. Un vecteur d'attributs est associé à chaque arête, il caractérise la relation de voisinage entre régions adjacentes. Ce vecteur peut comprendre :

- une information sur les positions relatives des régions,
- des attributs sur la frontière entre les deux régions :
la longueur, le gradient moyen sur la frontière.

12 Conclusion

Le choix des attributs pour caractériser une région est un problème difficile qui demande toute l'expérience du traiteur d'images. Ce choix dépend de la nature de l'image ou de la zone de l'image, du problème à résoudre (segmentation, reconnaissance de formes, ...) et du type de méthode employée pour résoudre le problème.

Quand on cherche à différencier deux objets d'une scène, le paramètre le plus discriminant est, dans la majorité des cas, la moyenne des niveaux de gris. Lorsque celle-ci ne suffit pas pour délimiter les objets ou classifier une région, il faut faire appel à des paramètres liés aux statistiques locales. Les attributs les plus fréquemment utilisés - et qui servent de référence à toute nouvelle méthode de caractérisation de texture - sont ceux extraits des matrices de cooccurrence.

Le paragraphe 2.3 du chapitre X présente des méthodes pour sélectionner les attributs les plus discriminants pour un problème donné. La méthode des paires est employée dans le système ARTHUR, développé par l'INRIA pour déterminer les attributs les plus performants parmi un vaste ensemble d'attributs englobant les attributs présentés dans ce chapitre (en faisant varier leurs paramètres) : les attributs issus des matrices de cooccurrence, ainsi que les attributs fractals sont souvent les plus performants [LECH91]. On verra un exemple d'application dans le chapitre XIII.

Par ailleurs, il existe peu de méthodes pour extraire d'une texture la primitive et la règle de placement. Pour les textures périodiques, on peut aisément déterminer la règle de placement (les périodes). Dans le cas d'une texture poissonnienne (*i.e.* dont la loi de placement des primitives est poissonnienne), des méthodes employant la morphologie mathématique permettent de calculer différents attributs moyens de la primitive, tels que le volume la surface volumique, *etc.* [PRET87] [PRET91] [PHIL88] [PHIL89].

Les primitives sont en fait rarement extraites ; dans certains cas particuliers, tels que les textures issues de la numérisation d'objets ovoïdes (grains d'une radiographie, noyaux cellulaires, par exemple), on peut l'approximer par une quadrique [PHIL88] [PHIL94]. Une transformation par chapeau haut-de-forme (*cf.* annexe C) peut également conduire à l'extraction des primitives. Les textures naturelles restent dans l'ensemble, difficiles à modéliser.

CHAPITRE III

Structures de données et algorithmes associés

1 Introduction

Le chapitre I a défini l'image comme une représentation planaire de la réalité élaborée à partir de capteurs d'origines diverses. Pour être analysée par l'ordinateur, l'image doit être convertie en données numériques. Cette digitalisation est dépendante de la nature du capteur et peut être très complexe. Quoi qu'il en soit, il n'est pas abusif de considérer que l'image numérique est un signal bidimensionnel discret. En conséquence, le niveau de représentation élémentaire dépend de la manière de partitionner finement le plan (ou de l'échantillonner (cf. § 2.1)). A un second niveau, on effectue des partitionnements du plan en *régions*, regroupant (en général) plusieurs pixels. Cet aspect est abordé au paragraphe 2.2 qui porte sur les partitions irrégulières. Différentes partitions du plan peuvent enfin être associées de manière hiérarchique en fonction de leur finesse d'analyse ; on obtient alors les représentations arborescentes et pyramidales du paragraphe 3.

La richesse du contenu sémantique est liée au niveau de représentation. Au niveau élémentaire, l'élément de base est le pixel dont le contenu sémantique est pauvre. Ce dernier s'enrichit pour le second niveau qui traduit des relations entre régions regroupant plusieurs pixels. Les représentations hiérarchiques sont *a priori* les plus significatives dans la mesure où elles associent des représentations de deuxième niveau.

Les lignes et les frontières de régions sont également des informations pertinentes. Là aussi, le premier niveau de représentation est le pixel qui doit être marqué d'une manière particulière. Il est nécessaire de caractériser les ensembles connexes de pixels de contours par des représentations de deuxième niveau au contenu sémantique plus riche. Les structures associées sont essentiellement les chaînes et les lignes polygonales ; elles sont présentés au paragraphe 4.

Chaque représentation doit être manipulable par l'ordinateur, on y associe donc des structures de données de type tableaux, arbres, graphes et pyramides de graphes pour lesquelles il existe des algorithmes classiques de traitement disponibles dans des ouvrages spécialisés [AHO95].

Ce chapitre décrit les structures employées à chaque niveau pour des applications telles que la segmentation en régions. Quelques algorithmes de base,

nécessaires à la création des structures manipulées sont présentés. Les notions de base sur les graphes sont en outre présentées dans l'annexe A.

2 Partitions élémentaires d'une image et structures associées

2.1 Partitions élémentaires

2.1.1 Pavage, maillage d'une image

a) Pavage

Un *pavage* est une partition du plan, utilisant toujours les mêmes formes élémentaires. Les éléments constituant un pavage sont appelés *tesselles*.

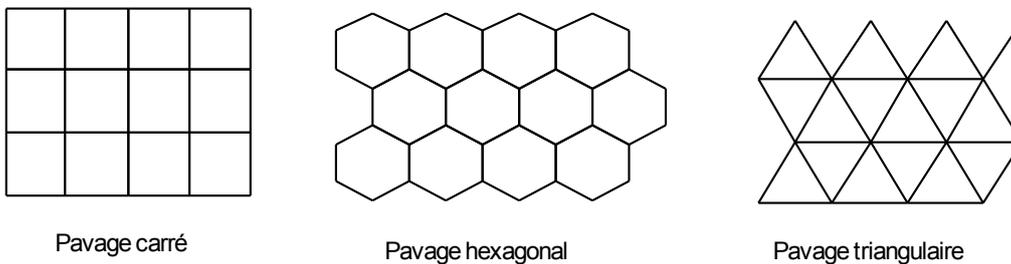


Figure 3.1 : Les trois types de pavage utilisés en imagerie.

Le nombre de pavages est illimité. En pratique, on impose des contraintes sur la géométrie des tesselles utilisées : ce seront des polygones convexes réguliers. De plus, pour des raisons de simplicité de codage, seuls sont utilisés dans la pratique, les pavages formés par la répétition d'une seule forme de tesselle n'autorisant que les connexions par les sommets. Il n'existe alors que trois types de pavages : rectangulaire, hexagonal et triangulaire (*cf.* figure 3.1). De ces trois pavages, le pavage rectangulaire (ou carré) est le plus utilisé, suivi de l'hexagonal. Le pavage triangulaire, peu pratique, est rarement employé [CHAS91].

La notion de récursivité du pavage d'une image est particulièrement importante pour les algorithmes de segmentation en régions. Un pavage est dit *récursif*, si les tesselles peuvent être regroupées (resp. décomposées) en un ensemble de tesselles identiques mais de taille plus grande (resp. plus petite).

Un pavage récursif présente l'intérêt de s'adapter à des analyses en multirésolution spatiale (changement d'échelle). Parmi les trois pavages proposés, seul le pavage hexagonal n'est pas récursif. Il est toutefois possible de regrouper les tesselles hexagonales, pour approcher cette notion de récursivité : les groupements obtenus ne sont plus des tesselles régulières, mais ils permettent de conserver l'information de changement d'échelle. La notion de récursivité est utilisée pour les « *quadrees* » (*cf.* § 3.1) ainsi que dans certains types de pyramides (*cf.* § 3.2) [JOLI90].

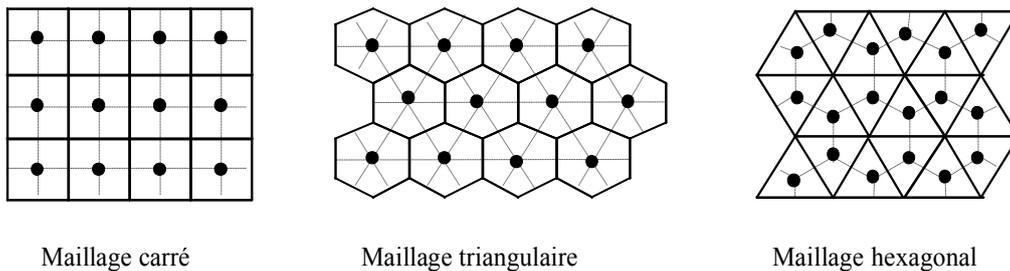
b) Maillage

A tout pavage du plan est associé un maillage.

Un *maillage* est un *graphe*² non-orienté dont les *nœuds* sont les *sites*. Tout site est associé à une tesselle du pavage. En général ce site correspond au centre de gravité de la tesselle associée. Une *arête* du graphe relie deux sites dont les tesselles associées sont adjacentes.

Il y a dualité entre maillage et pavage : à un pavage carré correspond un maillage carré, au pavage hexagonal correspond une maille triangulaire, au pavage triangulaire est associé le maillage hexagonal. On retrouvera aussi une dualité entre les polygones de Voronoï et les partitions de Delaunay (cf. figure 3.2 et § 2.2.2).

Le maillage est obtenu de manière implicite par échantillonnage de l'image analogique lors de la numérisation. Le pixel prend une valeur numérique qui intègre les valeurs de luminance sur un petit voisinage du site, le graphe est alors *valué*. La plupart des capteurs échantillonne en maillage carré (ou rectangulaire), le maillage triangulaire et le maillage hexagonal peuvent être simulés (cf. § 2.1.2.b).



En trait plein : limites des tesselles des pavages carré, hexagonal et triangulaire
En pointillé : maillages correspondants ; les points représentent les sites

Figure 3.2 : Maillages utilisés en imagerie

2.1.2 Voisinages et métriques associées

Les définitions de voisinages et les métriques associées sont liées au type de maillage.

a) Cas de la maille carrée

Soit l'image discrète considérée comme un sous-ensemble borné \mathcal{S} de \mathbb{N}^2 . Chaque point P du maillage est un *site* dont les coordonnées $[i_p, j_p]$ sont des entiers ; $i_p \in \{1, \dots, M\}$ est l'indice de ligne et $j_p \in \{1, \dots, N\}$ est l'indice de colonne.

Les deux métriques couramment utilisées en maillage carré sont désignées par d_4 et d_8 .

² Les notions de base sur les graphes sont rappelées en annexe A

La métrique d_4 (*city block metric*) est définie par :

$$d_4(P, Q) = |i_p - i_q| + |j_p - j_q| \quad (3-1)$$

Elle permet d'associer à P, un ensemble de points $\nu_4(P)$ appelé voisinage et défini par :

$$\nu_4(P) = \{Q \in \mathcal{S}, d_4(P, Q) \leq 1\} \quad (3-2)$$

Ce voisinage est représenté sur la figure 3.3. Le point P possède 4 voisins adjacents situés à une distance $d_4 = 1$. On dit que ν_4 est un *voisinage 4-connecte*.

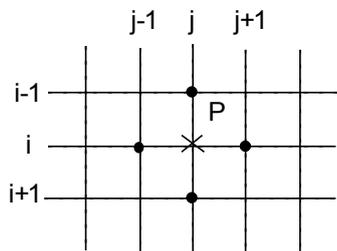


Figure 3.3 : Voisinage en 4-connectivité

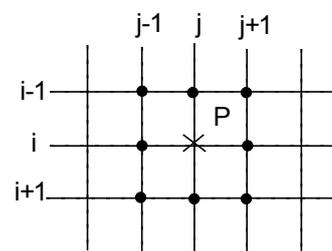


Figure 3.4 : Voisinage en 8-connectivité

La métrique d_8 (*chessboard metric*) est définie par :

$$d_8(P, Q) = \max(|i_p - i_q|, |j_p - j_q|) \quad (3-3)$$

Elle permet d'associer à P, un voisinage $\nu_8(P)$ défini par :

$$\nu_8(P) = \{Q \in \mathcal{S}, d_8(P, Q) \leq 1\} \quad (3-4)$$

Ce voisinage est représenté sur la figure 3.4. Le point P possède 8 voisins dont quatre d'entre eux correspondent à ν_4 . Les quatre autres voisins diagonaux sont situés à une distance $d_8 = 1$. On dit que ν_8 est un *voisinage 8-connecte*.

Les notions de voisinage présentées ci-dessus doivent être prise au sens de l'adjacence de sommets dans un graphe et non au sens topologique [CHAS91].

b) Simulation du maillage hexagonal à partir du maillage carré

La maille hexagonale fournit un voisinage $\nu_6(P)$ isotrope. La simulation du maillage hexagonal sur un maillage carré se fait en prenant les 6 sites voisins de $[i, j]$ de manière différente suivant que l'indice de colonne j est pair ou impair (*cf.* figure 3.5) :

$[i-1, j], [i, j-1], [i, j+1], [i+1, j-1], [i+1, j], [i+1, j+1]$ si j est pair,
 ou
 $[i-1, j-1], [i-1, j], [i-1, j+1], [i, j-1], [i, j+1], [i+1, j]$ si j est impair.

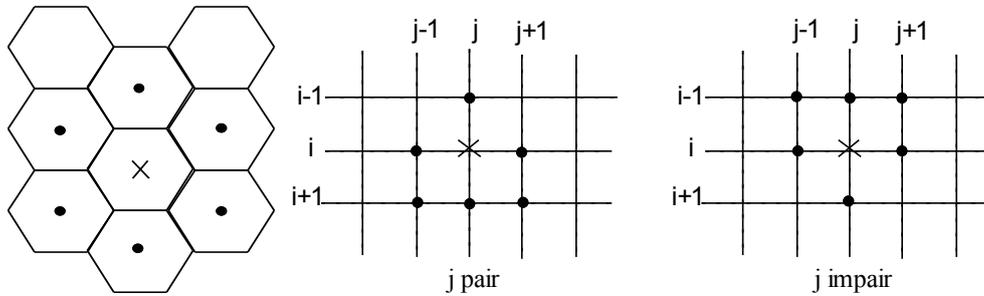


Figure 3.5 : Voisinage $v_6(P)$ défini sur un maillage hexagonal et simulation du maillage hexagonal sur une maille carrée

2.2 Partitions irrégulières

2.2.1 Graphes d'adjacence de régions

La segmentation est une partition de l'image en ensembles connexes de pixels appelés *régions*.

On désigne par *image d'étiquettes* une image où les pixels d'une même région ont tous la même valeur appelée étiquette de la région. Une image d'étiquettes peut s'obtenir par extraction des composantes connexes à partir d'une image de contours fermés ou à partir d'une image de pixels classifiés (cf. § 5). A une partition de l'image, on associe un graphe d'adjacence des régions. Une telle structure de données permet d'élever le niveau de représentation de l'image en passant du pixel à la région. Elle est notamment très utile pour une phase d'interprétation qui ferait suite à la segmentation.

Un *graphe d'adjacence* est un graphe non-orienté (cf. annexe A) dont les nœuds représentent les régions formant une partition de l'image. Deux nœuds sont reliés par une arête si les deux régions sont adjacentes, c'est-à-dire s'il existe au moins deux pixels voisins, appartenant chacun à l'une des deux régions. Le graphe d'adjacence dépend donc de la connexité choisie (cf. figure 3.6). Il est planaire pour un voisinage en 4-connexité, il ne l'est plus en 8-connexité, ceci conduit à des adjacences parfois gênantes telles que celles entre les régions 0 et 5 sur la figure 3.6.

Une segmentation obtenue par une détection de contour conduit à un autre type de partition de l'image : une partition en deux classes, les pixels de contour et les autres. La partition en régions s'obtient alors à partir des contours fermés : pour résoudre les problèmes de voisinage liés à la connexité, on emploie en général la 4-connexité pour les régions et la 8-connexité pour les contours. Par exemple, sur la figure 3.6 les zones étiquetées 1 à 4 peuvent être considérées comme les

contours (8-connexes) d'une région 5 (4-connexes). Les régions 0 et 5 ne sont alors pas adjacentes.

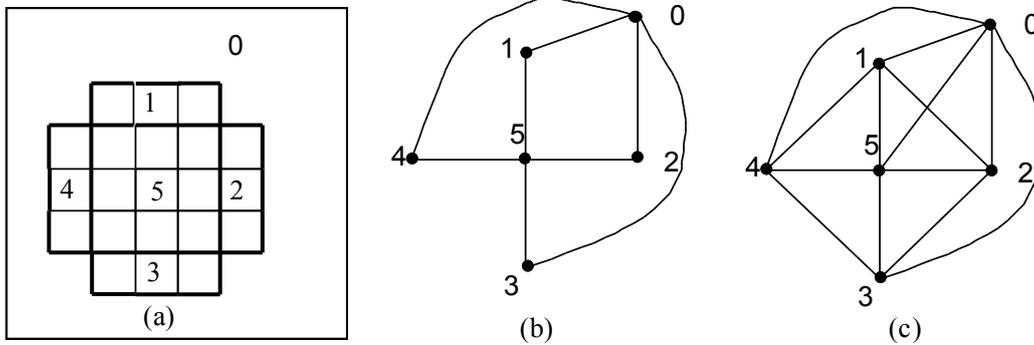
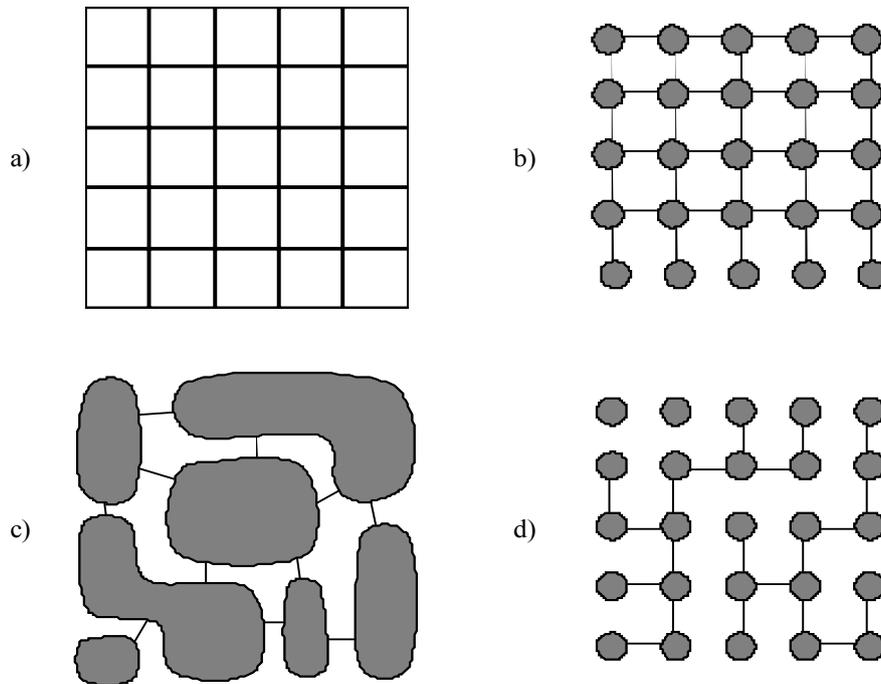


Figure 3.6 : (a) Partition en 6 régions (traits gras),
(b) graphes d'adjacence en 4-connexité et (c) en 8-connexité



a) pavage initial de l'image b) graphe d'adjacence équivalent, en utilisant la notion de voisinage entre les pixels c) graphe d'adjacence pour une représentation en région d) arbre, graphe partiel de b)

Figure 3.7 : Exemples de la représentation d'une image en graphe.

Si chaque élément de la partition en régions est réduit à un seul pixel, le graphe d'adjacence correspond au maillage (cf. figure 3.7.b).

Le graphe d'adjacence est en général valué. Les paramètres que l'on peut associer à chaque région sont définis dans le chapitre II. On peut également associer des paramètres aux arêtes du graphe.

Il existe d'autres types de graphes pour représenter une image ; par exemple, le graphe dual dont on verra une illustration avec le diagramme de Voronoï et son dual, le graphe de Delaunay (*cf.* § 2.2.2).

Des arbres peuvent également être employés pour représenter l'image (*cf.* le « *quadtree* » § 3.1.1) ou pour la parcourir (*cf.* parcours de Peano, § 3.1.2).

2.2.2 Diagramme de Voronoï

Le diagramme de Voronoï est une partition de l'image, construite à partir de germes et utilisant différents types de distances [PREP88]. Les germes peuvent être des points ou des objets.

a) Diagrammes de Voronoï ponctuels

Si on prend des germes ponctuels et la distance euclidienne, la partition obtenue est constituée de polygones appelés *polygones de Voronoï*.

Définition et propriétés

Soit $G = \{P_1, P_2, \dots, P_n\}$ un ensemble de n points (appelés germes) du plan euclidien \mathbb{R}^2 .

Pour chaque point P de G , la région polygonale composée des points plus proches du germe P que de tous les autres germes est appelée polygone de Voronoï et notée $V(P)$.

Ces polygones sont convexes ; certains ne sont pas bornés (*cf.* figure 3.8).

Le graphe planaire formé par les frontières des polygones constitue le *diagramme de Voronoï ponctuel* de G ; les arêtes sont des portions de médiatrices, les nœuds sont équidistants d'au moins trois germes.

Quelques propriétés du diagramme de Voronoï [KLEI89] sont rappelées ci-après :

Cas où il n'existe pas de quadruplet de points de G cocirculaires

Dans ce cas particulier, tout sommet de Voronoï est équidistant de trois germes exactement. Tout sommet du diagramme de Voronoï est alors centre d'un cercle, appelé cercle de Delaunay, (*cf.* figure 3.9) passant par trois germes et ne contenant aucun autre germe. La triangulation obtenue en joignant les triplets de sommets est appelée triangulation de Delaunay.

La dualité entre diagramme de Voronoï et triangulation de Delaunay est illustrée en figure 3.9.

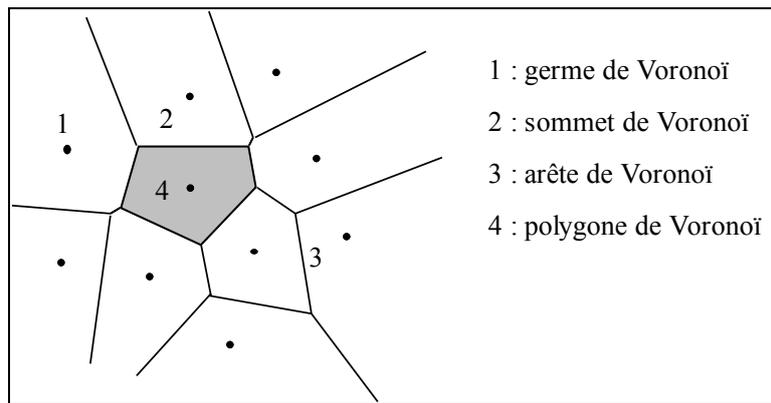


Figure 3.8 : Diagramme de Voronoï ponctuel

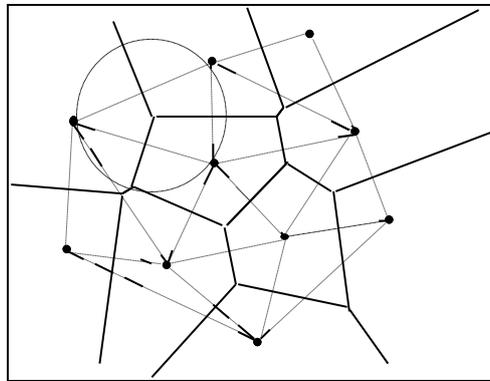


Figure 3.9 : Diagramme de Voronoï et triangulation de Delaunay associée, construits à partir de germes ponctuels. Un cercle de Delaunay est tracé.

Le diagramme de Voronoï est un graphe régulier d'ordre 3 car chaque sommet du diagramme possède exactement 3 voisins (éventuellement rejetés à l'infini dans le cas des bords de l'image).

Cas général

Tout sommet du diagramme de Voronoï plus proche du germe P dans l'ensemble G appartient à une arête du polygone $V(P)$.

$V(P)$ est non borné si et seulement si P appartient à la frontière de l'enveloppe convexe construite sur G .

b) Structures de données associées aux diagrammes de Voronoï

Le choix d'une bonne structure de données est important pour l'efficacité des algorithmes de construction et d'exploitation d'un diagramme de Voronoï.

Un polygone est associé à un germe unique. On peut lui affecter une étiquette qui permet un adressage direct dans une structure de tableau où chaque cellule est un couple germe-polygone.

Représentation d'un polygone

Un polygone est caractérisé par un *germe* et une *région* du plan. Le polygone sera représenté par une structure à deux champs : l'un codant le germe, l'autre codant la frontière polygonale. Représenter cette dernière par une *liste chaînée d'arêtes* est tout indiqué, car on ne connaît pas *a priori* le nombre d'arêtes associées à un germe (cf. figure 3.10).

Il est plus intéressant de mémoriser une liste d'arêtes qu'une liste de sommets. En effet, une arête de Voronoï contient plus d'informations car elle sépare deux germes voisins, et elle facilite les calculs (par exemple, pour le passage au dual et la construction de la triangulation).

Une arête est représentée par :

- ses deux extrémités,
- l'étiquette du germe voisin,
- l'arête suivante et l'arête précédente définies, par exemple, suivant un parcours dans le sens trigonométrique de la frontière.

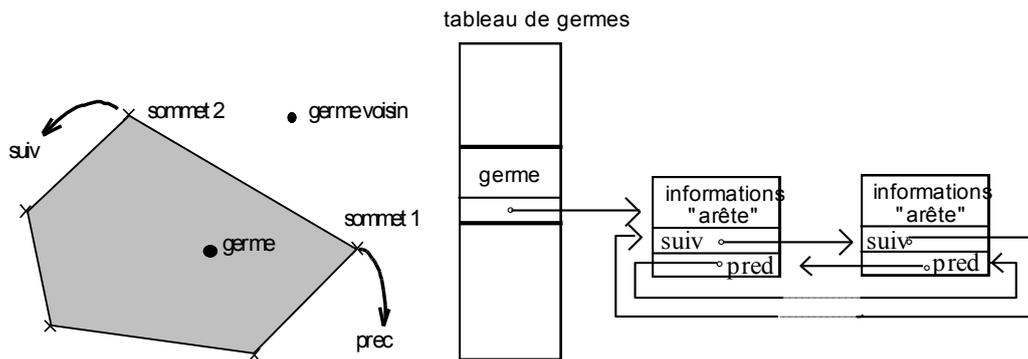


Figure 3.10 : Polygone de Voronoï, germes, arêtes, structure associée

Remarques :

- *A priori*, une seule extrémité par arête pourrait suffire, mais deux extrémités vont permettre de manipuler ces arêtes en tant que segments de façon simple.
- Une liste doublement chaînée permet de se déplacer dans un sens quelconque sur les arêtes du polygone.

Génération des germes

Le placement des germes dans l'image dépend du type de problème à résoudre. Leur nombre et leur position peuvent être fixe ou variable. A l'initialisation, ils sont placés aléatoirement, « manuellement », ou selon un marquage. Pour ce dernier cas, on peut utiliser par exemple un seuillage.

La morphologie mathématique fournit des outils pour extraire des germes ponctuels. Par exemple, la méthode du « chapeau haut-de-forme » exploite un élément structurant dont la forme et la taille s'adaptent aux formes que l'on

cherche à marquer. Pour obtenir des germes ponctuels à partir des pics ainsi extraits, on peut utiliser un amincissement séquentiel homotopique itéré jusqu'à stabilisation.

c) *Squelette par zones d'influence*

On peut créer une partition du plan en régions à partir d'un ensemble \mathbf{O} d'objets quelconques (segments, polygones, *etc.*) de telle sorte que chaque région englobe un objet o de \mathbf{O} . Une telle région est appelée la zone d'influence de l'objet. La *zone d'influence* de l'objet o est l'ensemble des points du plan plus proches de o que de tout autre objet de \mathbf{O} . Le graphe planaire formé par les frontières des régions est appelé le *diagramme de Voronoï généralisé* [MELK91].

L'ensemble des frontières des zones d'influence s'appelle en Morphologie Mathématique le *squelette par zone d'influence* (SKIZ). On peut le construire, pour des objets de forme quelconque, par amincissement séquentiel itéré jusqu'à stabilisation, ceci en employant un élément structurant homotopique, suivi d'un ébarbulage, du complémentaire de \mathbf{O} dans \mathcal{S} [SERR82].

Dans le cas où les objets sont des segments de droite ou des polygones, des algorithmes de construction plus rapides ont été proposés dans [ATTA94].

3 Structures hiérarchiques pour la décomposition d'une image

3.1 Arbre quaternaire ou « *quadtree* »

3.1.1 Principe de base

Le *quadtree* [SAME90] est une représentation de l'image exploitant la récursivité du pavage carré. Il ne s'applique que sur des images carrées dont le côté est une puissance de 2. Cette représentation correspond donc à la structure rigide du pavage carré.

Le *quadtree* est une arborescence dont la racine est l'image toute entière et dont chaque nœud (sauf les nœuds terminaux) possède exactement 4 fils. Le *quadtree* est défini de manière récursive : l'image (de taille $2^N \times 2^N$) est partagée en quatre quadrants ou blocs N-O, N-E, S-O, S-E (*cf.* figure 3.11). A chacun de ces quadrants est associé un nœud fils de la racine. Puis le processus de découpage en quatre quarts est itéré pour chacun des fils sans chevauchement des blocs. On peut découper ainsi l'image jusqu'aux pixels, l'arborescence possède alors $N+1$ niveaux.

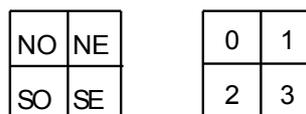


Figure 3.11 : Convention de désignation des blocs dans un *quadtree*

Un premier exemple d'application est le codage d'une image binaire (cf. figure 3.12). Chaque feuille du *quadtree* correspond à un bloc homogène de l'image (gris ou blanc). Les blocs hétérogènes (c'est-à-dire possédant à la fois des pixels gris et blancs) sont découpés en quatre : le nœud correspondant possède quatre fils. Une feuille mémorise de plus la couleur du bloc qu'elle représente (gris ou blanc), les nœuds non terminaux représentent les blocs hétérogènes.

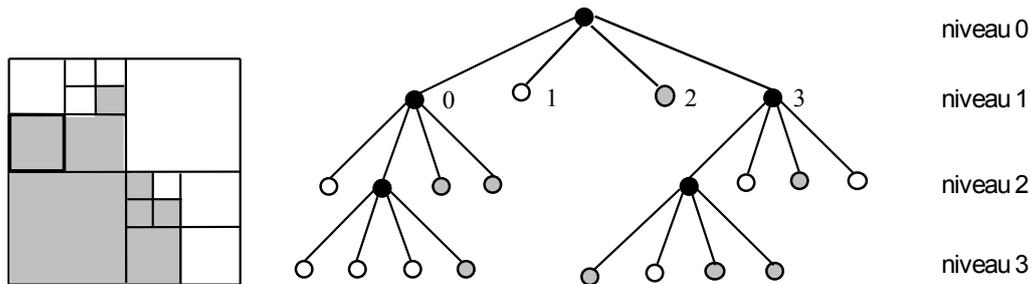


Figure 3.12 : Principe de codage d'une image binaire par un *quadtree*, permettant de représenter dans un ordre toujours identique les 4 fils de chaque sommet. On a représenté en noir les nœuds correspondant à des blocs hétérogènes

Cette structure de données peut s'adapter à toute segmentation en régions. Le principe de construction du *quadtree* est lié à la mesure d'un critère d'homogénéité sur une région. Ce critère peut être un niveau de gris constant, une variance minimale de niveau de gris ou tout autre attribut de région (cf. chapitre II).

La notion de voisinage nécessaire pour traiter une image, peut aussi être mise en œuvre dans un *quadtree*. Il existe, selon la méthode d'implantation utilisée pour coder un *quadtree*, divers algorithmes permettant d'accéder aux nœuds de l'arbre qui lui sont voisins dans l'image [SAME82]. Ce type d'opération élémentaire est très important dans le cadre de la segmentation région, car le *quadtree* est une méthode de décomposition de l'image permettant de proposer une partition initiale, avant une étape de fusion comme cela apparaît dans les méthodes de croissance de région (cf. chapitre XII).

3.1.2 Le *quadtree* linéaire

Dans la pratique, la représentation de l'arbre quaternaire sous forme arborescente pose plusieurs problèmes d'implantation. Premièrement, la mise en œuvre directe du découpage récursif serait très coûteuse en temps de calcul, puisque le test d'homogénéité d'un bloc nécessite la consultation de tous les pixels qui le constituent. Le *quadtree* est donc construit à partir de la racine, mais chaque point est consulté un nombre de fois égal à la profondeur du bloc pour lequel il apparaît dans l'arborescence. De plus, la représentation en mémoire implique l'utilisation de pointeurs ou d'indices, qui conduisent à une augmentation prohibitive du volume de codage. Enfin, la manipulation de la structure arborescente conduit aussi à utiliser des algorithmes de rafraîchissement d'arbres, pour les opérations les plus élémentaires, qui sont coûteux en temps de calcul.

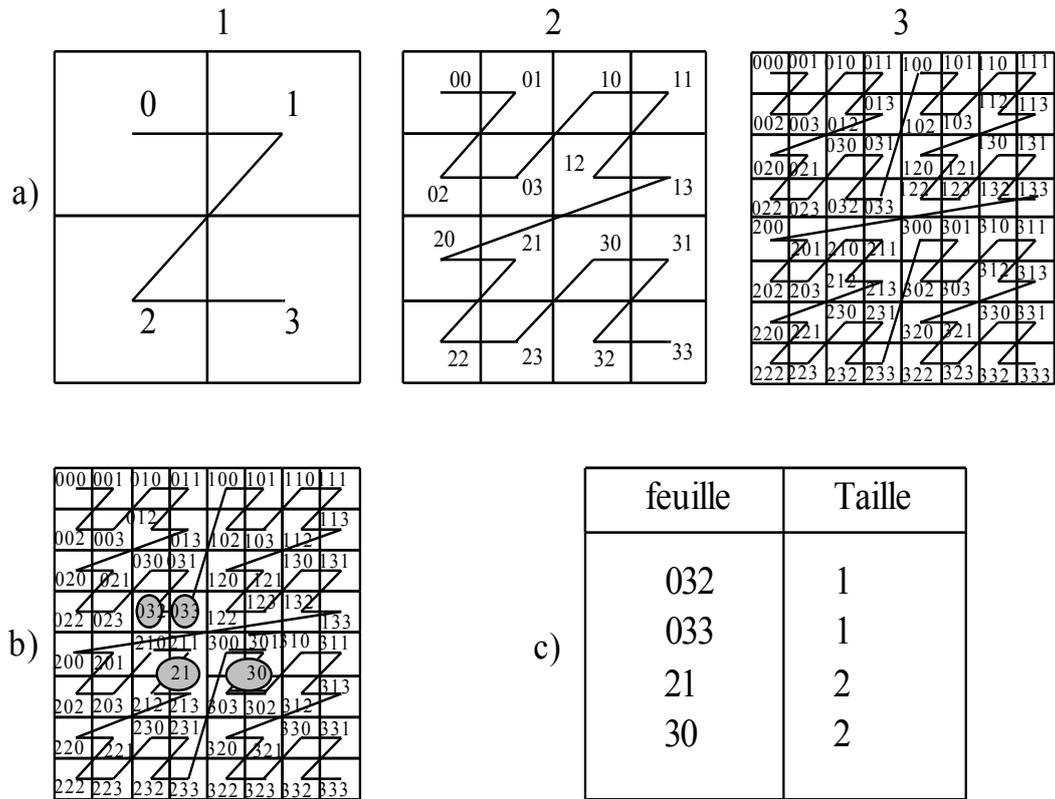


Figure 3.13 : Principe de la construction du quadtree par parcours de Peano :
 a- Courbe de Peano parcourant tout le plan aux première, deuxième et troisième itérations.
 b- Superposition d'un objet binaire à la grille obtenue après trois subdivisions récursives du plan
 c- Codage de l'objet représenté en b. Seuls les numéros identifiant les blocs constitutifs de l'objet sont mémorisés.

La mise en œuvre de fusion de blocs (cf. chapitre XII) suivant la structure de *quadtree* est en général moins coûteuse que la construction du *quadtree* par division. Ceci est vérifié quand les attributs exploités par les critères de fusion peuvent se calculer à partir des attributs des blocs « fils ». Il est alors inutile de parcourir tous les pixels de chaque bloc.

Un autre principe de construction par division utilisant le parcours de Peano est illustré par la figure 3.13. Au premier niveau de décomposition, les 4 quadrants de l'image sont analysés dans un sens particulier, qui est généralement celui représenté sur la figure 3.13.a.1. La figure 3.13.a montre le parcours pour les blocs de niveaux 1, 2 et 3. Le nombre de chiffres de chaque numéro correspond donc au niveau dans l'arborescence du bloc correspondant. Chaque élément de la courbe est numéroté dans l'ordre croissant, en base quatre. Cela conduit, si la décomposition de l'image est faite jusqu'au niveau du pixel, à une courbe de parcours de tous les points du plan, appelée courbe de Peano en Z. Il est donc possible de numérotter les points selon l'ordre défini par cette courbe.

On appelle *clef de Peano* la bijection de \mathbb{N}^2 dans \mathbb{N} qui associe à tout couple d'entiers $[i, j]$ l'entier obtenu en intercalant les bits de i et j , en représentation binaire (cf. figure 3.14).

La clef de Peano des coordonnées (exprimées en pixels) d'un point de l'image est le numéro de ce pixel sur la courbe de Peano.

La construction du *quadtree*, grâce à la clef de Peano, est simple : il suffit de parcourir les pixels de l'image suivant l'ordre de Peano pour détecter les quadrants homogènes. Ainsi, chaque pixel n'est visité qu'une seule fois et seuls les sommets présents dans le *quadtree* final sont créés.

Dans l'exemple de la figure 3.13.b, l'objet représenté est constitué par les 4 feuilles 032 et 033 (au niveau 3) et 21 et 30 (au niveau 2). Le parcours de Peano permet aussi de donner une représentation linéaire en ne codant que les feuilles grises. Il suffit, en effet, de ne retenir que les numéros des blocs homogènes (cf. figure 3.13.c). On peut noter que le numéro est ici donné en base 4, ce qui permet de faciliter son interprétation : en effet, chacun des chiffres désigne successivement, pour chaque niveau de décomposition, les quadrants dont est issue la feuille considérée. Il est inutile d'indiquer la taille du bloc qui est calculable à partir de la longueur du code de la feuille.

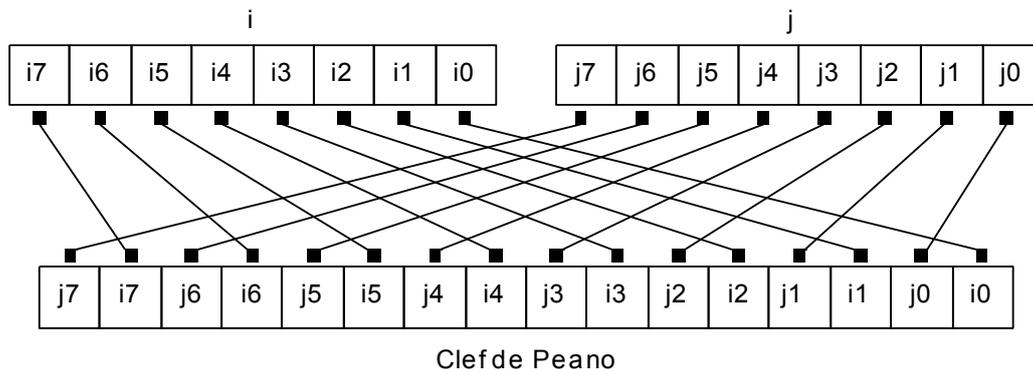


Figure 3.14 : Principe de la clef de Peano permettant de trouver la correspondance entre les coordonnées d'un pixel et son numéro sur la courbe de Peano.

3.1.3 Discussion

L'un des avantages des *quadtrees* est de proposer une méthode de division simple et facilement généralisable aux dimensions supérieures [MEAG82]. De plus, des opérations (addition, intersection, etc.) entre *quadtrees* peuvent être effectuées plus rapidement qu'entre des images, ce qui est intéressant pour des données très volumineuses.

Le problème majeur de cette structure provient de la rigidité des divisions réalisées sur l'image, empêchant une bonne adéquation aux données. En particulier, une simple translation de l'image d'un seul pixel perturbe complètement la structure du *quadtree*.

3.2 Les pyramides : modèles rigides et généralisations

3.2.1 La structure pyramidale

Le modèle pyramidal regroupe divers aspects déjà énoncés. L'idée de base est de procéder à une analyse des données à plusieurs échelles : une détection d'objet doit être possible quelle que soit la taille de celui-ci dans l'image. Pour cela, on procède à une analyse multi-résolution de l'image en utilisant un modèle pyramidal.

Une *pyramide* est une suite finie de graphes, permettant de prendre en compte toutes les résolutions de l'image initiale. Chacun de ces graphes constitue un *niveau* de la pyramide (cf. figure 3.15).

Lorsque les nœuds ne peuvent avoir qu'un seul nœud père, la pyramide est dite sans *recouvrement*, c'est le cas du *quadtree*, sinon, c'est une pyramide avec *recouvrement*.

La construction de la pyramide est essentiellement un processus ascendant : le niveau le plus bas de la pyramide représente l'image à pleine résolution. Les niveaux supérieurs sont des représentations à une résolution de plus en plus réduite de l'image de base.

3.2.2 Pyramides rigides et pyramides stochastiques adaptatives

Il existe deux familles distinctes de pyramides. Les pyramides rigides et les pyramides de graphes ou pyramides stochastiques adaptatives.

Dans le cas des pyramides rigides, chaque niveau de la pyramide peut généralement se représenter par une image. La dimension de ces images diminue au fur et à mesure que l'on s'éloigne de la base. Le rapport entre le nombre de nœuds (pixels) de deux graphes consécutifs est généralement constant, et le nombre de voisins par nœud est toujours le même. Par exemple, une suite de *quadtrees* est une pyramide rigide où le nombre de voisins à chaque niveau est de 3 et où le nombre de nœuds est dans un rapport 4 entre deux niveaux consécutifs.

Dans le cas des pyramides stochastiques adaptatives [MEER89] [MATH93] [MATH95], chaque nœud d'un niveau correspond à une région de l'image de base. La dimension de l'image représentée par un graphe est constante quel que soit le niveau. Le nombre de nœuds du graphe diminue lorsque l'on passe d'un niveau au niveau immédiatement supérieur. Par contre, le rapport entre le nombre de nœuds (régions) de deux graphes de niveaux consécutifs n'est plus constant et le nombre de voisins par nœud varie également. La structure de la pyramide dépend de l'image traitée.

3.2.3 Discussion

La structure pyramidale présente plusieurs attraits. Premièrement, elle permet de faciliter l'extraction d'informations et constitue une étape vers une structuration de plus haut niveau. De plus, elle se prête bien à la parallélisation des algorithmes, sur des machines appropriées, du fait de l'aspect local des traitements appliqués sur les données. En effet, le nombre de voisins par nœud, à un niveau donné, est

toujours assez faible, même pour les pyramides de graphes. Le nombre d'informations à traiter pour chaque nœud reste donc toujours assez limité.

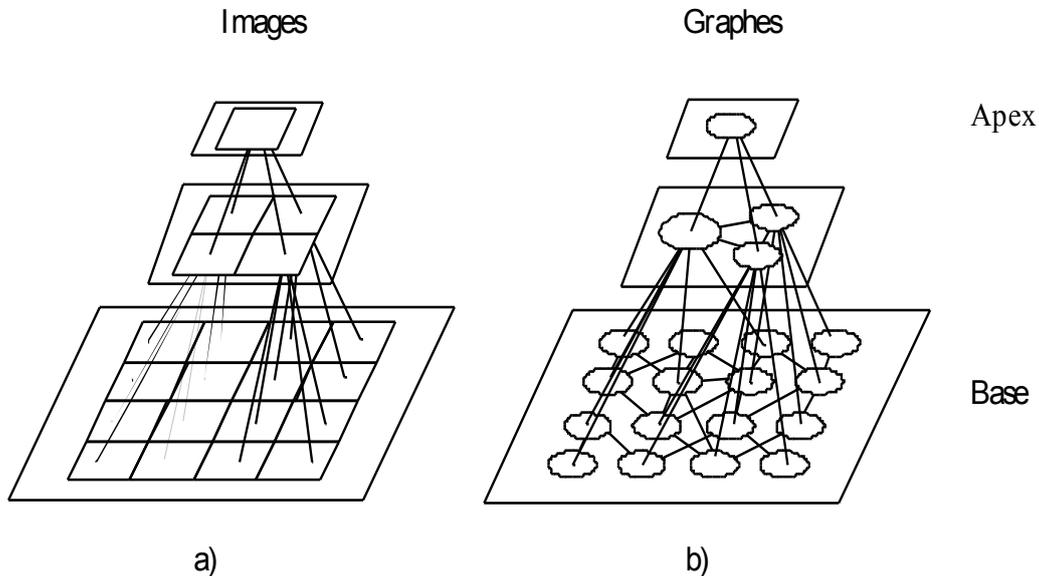


Figure 3.15 : Exemples de pyramides :
 a) pyramide rigide basée sur une structure de type quadtree b) pyramide stochastique adaptative

4 Codage de contours

Au paragraphe 2.1.2 la notion de voisinage, associé à un maillage et à une métrique, a été présentée. Le voisinage est une notion locale qui permet cependant de définir une frontière ou un contour de manière globale.

Le codage local le plus simple pour stocker une ligne fermée ou non est le code de Freeman : il caractérise le passage d'un pixel à son successeur ; la suite des codes locaux donne le codage d'un contour (ou d'une ligne) dans son intégralité (*cf.* § 4.1).

Par ailleurs, on peut simplifier le codage d'un contour ou d'une ligne, en l'approximant par une chaîne polygonale. Ceci a le double avantage de lisser le contour et de réduire la quantité d'information à stocker (une ligne polygonale est définie par ses sommets successifs). L'approximation polygonale est souvent employée avant une phase de reconnaissance des formes ou pour comparer des objets dans une approche multi-résolution. De nombreuses méthodes d'approximations polygonales existent dans la littérature, quelques-unes sont présentées au paragraphe 4.2.

4.1 Code de Freeman

Le code de Freeman associe à chaque déplacement élémentaire (vers un des voisins définis par la 8-connextité) un code entier α dans l'intervalle $[0, 7]$ suivant la convention donnée en figure 3.16. On parle de direction de Freeman [FREE61].

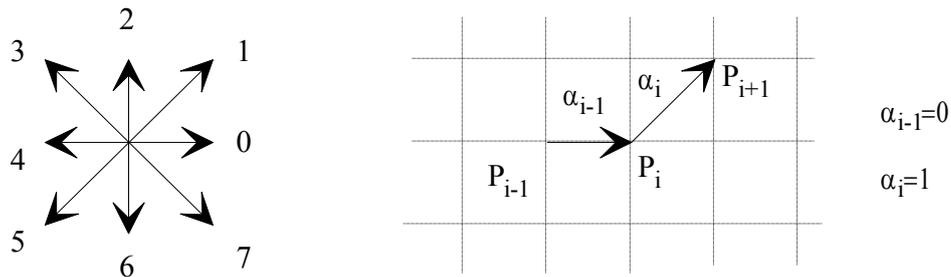
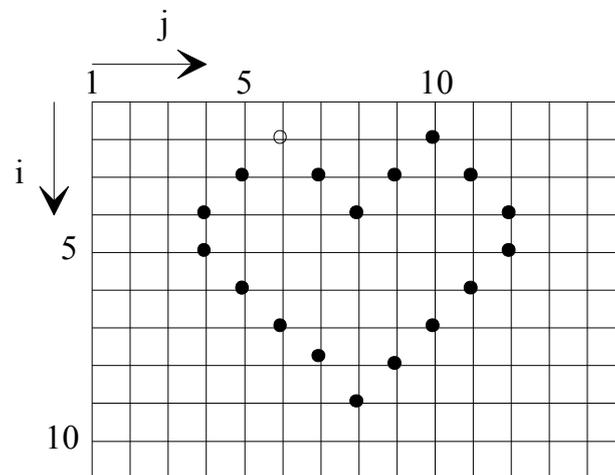


Figure 3.16 : Directions de Freeman et déplacement sur un contour.

L'utilisation des directions de Freeman est alors immédiate : toute information contenue dans une image et qui peut être parcourue de manière linéaire se code par la succession des directions de Freeman le long de ce parcours.



$P_0 = [1,6]$ et suite des directions : $\{7,7,1,1,7,7,6,5,5,5,5,3,3,3,3,2,1,1\}$

Figure 3.17 : Exemple de code de Freeman

Soit la suite de pixels $\{P_i\}_{i=0,..,n}$ où P_i est adjacent à P_{i-1} pour $i = 1, \dots, n$; on stocke pour le point P_i la direction α_i permettant d'obtenir le point P_{i+1} (cf. figure 3.16). Le *code de Freeman* est alors formé du point initial P_0 et de la suite $\{\alpha_i\}_{i=0,..,n}$ des directions de Freeman (cf. figure 3.17).

Souvent, la suite $\{P_i\}_{i=0,..,n}$ est le résultat d'un suivi de contour dans une image binaire, ou d'une fermeture de contour après une extraction de points de frontière dans une image en niveaux de gris.

Couramment, le code de Freeman est employé pour coder un arc, une courbe, un contour ou une frontière, puisque ce sont des entités par nature linéaire et que ce code permet d'en garder l'expression. Mais le code de Freeman permet bien sûr d'exprimer tout chemin dans une image.

Le code de Freeman est un codage exact : les coordonnées d'un point P_k se calculent aisément à partir de la donnée de P_0 et des $\{\alpha_i\}_{i=0,..,n}$. Il assure de plus une compression des données.

Des opérations géométriques telles que translation, homothétie de rapport entier, rotation d'un angle multiple de $\pi/2$ se réalisent simplement sur ce code.

Des opérations de réduction de code peuvent être définies : par exemple, la direction 0 et la direction 4 s'annulent. Ceci permet de comparer des pixels définis par leur code de Freeman, de rechercher des intersections, *etc.* [CHAS91]

Lorsque l'on code une courbe, un contour ou une frontière, il faut choisir le point de départ P_0 ; c'est souvent le premier point rencontré lors d'un balayage séquentiel de l'image.

Une manière de rendre le code de Freeman invariant en rotation pour une forme donnée, consiste à associer à la suite des directions de Freeman, le nombre défini par la suite de chiffres, puis à effectuer une permutation circulaire sur ces chiffres jusqu'à obtenir le plus petit nombre. Le résultat est unique et il caractérise une forme indépendamment de sa position dans l'image.

Par exemple, pour la figure 3.17, le résultat est le code :

$$\{1,1,7,7,1,1,7,7,6,5,5,5,5,3,3,3,3,2\}$$

qui caractérise la forme d'un cœur, quelles que soit sa position et son orientation dans l'image.

Le code de Freeman peut être transformé de manière à coder le nombre d'itérations successives dans une même direction ; suivant le type de forme, il pourra en résulter un taux de compression plus important.

Le principe du codage de Freeman s'étend sans difficulté aux autres maillages réguliers.

4.2 Codage de la frontière d'un objet en 8-connexité

Le suivi de contour permet de générer le code de Freeman de la frontière d'un objet. L'objet est une composante 8-connexe. On appelle points frontière, les points de l'objet 4-connectés au fond. Lorsque l'on est sur un point de contour P_i , on recherche le point de contour suivant en parcourant dans un sens donné (trigonométrique) les 8-voisins de P_i à partir d'un point dont on est certain qu'il appartient au fond. Il s'agit du point de fond qui précédait P_i dans l'exploration du voisinage de P_{i-1} . On désigne par dir_{init} la direction qui permet de passer du point P_i à ce point de fond pour commencer la localisation de P_{i+1} (*cf.* figure 3.18). Au départ, on part du point de fond situé à gauche de P_0 pour trouver P_1 .

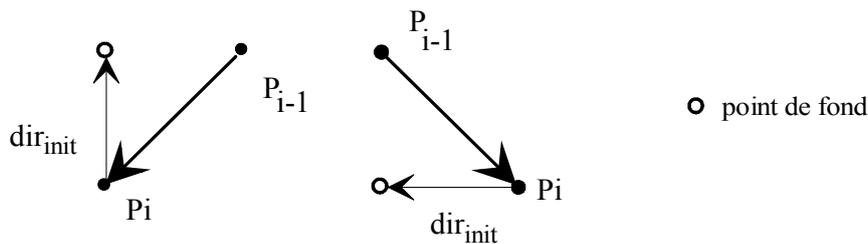


Figure 3.18 : Détermination de la direction initiale de recherche dans le suivi de contours

L'algorithme 3.1 présente le principe d'extraction de la chaîne de Freeman correspondant à la frontière d'un objet.

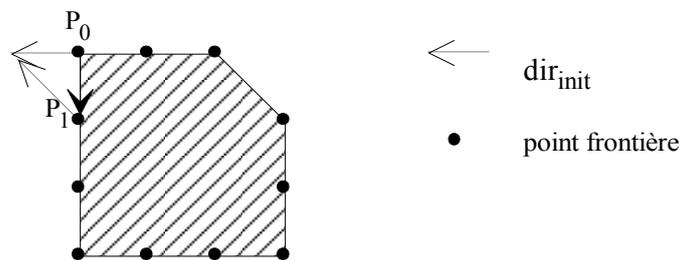


Figure 3.19 : Suivi de la frontière d'un objet

4.3 Approximations polygonales

Il s'agit de construire une représentation polygonale d'un contour ou d'une frontière de région : des points significatifs vont être les sommets de la ligne polygonale.

Plusieurs approches sont possibles :

- reconnaître les segments de droite discrets [ROSE74],
- approximer au mieux le contour pour un nombre de segments fixé, ou pour une erreur minimale fixée [PHIL68],
- chercher des points caractéristiques pour décrire un contour [DAVI77] :
 - points à forte courbure [ROSE73],
 - points détectés par un découpage récursif,
 - points détectés par un suivi des points de contours (méthodes incrémentales).

Les principes des deux dernières méthodes sont exposés § 4.3.1 et § 4.3.2.

```

Parcours de l'image du haut vers le bas et de la gauche vers la droite pour trouver le premier
point de contour du parcours;
/* Soit P0 le premier point, dans la direction dirinit=4, on trouve un point de fond
la chaîne des directions est stockée dans CH */
SI tous les voisins de P0 sont dans le fond ALORS P0 est un point isolé;
SINON
    /* soient :
        s le site courant et t le site suivant
        dir la direction désignant le point de frontière suivant à partir de s
        dirinit la direction à partir de laquelle on est sûr de trouver un point de fond
        à partir de s*/
    DEBUT
        fini := FAUX; t := [0, 0]; dir := 4; k := 1;
        TANT QUE NON fini FAIRE
            DEBUT
                TANT QUE t ∉ objet FAIRE
                    DEBUT
                        dir := (dir+1) mod 8;
                        calculer le nouveau site t à partir de s et dir;
                        dirinit := (dir+5) mod 8;
                    FIN
                    SI t est le deuxième point
                    ALORS mémoriser t dans P1;
                    SINON SI (s = P0) ET (t = P1) ALORS fini := VRAI
                    s := t;
                    t := point déduit de t suivant dirinit;
                    CH[k] := dir; k := k+1;
                    dir := dirinit;
                FIN
            FIN
        FIN
    FIN
FIN

```

Algorithme 3.1 : Elaboration de la chaîne de Freeman correspondant au contour d'un objet

4.3.1 Approximation par découpage récursif

Etant donné un arc (ensemble consécutif de points) $\{P_0, \dots, P_n\}$, on détecte les points caractéristiques de manière récursive. Le processus complet comporte une phase de division suivie d'une phase de regroupement des segments.

Le processus de découpage est récursif :

- on recherche entre P_0 et P_n un point caractéristique ;
- si un tel point, noté P_k , est détecté, alors on applique le même procédé entre P_0 et P_k , et entre P_k et P_n [PAVL74] (cf. figure 3.20).

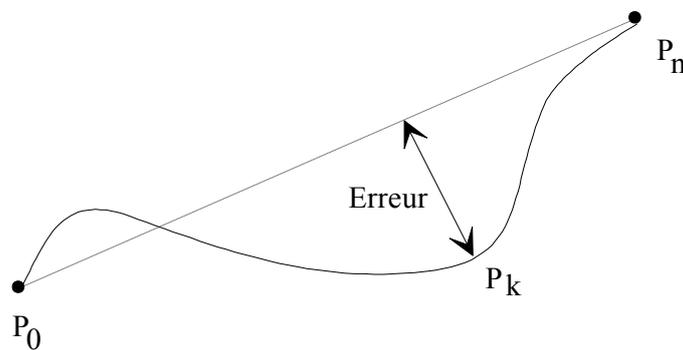


Figure 3.20 : Principe du découpage récursif
(en trait plein le contour, en pointillés le segment $[P_0 P_n]$).

Généralement, P_i est le point de l'arc $\{P_0, \dots, P_n\}$ pour lequel la distance au segment de droite $[P_0 P_n]$ est maximale. La finesse de l'approximation est associée à un seuil de tolérance latérale, noté E_m , au dessous duquel le découpage récursif n'est pas rappelé. Le nombre minimum de points que doit posséder un segment peut également être contrôlé par un seuil, noté L_m . D'où la procédure décrite par l'algorithme 3.2.

Le premier appel de la procédure est de la forme : *découpage* (P_0, P_n).

La phase de division est sensible au choix des points extrémités (cf. figure 3.21).

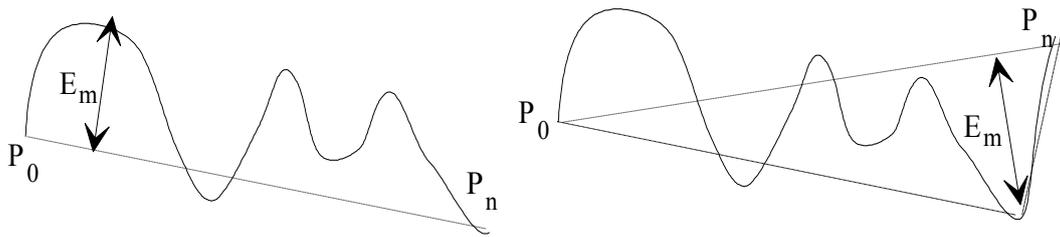


Figure 3.21 : Suivant les points extrémités, $\{P_0, \dots, P_n\}$ est approximé par un segment dans le premier cas et par deux segments dans le second, pour une même valeur de E_m (en trait plein le contour; en pointillés le segment $[P_0 P_n]$).

```

procédure découpage ( $P_i, P_j$ ) ;
DEBUT
SI  $j-i > L_m$ 
ALORS parcourir les points de  $P_{i+1}$  à  $P_{j-1}$  pour calculer Erreur : l'écart maximal à la
droite ( $P_i P_j$ );
    SI Erreur  $\geq E_m$  ALORS
        DEBUT
            soit le point  $P_k$  tel que  $i < k < j$  et (Erreur = écart en  $P_k$ ) ;
            découpage ( $P_i, P_k$ );
            inscrire  $P_k$  comme nouveau sommet de l'approximation ;
            découpage ( $P_k, P_j$ ) ;
        FIN
    FIN
FIN
  
```

Algorithme 3.2 : Procédure récursive de polygonalisation d'un contour, basée sur la distance maximale

La phase de regroupement assure la fusion de segments adjacents $[P_i P_j]$ et $[P_j P_k]$ pour former le segment $[P_i P_k]$ suivant le même principe de contrôle d'erreur entre P_i et P_k que lors de la phase de découpage. Des segments qui étaient traités séparément, bien qu'adjacents dans la récursivité, sont ainsi remis en relation (cf. figure 3.22).

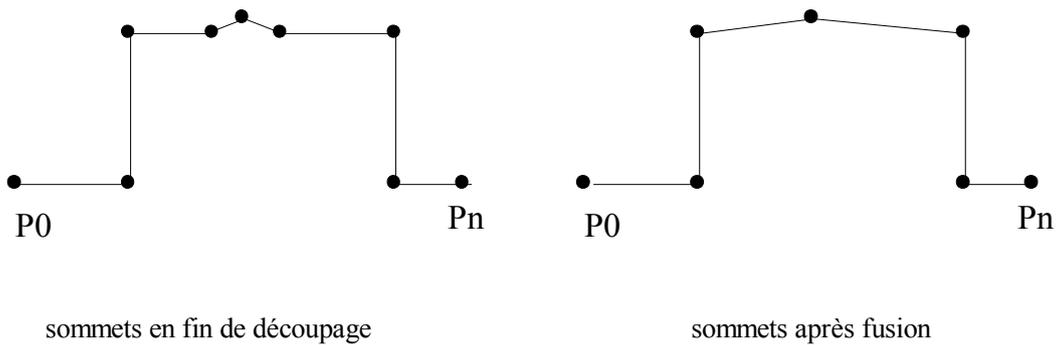


Figure 3.22 : Point de cassure généré par un bruit et supprimé par la phase de fusion.

4.3.2 Approximation itérative

Les procédés itératifs sont caractérisés par le fait que les points sont traités en « flot de données » dans l'ordre établi par le suivi du contour. Les points sont successivement regroupés jusqu'à ce qu'un certain critère, mis à jour à chaque ajout, ne soit plus vérifié.

```

procédure approximation ;
DEBUT
   $i_{\text{déb}} := 0$  ; Erreur := 0 ;
  POUR  $j := 1$  à  $n$  FAIRE
    DEBUT
      calculer les paramètres de la droite  $D$  passant par  $P_{i_{\text{déb}}}$  et  $P_j$  ;
      mettre à jour Erreur au moyen de  $D$  et  $\{P_k\}_{k=i_{\text{déb}}..j}$  ;
      SI Erreur  $> E_m$  ALORS
        DEBUT
          /* le seuil d'erreur est dépassé par  $P_j$  ,
          on termine le segment en cours en  $P_{j-1}$  */
           $P_{j-1}$  est stocké comme sommet de l'approximation ;
           $i_{\text{déb}} := j-1$  ; Erreur := 0 ;
        FIN
      FIN
    FIN
  SI  $P_n \notin \{\text{liste des sommets}\}$  ALORS ajouter  $P_n$  à cette liste ;
FIN
  
```

Algorithme 3.3 : Procédure itérative de polygonalisation d'un contour

Les extrémités du segment $[P_i P_j]$ correspondent au premier et au dernier point d'une séquence incluse dans l'arc $\{P_0, \dots, P_n\}$ à avoir satisfait le critère, et le point P_j devient le point initial pour continuer le procédé. L'intérêt des méthodes itératives est d'éviter le stockage de tout le contour (comme cela est nécessaire pour les méthodes récursives), et de pouvoir être mis en « pipeline » avec le processus de suivi de contour.

L'algorithme 3.3, présente la procédure itérative d'élaboration d'un contour qui a pour paramètres la liste des pixels $\{P_0, \dots, P_n\}$ et un seuil d'erreur E_m .

La droite D est soit la droite $(P_{idéb}, P_j)$, soit la droite de meilleure approximation au sens des moindres carrés du nuage de points $\{P_{idéb}, \dots, P_j\}$.

Erreur est soit le maximum de la distance entre chaque point de l'arc $\{P_{idéb}, \dots, P_j\}$ et D , soit la distance cumulée pour tous les points de cet arc.

Une variante permet de rester dans l'espace discret [WALL84], en utilisant simplement un calcul d'erreur qui est le cumul des aires signées des triangles $(P_{idéb}, P_{j-1}, P_j)$ (cf. Algorithme 3.4).

```

procédure approximation-bis ;
DEBUT
iidéb := 0 ; Erreur := 0 ;
POUR j := 1 à n FAIRE
    DEBUT
        /* Pidéb a pour coordonnées [xidéb , yidéb],
        Pj a pour coordonnées [xj , yj] */
        Erreur := Erreur + (xj - xidéb).(yj-1 - (yj - yidéb).(xj - xj-1);
        SI |Erreur| ≥ Em . ((xj - xidéb)2 + (yj - yidéb)2)1/2 ALORS
            DEBUT
                /*le seuil d'erreur est dépassé par Pj,
                on termine le segment en cours en Pj-1*/
                Pj-1 est stocké comme sommet de l'approximation ;
            FIN
            iidéb := j-1 ; Erreur := 0 ;
        FIN
    FIN
FIN

```

Algorithme 3.4 : Variante de la procédure itérative de polygonalisation d'un contour

Dans cet algorithme, E_m représente le maximum d'erreur autorisé par unité de longueur du segment approximant $[P_{idéb} P_j]$.

5 Etiquetage en composantes connexes

La notion de voisinage (*cf.* 2.1.2) permet de définir la connexité (4- ou 8-connexité). Une composante 4-connexe (resp. 8-connexe) est telle qu'entre tout couple de pixels de la composante, il existe un chemin 4-connexe (resp. 8-connexe).

L'étiquetage en composantes connexes d'une image attribue à tous les pixels d'une composante connexe une même étiquette (valeur entière). Chaque composante connexe est alors identifiée par son étiquette.

Les algorithmes d'étiquetage en composantes connexes détectent les adjacences entre pixels et définissent l'étiquette du point courant en fonction de celles des points voisins.

La méthode la plus classique repose sur un balayage séquentiel de l'image. On considère pour chaque point P ses voisins déjà traités : deux points pour la 4-connexité et quatre points pour la 8-connexité. Ces points sont les « prédécesseurs » de P lors d'un balayage ligne par ligne de l'image.

Il existe une méthode, non détaillée ici, où l'image est parcourue de manière itérative suivant une alternance de balayages « avant » (indices de lignes et de colonnes croissants) et de balayages « arrière » (indices de lignes et de colonnes décroissants) jusqu'à stabilisation de l'étiquetage.

Pour réduire à deux le nombre de balayages de l'image, il est nécessaire de construire une *table d'équivalence* \mathbf{T} [ROSE66] qui permet de gérer les équivalences d'étiquettes qui apparaissent lors du parcours séquentiel (*cf.* figure 3.23).

L'algorithme 3.5 exploite cette table de correspondance \mathbf{T} qui est initialisée par $\mathbf{T}[k] := k$. La taille de \mathbf{T} dépend de la nature de l'image. Dans cet algorithme, on suppose que tous les pixels de l'image de départ \mathbf{A} sont à zéro sauf des ensembles de pixels à 1 dont il faut extraire les composantes connexes. L'étiquetage est effectué sur chaque pixel de l'*image des étiquettes* \mathbf{E} de même taille que \mathbf{A} .

```

DEBUT
k := 1;
/* dans l'algorithme s représente un site de coordonnées i,j */
POUR i := 1 A M FAIRE
    POUR j := 1 A N FAIRE
        DEBUT
            SI (tous les prédécesseurs de A[s] sont à zéro) ALORS
                E[s] := k; k := k+1;
            SINON
                SI tous les prédécesseurs de A[s] ont la même étiquette  $\varepsilon$  ALORS
                    E[s] :=  $\varepsilon$ ;
                SINON
                    DEBUT
                        soit  $\varepsilon = \min\{T[E[s_p]] \mid (s_p \in \{\text{ensemble des prédécesseurs de s}\})$ 
ET (E[sp] ≠ 0)}
                    E[s] :=  $\varepsilon$ ;
                    POUR chaque prédécesseur d'étiquette  $\beta$  telle que T[ $\beta$ ] ≠  $\varepsilon$  FAIRE
                        DEBUT
                            TANT QUE T[ $\beta$ ] ≠  $\varepsilon$  FAIRE
                                DEBUT
                                    m := T[ $\beta$ ]; T[ $\beta$ ] :=  $\varepsilon$ ;  $\beta$  := m;
                                FIN
                            FIN
                    FIN
                FIN
        FIN
    FIN
FIN

```

Algorithme 3.5 : Etiquetage de composantes connexes

A la fin de ce balayage de l'image, on actualise la table **T** de manière à ce qu'à tout indice corresponde l'étiquette définitive de l'objet (les étiquettes attribuées aux objets dans l'image ne seront pas forcément consécutives). On applique donc l'algorithme 3.6.

```

POUR k := 1 JUSQU'A nombre maximum d'étiquettes utilisées FAIRE
    DEBUT
        m := k;
        TANT QUE T[m] ≠ k FAIRE m := T[m] ;
        T[k] := m;
    FIN;
    
```

Algorithme 3.6 : Actualisation de la table d'équivalence

Le second balayage de l'image exploite la table d'équivalence pour affecter à chaque pixel la plus grande étiquette d'une classe d'étiquette équivalente.

1	1	1	0	1	1	0	1
1	1	0	1	1	0	1	1
1	0	1	1	0	1	1	1
1	1	1	0	1	1	1	1
0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

image de départ

1	1	1	0	2	2	0	3
1	1	0	4	2	0	5	3
1	0	6	2	0	7	3	3
1	1	1	0	8	3	3	3
0	0	0	9	3	3	3	3
10	10	10	3	3	3	3	3
10	10	10	3	3	3	3	3
10	10	3	3	3	3	3	3

image des étiquettes après le premier passage

1	1	1	0	1	1	0	3
1	1	0	1	1	0	3	3
1	0	1	1	0	3	3	3
1	1	1	0	3	3	3	3
0	0	0	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3

image des étiquettes après mise à jour

T	1	2	3	4	5	6	7	8	9	10
	1	2	3	4	5	6	7	8	9	10
				2	3	2	3	3	3	3
		1		1		1				

table d'équivalence T : les colonnes

correspondent aux mises à jour successives,

la première colonne correspond à l'état initial

Figure 3.23 : Etiquetage en 4-connexité

CHAPITRE IV

Prétraitements

Comme on l'a vu précédemment, la segmentation consiste à effectuer une partition de l'image en régions homogènes par rapport à un ou plusieurs critères. Les régions obtenues se distinguent les unes des autres par des différences significatives selon ces mêmes critères.

L'étape de prétraitement a pour but de faciliter la segmentation en renforçant la ressemblance entre pixels appartenant à une même région, ou en accentuant la dissemblance entre pixels appartenant à des régions différentes.

Les images comportent généralement une grande quantité de données. Pour éviter des temps de traitement trop élevés, on cherche donc à leur appliquer des opérateurs locaux, qui ne font appel, pour le calcul, qu'à un nombre limité de pixels, situés dans un proche voisinage du pixel courant.

Quelques méthodes de prétraitement sont maintenant présentées. Elles concernent :

- la modification d'histogramme,
- la réduction de bruit,
- le rehaussement de contraste.

Elles sont exposées en considérant que la segmentation est basée sur l'intensité des pixels. Naturellement, ces méthodes peuvent être utilisées avec tout attribut numérique calculable en chaque pixel (*cf.* chapitre II).

1 Modification d'histogramme

On cherche à améliorer l'image en lui appliquant une transformation ponctuelle d'intensité. A tout pixel d'intensité a_s on associe une intensité $a'_s = T(a_s)$. La transformation T est choisie croissante, de façon à conserver les contrastes relatifs entre régions (une région claire sur un fond sombre apparaîtra plus claire que le fond dans l'image transformée). L'image est considérée ici, soit comme un signal déterministe, soit comme un ensemble de réalisations d'une variable aléatoire.

Du fait de leur caractère ponctuel, les méthodes de transformation d'histogramme n'affectent pas la forme des régions. Elles en modifient uniquement l'apparence visuelle.

1.1 Expansion de dynamique

Cette méthode élémentaire consiste à utiliser au mieux l'échelle de niveaux de gris disponible sur le système d'acquisition d'image.

Soient $A[i, j]$ l'image de départ et $A'[i, j]$ l'image après transformation. Soient $[a_0, a_1]$ l'intervalle des intensités présentes dans l'image et $[a_{\min}, a_{\max}]$ l'intervalle disponible. L'expansion de dynamique correspond à la transformation linéaire T suivante :

$$a'_s = \alpha + \beta a_s$$

telle que :

$$\forall a \in [a_0, a_1] \quad a \xrightarrow{T} a' \in [a_{\min}, a_{\max}]$$

avec :

$$\alpha = \frac{a_{\min} \cdot a_1 - a_{\max} \cdot a_0}{a_1 - a_0} ; \beta = \frac{a_{\max} - a_{\min}}{a_1 - a_0} \quad (4-1)$$

Cette transformation ne fait pas qu'améliorer l'aspect visuel de l'image (cf. figure 4.1). En effet, les données étant codées sur un nombre limité de bits, généralement 8, on atténue ainsi les erreurs d'arrondi pouvant se propager dans les traitements ultérieurs. Il est alors possible d'éviter de recourir à la représentation des nombres en virgule flottante, coûteuse en temps de calcul et en place mémoire.

1.2 Egalisation d'histogramme

Cette transformation consiste à rendre le plus plat possible, l'histogramme des niveaux de gris de l'image. L'image est associée à une modélisation probabiliste dont l'histogramme représente la densité de probabilité de la variable niveau de gris. Cette méthode permet souvent de renforcer le contraste sur des détails de l'image qui sont masqués par des variations d'intensité de plus grande amplitude et à plus grande échelle.

L'image $A[i, j]$ est considérée comme un ensemble de réalisations d'une variable aléatoire A admettant une densité de probabilité $f(a)$ non nulle sur $]a_{\min}, a_{\max}[$ et une fonction de répartition $F(a)$. f représente donc l'histogramme normalisé de $A[i, j]$ et F son histogramme normalisé cumulé.

On cherche alors une transformation T continue, dérivable au moins par morceaux et strictement croissante telle que la variable aléatoire $B = T(A)$ soit uniformément répartie dans l'intervalle $[b_{\min}, b_{\max}]$. Soit $g(b)$ la densité de probabilité de B et T' la fonction dérivée de T . On obtient :

$$g(b) = \begin{cases} f(a) \frac{1}{T'(a)} = \frac{1}{b_{\max} - b_{\min}} & \text{pour } b_{\min} < b < b_{\max} \\ 0 & \text{ailleurs} \end{cases} \quad (4-2)$$

avec :

$$b = T(a) \quad \text{et } a \in [a_{\min}, a_{\max}] \quad 3$$

ceci équivaut à :

$$T'(a) = (b_{\max} - b_{\min}) f(a) \quad (4-3)$$

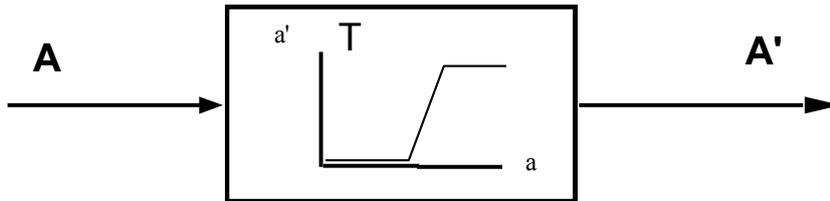
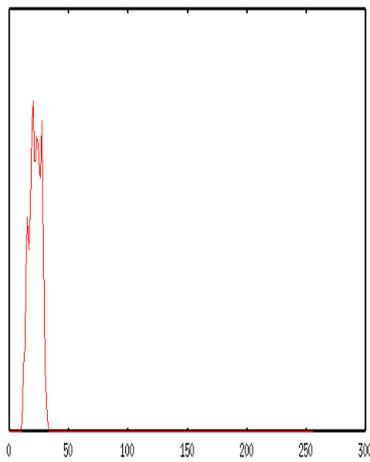


image initiale

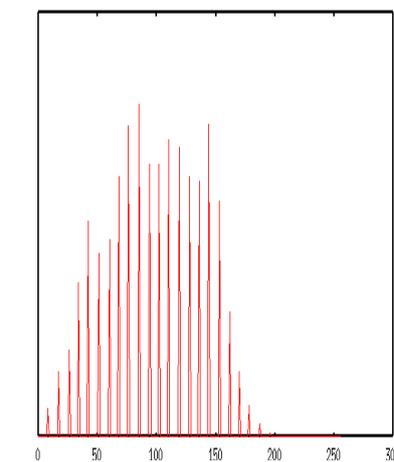


image transformée

effectif



histogramme de départ



histogramme après expansion

Figure 4.1 : Expansion de dynamique des intensités

La transformation T est alors définie par :

$$T(a) = (b_{\max} - b_{\min}) F(a) + b_{\min} \text{ pour } a \in [a_{\min}, a_{\max}] \quad (4-4)$$

En considérant des petites fluctuations autour du niveau a , on peut linéariser la transformation T . L'amplitude des fluctuations est multipliée par le facteur $T'(a) = (b_{\max} - b_{\min}) \cdot f(a)$. L'égalisation a donc pour effet d'amplifier les fluctuations dans les zones où celles-ci sont faibles, ces zones ayant des intensités de densité de probabilité élevée. La transformation étant continue et monotone, le sens et la position des transitions sont conservés.

En pratique, la fonction de répartition $F(a)$ est estimée par l'histogramme cumulé normalisé.

La figure 4.2 permet d'évaluer l'effet de l'égalisation d'histogramme. Dans l'image égalisée, les détails visibles dans les régions stationnaires sont amplifiés. En général, il n'est pas possible de faire une égalisation parfaite de l'histogramme de l'image transformée. En effet, les intensités de l'image de départ occupent des valeurs entières $a \in [a_{\min}, a_{\max}]$ et les intensités de l'image d'arrivée sont obtenues par arrondi des valeurs $T(a)$, ce qui peut aboutir à privilégier certains niveaux.

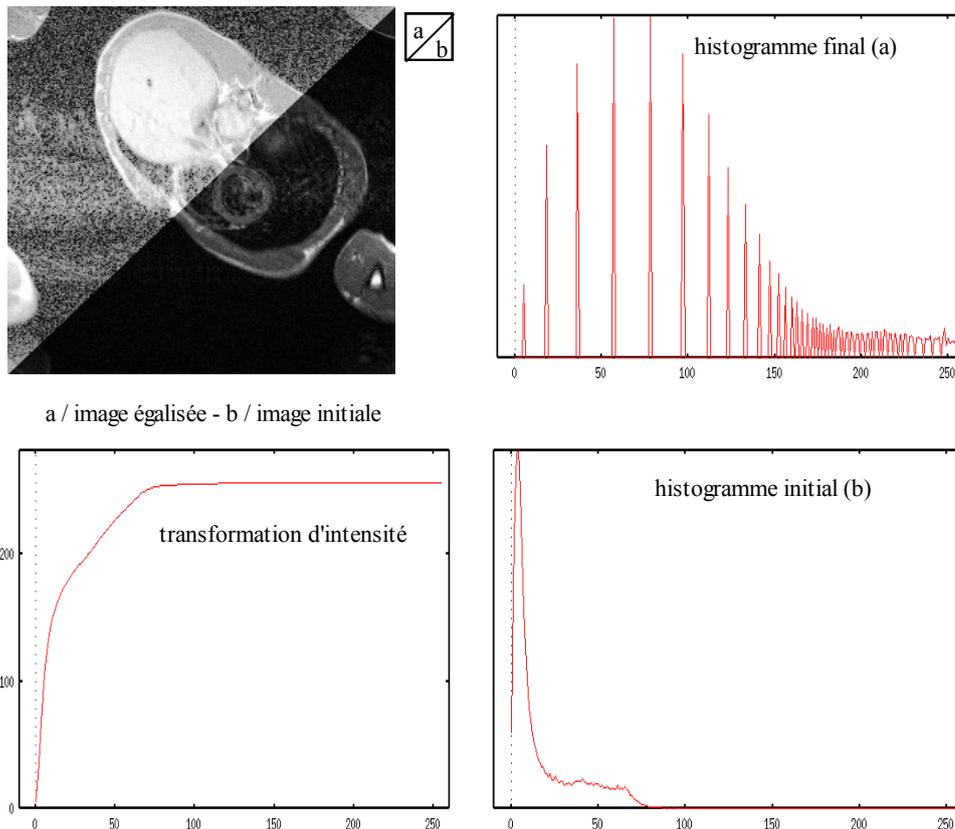


Figure 4.2 : Egalisation d'histogramme

Par ailleurs, quand le nombre de niveaux de gris rencontrés dans l'image initiale est faible, la transformation ne pourra que répartir différemment ces niveaux, mais sans changer leur effectif. Une solution possible à ce problème [HUMM77] [ROSE82] consiste à effectuer un tirage au sort pour l'affectation du niveau d'arrivée. Mais ceci a tendance à bruyter l'image d'arrivée, tout en complexifiant la transformation.

1.3 Spécification d'histogramme

Cette transformation consiste à rendre la distribution d'intensité de l'image voisine d'une distribution spécifiée à l'avance [GONZ87]. Afin de conserver la position et le sens des transitions présentes dans l'image initiale, on choisit une transformation croissante. La distribution de référence peut être obtenue soit à partir d'une image, soit à partir d'une région d'une image.

Soit **A** l'image initiale, **R** l'image de référence et **B** l'image transformée. La transformation d'intensité est notée **T**. Comme précédemment, les images sont considérées ici comme des ensembles de réalisations de variables aléatoires absolument continues notées respectivement A , R et B . Ces variables aléatoires sont caractérisées par leurs densités de probabilité f_A , f_R , f_B et leurs fonctions de répartition F_A , F_R , F_B . On suppose que f_R est non nulle sur l'intervalle $]r_{\min}, r_{\max}[$.

Soit $a \in]a_{\min}, a_{\max}[$, l'intensité d'un pixel dans l'image initiale **A** et $b = T(a)$ l'intensité correspondante dans l'image transformée. On détermine **T** à l'aide des relations suivantes :

$$P(B \leq b) = P(A \leq a), \text{ ce qui revient à :}$$

$$F_B(T(a)) = F_A(a) \text{ pour } a \in]a_{\min}, a_{\max}[$$

Or, on veut avoir $F_B = F_R$. Puisque F_R est inversible sur $]r_{\min}, r_{\max}[$, on prend :

$$T = F_R^{-1} \circ F_A \quad (4-5)$$

La méthode de spécification d'histogramme revient à appliquer à l'image **A** une transformation d'égalisation T_1 et à appliquer à l'image résultante **C**, l'inverse T_0^{-1} de la transformation d'égalisation de **R**.

La distribution d'intensité de l'image **B** est alors identique à celle de l'image **R**.

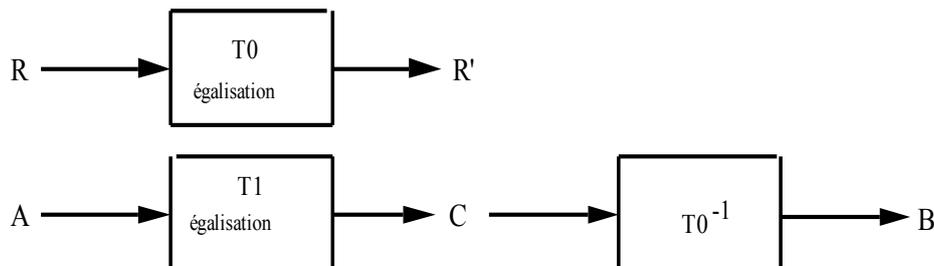


Figure 4.3 : Spécification d'histogramme

Compte tenu de la quantification des données, les histogrammes de **B** et de **R** sont, en général, légèrement différents. Toutefois, en faisant abstraction des formes représentées dans chacune des images, les aspects visuels de $B[i, j]$ et $R[i, j]$ sont semblables.

La figure 4.4 donne un exemple de spécification d'histogramme.

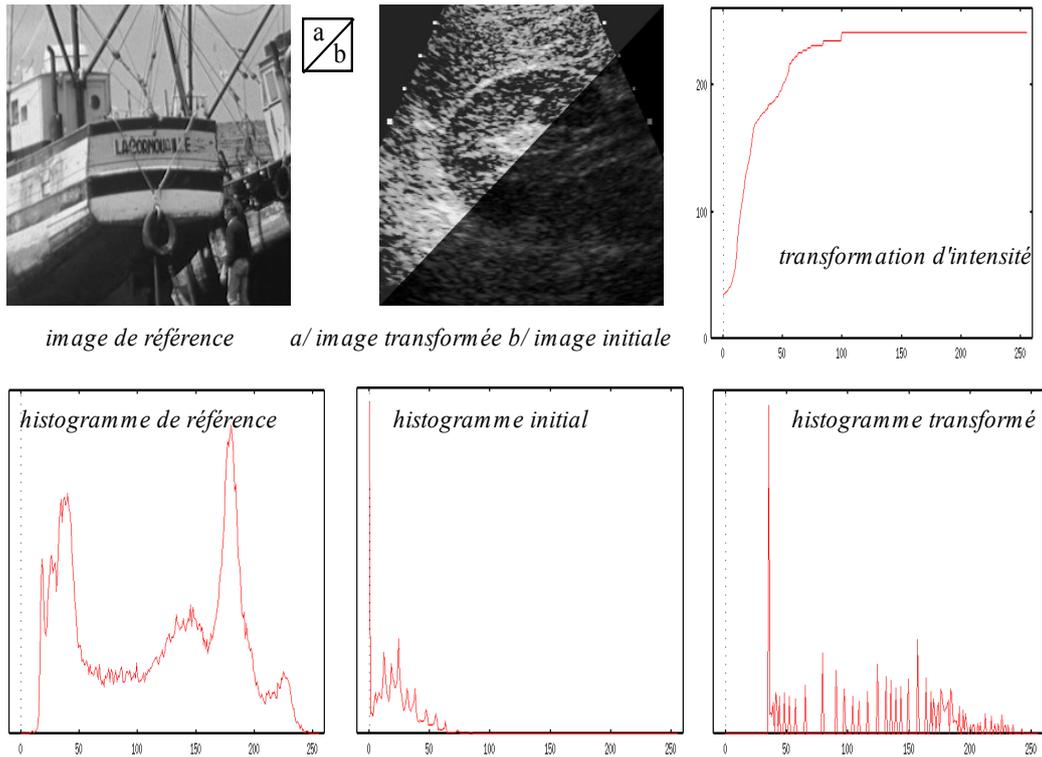


Figure 4.4 : Spécification d'histogramme

Cette transformation est utile par exemple lorsque, un réglage ayant été effectué, les conditions d'acquisition d'une série d'images fluctuent autour des conditions nominales ayant servi pour le réglage. La spécification permet de se ramener aux conditions nominales.

2 Réduction de bruit

Les régions formant l'image sont caractérisées par leur intensité moyenne. Les fluctuations autour de cette intensité moyenne peuvent provenir soit du dispositif d'acquisition (caméra, amplificateurs, quantification,...) soit de la scène elle-même (poussières, rayures,...). Elles constituent des perturbations qui affectent la qualité de la segmentation et sont généralement désignées sous le terme de *bruit d'image*. L'échelle spatiale des fluctuations est relativement faible par rapport aux dimensions des régions. Le bruit est donc plutôt de type haute fréquence.

Dans la plupart des cas, le bruit d'image est considéré comme étant aléatoire, centré et additif. Le prétraitement se ramène alors à un problème de traitement du signal : retrouver par filtrage les niveaux d'intensité nominaux de chacune des régions.

L'objectif est donc de réduire l'amplitude des variations d'intensité dans chaque région, tout en conservant les transitions entre régions adjacentes. Cette préoccupation apparaît dans les différentes approches de la segmentation, en particulier lors de l'emploi de techniques dérivatives. Pour cette raison, certaines présentations de ce paragraphe seront reprises sous une forme voisine dans le chapitre V. Les propriétés des filtres appliqués à l'image sont étudiées en tenant compte des objectifs du prétraitement. On s'intéresse donc essentiellement à la variance en sortie des filtres ainsi qu'à leur effet sur les transitions correspondant aux frontières entre régions.

Les méthodes les plus simples et les plus faciles à implanter sont fondées sur le filtrage linéaire stationnaire (invariant par translation). Compte tenu des limitations de cette approche, on présentera également quelques méthodes basées sur des opérateurs non linéaires ou adaptatifs. Les cas de bruits multiplicatifs ou convolutifs seront succinctement abordés.

2.1 Modèles de bruit d'image

En général, le bruit d'image est considéré comme un champ aléatoire. On le caractérise au premier ordre par sa densité de probabilité f ou sa fonction de répartition F , au deuxième ordre par la densité conjointe ou bien par sa fonction de corrélation. En pratique, la caractérisation du bruit va rarement au delà du deuxième ordre. Le cas des champs markoviens est traité plus particulièrement aux chapitres VII et XI. Les méthodes présentées le sont alors sous l'angle de la segmentation. Elles peuvent naturellement être employées dans un but de restauration d'image.

La nature plus ou moins *impulsionnelle* du bruit peut être décrite à l'aide de densités de probabilité de la forme :

$$f(a) = C \cdot \exp(-K|a|^\alpha) \quad (4-6)$$

Pour $\alpha = 2$, on retrouve le bruit *gaussien* et pour $\alpha = 1$ le bruit *exponentiel*. Le caractère impulsionnel du bruit est lié à la probabilité d'avoir des valeurs aberrantes (par exemple, des intensités s'écartant de plus de trois écarts-types de l'espérance mathématique). Le tableau 4.1 donne la probabilité d'un tel événement pour des distributions gaussiennes ($\alpha = 2$), exponentielles ($\alpha = 1$) et uniformes ($\alpha \rightarrow \infty$).

bruit	uniforme	gaussien	exponentiel
probabilité	0.0	0.003	0.015

Tableau 4.1 : Probabilité d'obtention de valeurs aberrantes

Dans la plupart des cas, on considère le bruit comme additif. L'image à traiter **A** peut être mise sous la forme :

$$A = g + B \quad (4-7)$$

où **g** représente l'information utile et **B** un champ aléatoire.

Il est parfois plus efficace d'adopter un modèle de bruit multiplicatif par exemple en présence de speckle dans des images radar ou de grain sur des films radiographiques. L'image se met alors sous la forme :

$$A = g \cdot B \quad (4-8)$$

Enfin, on pourra aussi considérer le modèle de bruit convolutif, utile pour décrire les dégradations dues à un effet de bougé, de mauvaise mise au point voire de turbulence atmosphérique. L'image à analyser est alors :

$$A = g * B \quad (4-9)$$

2.2 Filtrés linéaires stationnaires

Ces opérateurs sont caractérisés par leur réponse impulsionnelle $h(x, y)$. La relation entrée-sortie est décrite par l'équation de convolution

$$C(x, y) = A * h(x, y) = \iint A(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \quad (4-10)$$

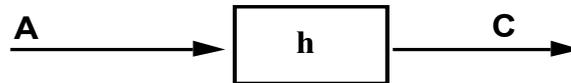


Figure 4.5 : Filtrage linéaire stationnaire (cas continu)

Dans le cas d'une image numérique, l'équation (4-10) est remplacée par la relation de convolution discrète :

$$B_{i,j} = \sum_m \sum_n h_{m,n} A_{i-m,j-n} = \sum_m \sum_n A_{m,n} h_{i-m,j-n} \quad (4-11)$$

Le support du noyau de convolution h peut être borné ou non borné. Comme on accorde *a priori* la même importance aux pixels situés d'un côté ou de l'autre du pixel courant, les noyaux de convolution utilisés sont généralement symétriques par rapport à chacune des coordonnées.

2.2.1 Cas continu

Pour simplifier, on se place dans le cas d'une image $A(x, y)$ dont l'information utile est un échelon unitaire vertical $U(x, y)$ perturbée par un bruit blanc stationnaire centré $B(x, y)$ de densité spectrale N_0^2 :

$$U(x, y) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases} \quad (4-12)$$

$$A(x, y) = U(x, y) + B(x, y) \quad (4-13)$$

Cette image est filtrée par un opérateur de réponse impulsionnelle h .

La sortie est donnée par :

$$C(x, y) = h * A(x, y) = h * U(x, y) + h * B(x, y) = D(x, y) + N(x, y) \quad (4-14)$$

$N(x, y)$ est une perturbation aléatoire dont on voudra minimiser la variance. Le bruit B étant centré, il en va de même pour N . On a donc :

$$E\{C(x, y)\} = D(x, y) \quad (4-15)$$

$D(x, y)$ représente la composante déterministe que l'on veut proche de l'échelon initial $U(x, y)$. Son analyse permet d'évaluer l'aptitude du filtre à préserver les transitions. Lorsque la fenêtre d'analyse est incluse dans une région de l'image non bruitée, d'intensité constante I_0 , la sortie du filtre est :

$$d = I_0 \iint h(x, y) dx dy \quad (4-16)$$

Afin de ne pas modifier l'intensité moyenne des régions de l'image et donc en conserver les différences, on choisira h telle que :

$$\iint h(x, y) dx dy = 1 \quad (4-17)$$

La variance du bruit en sortie du filtre est donnée par :

$$\text{var}(N) = E\{N^2(x, y)\} = \iint S_{NN}(u, v) dudv = \iint |\hat{h}(u, v)|^2 S_{BB}(u, v) dudv \quad (4-18)$$

où S_{BB} est la densité spectrale de puissance du bruit \mathbf{B} , S_{NN} celle de N

$\hat{h}(u, v)$ est la transformée de Fourier de la réponse impulsionnelle $h(x, y)$.

Le bruit d'image étant de type large bande, on pourra l'assimiler à un bruit blanc de densité spectrale N_0^2 . Dans ces conditions, en utilisant la relation de Parseval, l'équation (4-18) devient :

$$\text{var}(N) = N_0^2 \iint h^2(x, y) dx dy \quad (4-19)$$

2.2.2 Exemples de filtres réducteurs de bruit

a) filtre moyenneur

La réponse impulsionnelle d'un tel filtre est constante dans le pavé $[-t/2, t/2] \times [-t/2, t/2]$ et vaut :

$$h(x, y) = \frac{1}{t^2} \quad (4-20)$$

Lorsque l'image est perturbée par un bruit blanc de densité spectrale N_0^2 , en appliquant la relation (4-18), on obtient la variance du bruit filtré :

$$\text{var}(N) = \frac{N_0^2}{t^2} \quad (4-21)$$

La composante déterministe du signal de sortie vaut alors :

$$D(x, y) = \begin{cases} 0 & \text{si } x \leq -\frac{t}{2} \\ \frac{1}{2} + \frac{x}{t} & \text{si } x \in \left] -\frac{t}{2}, +\frac{t}{2} \right[\\ 1 & \text{si } x \geq \frac{t}{2} \end{cases} \quad (4-22)$$

Il en résulte un élargissement de la zone de transition (cf. figure 4.6)

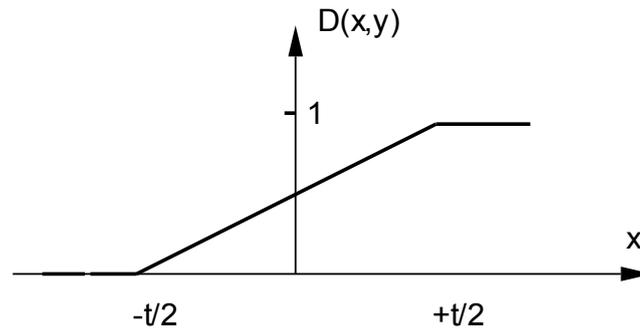


Figure 4.6 : Elargissement de la zone de transition

b) filtre gaussien

Ce filtre a été particulièrement étudié par D. Marr [MARR80], sa réponse impulsionnelle est :

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (4-23)$$

Pour une image initiale \mathbf{A} comportant un échelon unitaire vertical perturbé par un bruit blanc de densité spectrale N_0^2 , la variance du bruit filtré est :

$$\text{var}(N) = \frac{N_0^2}{4\pi\sigma^2} \quad (4-24)$$

La composante déterministe $D(x, y)$ vaut :

$$D(x, y) = \operatorname{erf}\left(\frac{x}{\sigma}\right) \quad (4-25)$$

où erf désigne la primitive de la fonction $\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ valant 1/2 à l'origine.

Le principal avantage du filtre gaussien est qu'il présente le meilleur compromis entre localisation spatiale et fréquentielle (*cf.* annexe D).

c) filtre exponentiel

Ce filtre est présenté en détail dans le chapitre V, sa réponse impulsionnelle est donnée par [SHEN85] :

$$h(x, y) = \frac{\beta^2}{4} \exp(-\beta(|x| + |y|)) \quad (4-26)$$

Pour une image initiale \mathbf{A} comportant un échelon unitaire vertical perturbé par un bruit blanc de densité spectrale N_0^2 , la variance du bruit filtré est :

$$\operatorname{var}(N) = \frac{\beta^2}{16} N_0^2 \quad (4-27)$$

et la composante déterministe :

$$D(x) = \begin{cases} \frac{1}{2} e^{\beta x} & \text{si } x \leq 0 \\ 1 - \frac{1}{2} e^{-\beta x} & \text{si } x > 0 \end{cases} \quad (4-28)$$

Pour chacun de ces deux derniers opérateurs, on peut diminuer la puissance du bruit en augmentant le paramètre σ ou en diminuant le paramètre β . Ceci se fait au détriment de la netteté des transitions dans l'image de sortie.

2.2.3 Cas discret

L'évaluation des performances d'un filtre linéaire se fait, là encore, en considérant un modèle d'image simplifié comportant un échelon vertical perturbé par un bruit additif.

$$A_{i,j} = U_{i,j} + B_{i,j} \quad (4-29)$$

où : $U[i, j] = \begin{cases} 0 & \text{pour } i < 0 \\ 1 & \text{sinon} \end{cases}$ et $B[i, j]$ représente un bruit aléatoire stationnaire centré et indépendant de U .

Soit $h[m, n]$ la réponse impulsionnelle du filtre.

L'image filtrée comporte une composante déterministe :

$$D_{i,j} = \sum_m \sum_n h_{m,n} U_{i-m,i-n} \quad (4-30)$$

et une composante aléatoire :

$$N_{i,j} = \sum_m \sum_n h_{m,n} B_{i-m,i-n} \quad (4-31)$$

Comme précédemment, on impose au filtre de ne pas altérer les intensités moyennes des régions. Ceci conduit à :

$$\sum_m \sum_n h_{m,n} = 1 \quad (4-32)$$

Si B est un bruit blanc de variance N_0^2 , la variance du bruit en sortie a pour valeur :

$$\text{var}(N) = N_0^2 \sum_m \sum_n h_{m,n}^2 \quad (4-33)$$

Pour un filtre à réponse impulsionnelle finie, on calcule de façon simple les coefficients optimaux au sens de la variance de sortie. L'équation (4-31) s'écrit alors :

$$N_{i,j} = \sum_m \sum_n h_{-m,-n} B_{i+m,i+n} \quad (4-34)$$

Pour alléger l'écriture, on procède à une renumérotation, de 1 à L , des pixels de la fenêtre d'analyse. Pour un pixel donné $[i, j]$, la sortie apparaît donc comme un produit scalaire entre un vecteur de coefficients \mathbf{h} et un vecteur de données \mathbf{b} .

1	2			
	k			
				L

Figure 4.7 : Renumerotation des pixels de la fenêtre d'analyse

La sortie est alors donnée par :

$$N_{i,j} = \sum_{k=1}^L h_k b_k = {}^t \mathbf{h} \mathbf{b} \quad (4-35)$$

et la variance vaut :

$$\text{var}(N) = {}^t \mathbf{h} \mathbf{R} \mathbf{h} \quad (4-36)$$

où $\mathbf{R} = E\{\mathbf{b}'\mathbf{b}\}$ est la matrice de covariance du vecteur bruit.

La contrainte exprimée dans l'équation (4-32) peut se mettre sous la forme

$${}^t\mathbf{h}\mathbf{e} = 1 \quad (4-37)$$

avec : ${}^t\mathbf{e} = (1, 1, \dots, 1)$

Minimiser $\text{var}(N)$ sous la contrainte (4-37) aboutit à la solution optimale :

$$\mathbf{h}_{\text{opt}} = \frac{\mathbf{R}^{-1}\mathbf{e}}{{}^t\mathbf{e}\mathbf{R}^{-1}\mathbf{e}} \quad (4-38)$$

Dans le cas d'un bruit blanc, la matrice \mathbf{R} vaut $N_0^2 \mathbf{I}_d$ où \mathbf{I}_d est la matrice identité et donc $\mathbf{R}^{-1} = \frac{1}{N_0^2} \mathbf{I}_d$. Les coefficients de la réponse impulsionnelle sont tous égaux et valent chacun $1/L$ pour satisfaire la contrainte. On retrouve le filtre moyennneur sous sa forme discrète.

2.2.4 Problèmes liés à l'implantation des opérateurs

Compte tenu du volume important des données, on cherche à implanter ces opérateurs de manière peu coûteuse en temps calcul et taille mémoire, et si possible de façon parallélisable.

Pour un filtre à nombre fini de coefficients, l'image en sortie est décrite par la relation :

$$C_{i,j} = \sum_{m=-M}^M \sum_{n=-N}^N h_{m,n} A_{i-m,j-n} \quad (4-39)$$

Le calcul de l'intensité pour chaque pixel de l'image filtrée nécessite $(2M+1)(2N+1)$ multiplications-additions. Si le calcul s'effectue en balayant l'image ligne par ligne, il est nécessaire de stocker temporairement au moins $(2M+1)$ lignes de l'image initiale.

Si la réponse impulsionnelle du filtre peut se factoriser sous la forme :

$$h_{m,n} = h_{1m} \cdot h_{cn} \quad (4-40)$$

l'équation (4-39) s'écrit :

$$C_{i,j} = \sum_{m=-M}^M h_{1m} \sum_{n=-N}^N h_{cn} A_{i-m,j-n} = \sum_{m=-M}^M h_{1m} A_{ci-m,j} \quad (4-41)$$

L'image \mathbf{C} s'obtient par filtrage linéaire selon les colonnes par h_c , puis filtrage linéaire selon les lignes par h_1 (cf. figure 4.8)

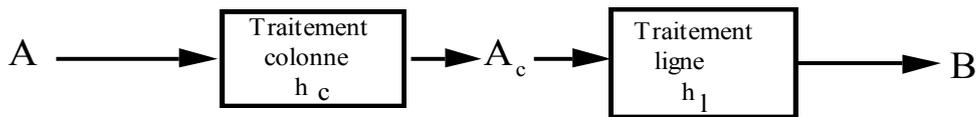


Figure 4.8 : Filtrage séparable

On parle alors de *filtrage séparable*. Le calcul de chaque pixel de l'image C nécessite $2(M+N+1)$ multiplications-additions et le stockage temporaire d'une ligne (ou colonne) de l'image traitée.

On peut vérifier facilement que les opérateurs définis précédemment (moyenneur, gaussien, exponentiel) sont séparables. La réalisation des filtres à réponse impulsionnelle infinie pose des problèmes classiques traités, par exemple, dans [BELL84]. Certains, comme les filtres exponentiels peuvent être implantés sous forme récursive (*cf.* annexe E). Ce n'est pas le cas du filtre gaussien dont la réponse impulsionnelle doit être approximée ou tronquée.

L'effet d'un filtrage linéaire peut être apprécié sur les images de la figure 4.9 (le filtre moyenneur utilisé est de taille : 15×15).

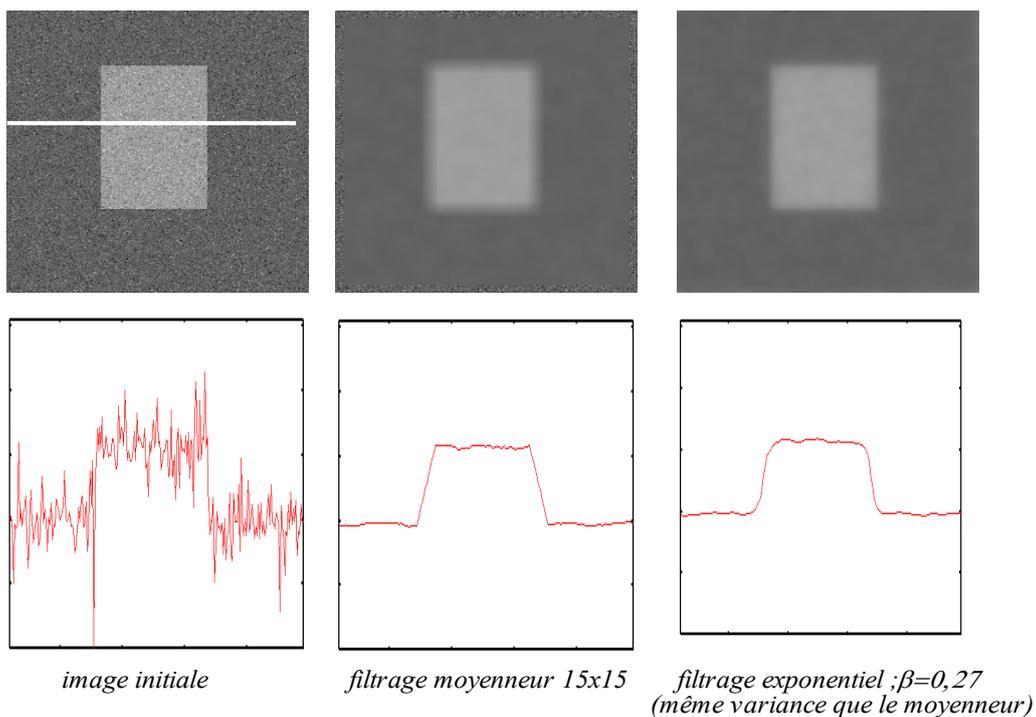


Figure 4.9 : Filtrage linéaire moyenneur et exponentiel

L'image initiale représente un objet rectangulaire posé sur un fond sombre. On remarque, dans l'image filtrée, l'effet de flou sur les transitions entre zones claires et sombres. Avec le filtre à réponse impulsionnelle exponentielle, le gradient est plus fort au niveau de la transition, ce qui limite l'effet de flou.

2.3 Filtres non-linéaires stationnaires

Le principal inconvénient des filtres linéaires est que la réduction du bruit s'accompagne d'un étalement des transitions entre régions. La détermination de leurs coefficients résulte d'un compromis. Cette difficulté peut être surmontée par l'utilisation de filtres non linéaires. Par contre, ces derniers introduisent généralement des modifications irréversibles de l'image. Ces transformations pourront affecter la perception visuelle de l'image. Cependant, elles en constituent une simplification, rendant ainsi plus aisée l'étape de segmentation.

Le domaine du filtrage non-linéaire est extrêmement vaste. Les opérateurs les plus courants en prétraitement d'image sont les filtres d'ordre, les filtres homomorphiques et les filtres morphologiques.

2.3.1 Filtres d'ordre

a) Principe

On considère une fenêtre rectangulaire de taille $(2M+1) \times (2N+1)$ pixels, centrée sur le pixel $[i, j]$ d'une image \mathbf{A} . En numérotant de 1 à $L = (2M+1) \times (2N+1)$ les pixels de cette fenêtre (cf. figure 4.7), on définit l'ensemble des données disponibles :

$$\mathbf{a} = \{a_k | k = 1 \dots L\} \quad (4-42)$$

Puis on trie les éléments de \mathbf{a} par ordre croissant. On appelle $j^{\text{ème}}$ statistique d'ordre, notée $a_{(j)}$, la valeur de rang j dans \mathbf{a} .

$$a_{(1)} \leq a_{(2)} \leq \dots \leq a_{(j)} \leq a_{(j+1)} \dots \leq a_{(L)} \quad (4-43)$$

Pour un filtre d'ordre, la sortie est définie comme une fonction des statistiques d'ordre. Les données sont prises en compte selon leur rang dans la fenêtre d'analyse et non selon leur disposition spatiale.

La caractérisation statistique des filtres d'ordre est faite en considérant des bruits à distribution continue, le plus souvent de type bruit blanc. Dans ces conditions, les densités de probabilité sont données par :

- densité de la statistique de rang r :

$$f_{(r)}(x) = \frac{L!}{(r-1)!(L-r)!} F^{r-1}(x) [1-F(x)]^{L-r} f(x) \quad (4-44)$$

- densité conjointe des statistiques de rang r et p ($r < p$) :

$$f_{(r)(p)}(x, y) = K(r, p) F^{r-1}(x) [F(y) - F(x)]^{p-r-1} [1-F(y)]^{L-p} f(x) f(y) \quad (4-45)$$

$$\text{avec } K(r, p) = \frac{L!}{(r-1)!(p-r-1)!(L-r)!}$$

pour $x \leq y$

- densité du vecteur des statistiques d'ordre :

$$f_{(1)..(L)}(x_1, \dots, x_L) = L! f(x_1) \dots f(x_L) \quad (4-46)$$

pour $x_1 \leq x_2 \leq \dots \leq x_L$

b) L-filtre

La sortie $C[i, j]$ d'un L-filtre [DAVI81] s'obtient par combinaison linéaire des statistiques d'ordre (cf. figure 4.10) :

$$C[i, j] = \sum_{k=1}^L c_k a_{(k)} \quad (4-47)$$

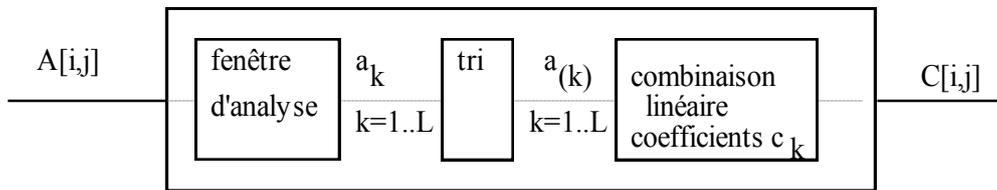


Figure 4.10 : L-filtre

Afin de ne pas modifier l'intensité des zones homogènes de l'image, les coefficients doivent satisfaire la contrainte :

$$\sum_{k=1}^L c_k = 1 \quad (4-48)$$

Dans le cas où l'image est homogène sans transition et localement modélisable par un bruit blanc de densité f et de fonction de répartition F , il est possible d'optimiser le choix des coefficients c_k au sens d'un critère d'erreur quadratique [LLOY52]. Si la densité de probabilité f est symétrique par rapport à la valeur moyenne, que l'on peut supposer nulle, on obtient à l'aide d'une méthode utilisant les multiplicateurs de Lagrange [BOVI83] :

$$\mathbf{c} = \frac{\mathbf{R}^{-1}}{{}^t\mathbf{e} \mathbf{R}^{-1} \mathbf{e}} \quad (4-49)$$

avec : ${}^t\mathbf{c} = c_1, \dots, c_L$; ${}^t\mathbf{e} = 1, 1, \dots, 1$

\mathbf{R} est la matrice de corrélation du bruit après tri. Le terme général R_{rp} est donné, pour $r \neq p$ par :

$$R_{rp} = \iint xy f_{(r)(p)}(x, y) dx dy \quad (4-50)$$

et pour $r = p$ par :

$$R_{pp} = \int x^2 f_{(p)}(x) dx \quad (4-51)$$

La variance du signal de sortie du filtre est alors : $\mathbf{t} \mathbf{c} \mathbf{R} \mathbf{c}$.

On peut montrer [LLOY52][DAVI81] que, dans ces conditions, la puissance du bruit en sortie du L-filtre optimal est toujours inférieure (ou au pire égale) à celle du meilleur filtre linéaire (filtre moyennneur). Les valeurs des coefficients optimaux dépendent de la forme de la densité f .

Pour un bruit gaussien de variance N_0^2 , on obtient :

$$c_k = \frac{1}{L} \quad \forall k \quad (4-52)$$

On retrouve le filtre moyennneur dont la variance de sortie est :

$$\frac{1}{L} N_0^2$$

Pour un bruit uniforme, on obtient :

$$c_k = \begin{cases} 1/2 & \text{si } k = 1 \text{ ou } k = L \\ 0 & \text{pour } 1 < k < L \end{cases} \quad (4-53)$$

Il s'agit du *filtre milieu*, la variance de sortie vaut alors :

$$\frac{6}{(L+1)(L+2)} N_0^2 \quad (4-54)$$

Les L-filtres les plus fréquemment utilisés sont :

le filtre médian :

$$c_k = \begin{cases} 1 & \text{si } k = \frac{L+1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (4-55)$$

Ce filtre fait l'objet du paragraphe suivant.

les filtres à moyenne tronquée :

$$c_k = \begin{cases} \frac{1}{2p+1} & \text{si } k = \frac{L+1}{2} - p, \dots, \frac{L+1}{2} + p \\ 0 & \text{ailleurs} \end{cases} \quad (4-56)$$

c) Filtre médian

Comme son nom l'indique, le filtre médian sélectionne le pixel de la fenêtre d'analyse ayant la valeur médiane.

Dans une zone stationnaire perturbée par un bruit blanc à distribution continue, la densité de probabilité de la sortie, notée $g(c)$, est égale à :

$$g(c) = \frac{L!}{q!q!} F^q(c) \cdot [1 - F(c)]^q \cdot f(c) \quad (4-57)$$

f est la densité de probabilité parente, F la fonction de répartition et $L = 2q+1$.

Ce filtre est insensible à la présence de valeurs aberrantes si leur nombre est inférieur à $L/2$. Il est donc très utile dans le cas de perturbations à caractère impulsif, ou ayant une densité s'étalant vers l'infini (le bruit exponentiel, par exemple). De plus, le filtre médian a la propriété de laisser inchangées les transitions monotones entre régions [GALL81]. Ces deux propriétés, ajoutées à l'existence d'algorithmes rapides et de dispositifs temps réel [ATAM80] [HUAN79] [OCHO87] [RICH90], en font un opérateur très employé. Il présente néanmoins quelques inconvénients. Pour des bruits à distribution assez concentrée (gaussien, uniforme), ses performances sont faibles par rapport au filtre d'ordre optimal ou même au filtre linéaire optimal. Par ailleurs, il peut affecter la géométrie des régions de l'image. Les zones formant un angle aigu ont tendance à être arrondies par le filtrage (*cf.* figure 4.1). On perd ainsi l'information sur les points anguleux. Différentes améliorations ont été proposées pour pallier ces inconvénients. Elles consistent en fait à introduire des informations spatiales dans le fonctionnement du filtre, par le biais de traitements récursifs, de coefficients de pondération ou d'opérateurs multiétages.

filtres multiétages

Ces opérateurs comportent généralement deux étages. Le premier étage est constitué de filtres agissant directement sur les données de l'image. Le deuxième et les suivants traitent les sorties des filtres de l'étage précédent.

Pour le filtre max/médian [ARCE87], le premier étage est formé de filtres médians 1D agissant dans les directions horizontale, verticale et diagonales. Le deuxième étage sélectionne le maximum des valeurs obtenues. Ce filtre préserve la géométrie des zones angulaires mais il introduit un biais sur l'intensité moyenne des régions. Pour éviter ce phénomène, d'autres opérateurs tel que le filtrage multimédian [NIEM88] ou max-min/médian [WANG90] peuvent être utilisés. Il faut cependant souligner que l'amélioration des propriétés géométriques se fait au détriment de l'efficacité en lissage de bruit. Pour améliorer l'effet de réduction de bruit, on peut alors remplacer les filtres médians du premier étage par des filtres médians récursifs [BOLO91] ou bien, lorsque le bruit d'image a une distribution assez concentrée (gaussienne ou uniforme), par des filtres linéaires 1D, ce qui conduit aux structures dites de filtres médians hybrides [HEIN87].

filtre médian pondéré

Cet opérateur est défini de la façon suivante : pour une fenêtre d'analyse telle que celle décrite en figure 4.7, la sortie du filtre est :

$$C[i, j] = \text{median} \left\{ \overbrace{a_1, \dots, a_1}^{w_1 \text{ fois}}, \overbrace{a_2, \dots, a_2}^{w_2 \text{ fois}}, \dots, \overbrace{a_L, \dots, a_L}^{w_L \text{ fois}} \right\} \quad (4-58)$$

Formellement, cet opérateur minimise l'erreur absolue pondérée [PITA90] :

$$\varepsilon(c) = \sum_{k=1}^L w_k |c - a_k|$$

Pour un choix approprié des coefficients entiers w_k , en particulier par la répartition des coefficients nuls, on peut privilégier ou éliminer des structures ayant une direction donnée [BROW84] [SUN94]. Il est à noter que le meilleur effet de réduction de bruit est obtenu quand tous les coefficients w_k sont égaux à 1 (médian standard).

La figure 4.11 illustre les performances du filtrage médian. On remarque, sur les deux images filtrées, que les principales transitions entre régions sombres et claires sont correctement préservées. Par contre, les structures de taille réduite, telles que les petites zones claires situées entre cellules sombres, sont supprimées par le filtrage médian et mieux conservées par le filtre max/médian. Pour ce dernier, l'effet de lissage est un peu plus faible que pour le médian standard.

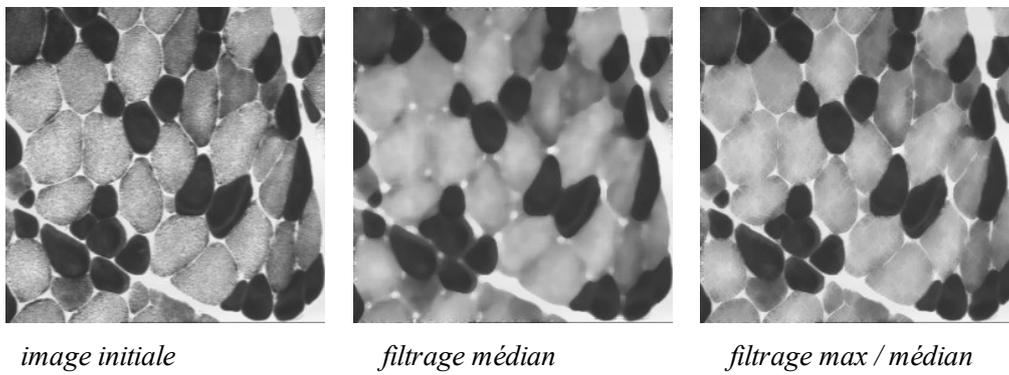


Figure 4.11 : Filtrage médian et max / médian. Taille du filtre 7×7

d) *Filtre de rang (« rank order filter »)*

La sortie d'un filtre de rang r est définie par (4-59), en utilisant les notations (4-43) :

$$C[i, j] = A_{(r)} \quad 1 \leq r \leq L, \text{ } r \text{ fixé} \quad (4-59)$$

Les filtres de rang sont des cas particuliers de L-filtres. Ils ont un effet dissymétrique, ce qui est parfois souhaitable. Ils sont surtout utiles pour l'élimination de perturbations impulsionnelles. Un filtre de rang r élimine une impulsion négative formée de moins de r pixels. Il élimine également toute impulsion positive formée d'au plus $L-r$ pixels. Le principal inconvénient de cet opérateur est qu'il introduit un biais qui est d'autant plus important que la taille L du filtre est élevée, que le rang retenu r s'écarte de la médiane $(L+1)/2$ et que la variance du bruit est forte.

Pour $r = 1$ et $r = L$, on retrouve les opérations morphologiques d'érosion et de dilatation (cf. § 2.3.3)

e) *Filtre de type « K plus proches voisins »*

En reprenant les notations (4-43), on définit parmi les éléments de \mathbf{a} les K éléments $a^*_{(k)}$ minimisant la quantité [KIM86] :

$$\sum_p^{p+K-1} |A_{i,j} - a^*(k)| \quad (4-60)$$

Les éléments $a^*(k)$ sont appelés les K plus proches voisins de $A[i, j]$.

La sortie du filtre est donnée par :

$$C_{i,j} = \frac{1}{K} \sum_{i=1}^K a^*(k) \quad (4-61)$$

Une variante consiste à prendre les K valeurs les plus proches de $A[i, j]$ par le rang et d'en calculer la médiane, au lieu de la moyenne arithmétique [ITO88]. Ce type d'opérateur vise à éliminer le bruit impulsionnel, à conserver les transitions entre régions, tout en renforçant le poids accordé au pixel en cours de traitement. Le choix de la valeur du paramètre K est lié à ce que l'utilisateur considère comme étant la taille des structures significatives dans l'image.

f) M-filtres

Les M-filtres sont définis par l'intermédiaire d'une fonction de coût φ , généralement continue et paire. Plus précisément, en reprenant les notations (4-42) la sortie du filtre est, pour un ensemble de données \mathbf{a} :

$$C[i, j] = \arg \min \sum_{k=1}^L \varphi(t - a_k) \quad (4-62)$$

Le M-filtre standard [LEE85] est la version « filtrage » de l'estimateur robuste introduit par Huber [HUBE64]. Il est caractérisé par la fonction de coût suivante :

$$\varphi(t) = \begin{cases} -b.t - \frac{b^2}{2} & t < -b \\ \frac{t^2}{2} & |t| \leq b \\ b.t - \frac{b^2}{2} & t > b \end{cases} \quad (4-63)$$

Lorsque les données sont concentrées dans un intervalle de largeur $2b$, le filtre se comporte comme un moyenneur. Inversement, si elles sont très dispersées, il agit comme un médian, ce qui lui confère sa robustesse.

Le filtrage $d\alpha$ [BOLO92], ou sa version approchée intitulée γ -filtre [ASTO89], constitue une autre classe de M-filtres. La fonction de coût est :

$$\varphi(t) = |t|^\alpha \quad (4-64)$$

Pour $\alpha=1$, on retrouve le filtrage médian et pour $\alpha=2$ le filtre moyenneur. Quand α tend vers l'infini, le filtre $d\alpha$ tend vers le filtre milieu. La structure de ce filtre est celle d'un estimateur du maximum de vraisemblance dans le cas de bruits

blancs additifs ayant une densité décrite par l'équation (4-6). Pour des images très bruitées, on privilégie l'effet de réduction de bruit en choisissant une valeur de α adaptée au caractère plus ou moins impulsionnel du bruit. Par contre, si le niveau de bruit est modéré, il faut prendre en compte l'effet du filtrage sur les transitions entre régions. Un bon compromis consiste à choisir une valeur de α comprise entre 1.5 et 2. L'effet de réduction de bruit est voisin de celui d'un moyenneur mais l'élargissement de la zone de transition est moindre.

2.3.2 Filtres homomorphiques

Le filtrage homomorphique est réalisé par une combinaison de traitements linéaires et non linéaires. Il permet souvent de se ramener à un problème de lissage de bruit additif.

Le schéma de principe d'un filtre homomorphique est donné dans la figure 4.12.

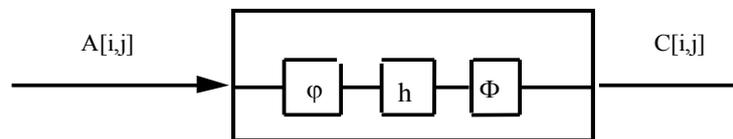


Figure 4.12 : Schéma de principe d'un filtre homomorphique

φ et Φ sont des opérateurs non-linéaires réciproques. h est la réponse impulsionnelle d'un filtre linéaire généralement de type passe-bas.

On obtient des moyennes non-linéaires en faisant les choix suivants :

- Pour $a \geq 0$ et $p \neq 0$, on prend $\varphi(a) = a^p$ et $\Phi(a) = a^{1/p}$, h est un filtre moyenneur.
- Pour $p = 1$, on retrouve le filtre moyenneur ; $p = -1$ produit la moyenne harmonique. En prenant la fonction logarithme népérien (resp. exponentielle) pour φ (resp. Φ), on obtient la moyenne géométrique.

La relation entre l'entrée A et la sortie C est non-linéaire, mais elle peut s'exprimer analytiquement. Il est alors possible de déterminer, par calcul, les propriétés statistiques de la sortie du filtre, au moyen de développements en série de Taylor. On peut montrer ainsi que la variance de sortie de ce filtre peut être rendue inférieure à celle du filtre médian pour des bruits à distribution gaussienne ou uniforme, ou inférieure à celle du filtre moyenneur pour un bruit uniforme [PITA90]. Des valeurs positives de p permettent d'atténuer ou de supprimer les impulsions négatives. Les valeurs négatives de p ont le même effet sur les impulsions positives.

On obtient des filtres de type « cepstres » en définissant l'opérateur φ par $\varphi = TF^{-1} \circ LOG \circ TF$ et l'opérateur Φ par $\Phi = TF^{-1} \circ EXP \circ TF$; dans ces expressions, TF désigne l'opérateur de transformation de Fourier, LOG le logarithme complexe et EXP l'exponentielle complexe. Les opérateurs cepstres

sont utilisés en présence de bruits convolutifs dus, par exemple, à des effets de bougé ou de turbulence [KUNT93].

Notons \mathbf{g} l'image sans défaut, \mathbf{B} le bruit convolutif et \mathbf{A} l'image disponible donnée par :

$$\mathbf{A} = \mathbf{g} * \mathbf{B} \quad (4-65)$$

Alors en posant $\hat{\mathbf{A}} = TF\{\mathbf{A}\}$, $\bar{\mathbf{A}} = TF^{-1}\left\{LOG\left(\hat{\mathbf{A}}\right)\right\}$ et en gardant les mêmes

notations pour les termes \mathbf{g} et \mathbf{B} , on a en sortie de l'opérateur ϕ :

$$\phi(\mathbf{A}) = \bar{\mathbf{A}} = \bar{\mathbf{g}} + \bar{\mathbf{B}} \quad (4-66)$$

Dans le cas idéal, on peut éliminer le terme de bruit $\bar{\mathbf{B}}$. La sortie de l'opérateur cepstre est, dans ces conditions :

$$\mathbf{C} = \phi(\bar{\mathbf{g}}) = TF^{-1}\left\{\exp\left(TF\{\bar{\mathbf{g}}\}\right)\right\} = \mathbf{g} \quad (4-67)$$

En pratique, on cherche à atténuer le bruit $\bar{\mathbf{B}}$ en se basant sur ses propriétés spectrales si elles sont connues ou bien, en se fondant sur des hypothèses de stationnarité et d'ergodisme, en analysant un ensemble de sous-images de taille réduite [KUNT93].

Les filtres homomorphiques sont également utilisés lorsque le bruit est multiplicatif (cf. § 2.5).

2.3.3 Filtrage morphologique

L'objectif de la phase de prétraitement est de simplifier l'image de façon à faciliter sa segmentation. Partant d'un modèle des perturbations (bruit d'image), l'approche classique consiste à atténuer ou supprimer le bruit pour retrouver l'information utile sous-jacente. Inversement, lorsque l'on dispose d'informations *a priori* sur la forme et la taille des structures à localiser, ainsi que sur le « signe » du bruit (impulsions positives ou négatives), on peut chercher à renforcer cette caractéristique par un traitement dissymétrique (en terme de voisinage et d'intensité) fondé sur les techniques de la morphologie mathématique (cf. annexe C). En terme de réduction de bruit, l'ouverture morphologique supprime les impulsions positives et préserve les impulsions négatives. L'effet inverse est obtenu par fermeture.

Une transformation vérifiant : $\forall A, T\{A\} = T\{T\{A\}\}$ est dite *idempotente*. Une transformation vérifiant : $A(s) \leq B(s) \Rightarrow T\{A\}(s) \leq T\{B\}(s)$ est dite *croissante*. Une transformation croissante et idempotente est appelée *filtre morphologique*.

En utilisant un élément structurant plan \mathbf{g}_K associé à un compact \mathbf{K} , l'ouverture et la fermeture s'obtiennent, pour des images à support discret, par combinaison de filtres de rang. En effet, pour un site quelconque s , les valeurs de l'érodé $\mathbf{C} = \mathbf{A} \ominus \mathbf{g}_K$ et du dilaté $\mathbf{D} = \mathbf{A} \oplus \mathbf{g}_K$ sont données, en reprenant les notations (4-43) par :

$$\begin{aligned} C[s] &= a_{(l)} && \text{(filtre minimum)} \\ D[s] &= a_{(L)} && \text{(filtre maximum)} \end{aligned} \quad (4-68)$$

L'ouverture par g_K comprend l'érosion par g_K suivie de la dilatation par $\overset{\vee}{g}_K$ et inversement, la fermeture par g_K comprend la dilatation par g_K suivie de l'érosion par $\overset{\vee}{g}_K$.

L'ouverture et la fermeture morphologiques filtrent le bruit quelle que soit son amplitude relative, pourvu qu'elle soit positive pour la première et négative pour la seconde (et que le support de l'impulsion soit inclus dans le compact K). Ces deux opérateurs, pris séparément, induisent donc une modification de la moyenne des intensités, d'autant plus importante que la taille de l'élément structurant est grande et que l'amplitude du bruit est forte. Pour pallier cet inconvénient, on peut faire appel au filtrage alterné séquentiel défini, dans le cas discret, de la façon suivante :

Soit $\{O_i\}$ et $\{F_i\}$ une famille d'ouvertures et de fermetures associées à des éléments structurants convexes K_i de même forme et de taille croissante ($K_i \subset K_{i+1}$).

Le filtre alterné séquentiel est obtenu par :

$$FAS_i = O_i \circ F_i \circ O_{i-1} \circ F_{i-1} \circ \dots \circ O_1 \circ F_1 \quad (4-69)$$

ou, en permutant les opérations d'ouverture et de fermeture, par :

$$FAS'_i = F_i \circ O_i \circ F_{i-1} \circ O_{i-1} \circ \dots \circ F_1 \circ O_1 \quad (4-70)$$

Parmi les filtres morphologiques, le filtre alterné séquentiel est un de ceux qui ont les meilleures propriétés de réduction de bruit (*cf.* figure 4.13). Sur ces opérateurs, des développements mathématiques ou appliqués sont accessibles dans [MATH83], [SERR86], [SERR88], [COST89].

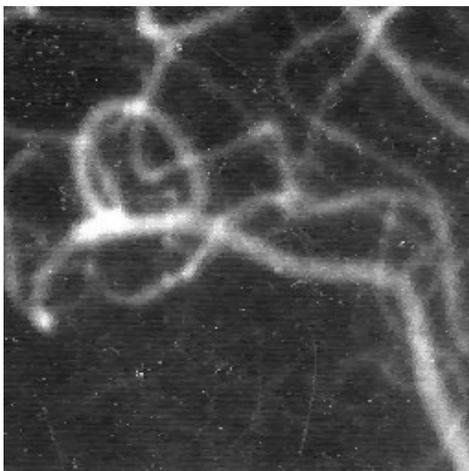


image initiale

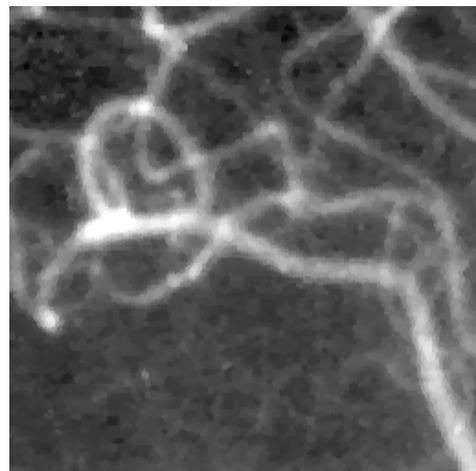


image filtrée par un compact hexagonal (FAS₂)

Figure 4.13 : Filtrage alterné séquentiel

2.4 Filtres adaptatifs

Lorsque l'on effectue un traitement en vue de la segmentation d'une image, deux types de situations sont rencontrés. Soit la fenêtre d'analyse est située à l'intérieur d'une région, et l'on a affaire à un signal stationnaire éventuellement perturbé par un bruit à caractère impulsionnel ou continu, soit elle inclut une transition entre deux régions (contour). Le principe du filtrage adaptatif est d'ajuster la structure, ou les coefficients, d'un opérateur à chacune de ces situations. Dans une zone stationnaire, on peut réduire le bruit en employant un filtre de taille relativement importante dont la structure dépendra du caractère impulsionnel ou non de la perturbation. Lorsque l'on est en présence d'un contour, il faut éviter l'effet d'élargissement de la zone de transition qui peut entraîner une perte d'information sur la géométrie des régions. Pour cela, on peut diminuer la taille du filtre ou rendre le comportement de ce dernier proche d'un filtrage médian. Comme précédemment, on recherche des opérateurs à structure simple. Un tel filtre adaptatif comporte un étage de décision et un étage de filtrage des données. L'étage de décision incorpore des connaissances *a priori* fournies par l'utilisateur. Par exemple, la différence entre un parasite impulsionnel formé de quelques pixels et une région de petite taille peut tenir au nombre de pixels, à l'amplitude de la transition, *etc.* Quelques exemples de filtres adaptatifs sont présentés dans les paragraphes suivants.

2.4.1 Moyennes adaptatives

Pour une fenêtre d'analyse de L pixels, telle que définie en figure 4.7, la sortie du filtre est donnée par :

$$C[i, j] = \frac{\sum_{k=1}^L w(a_k - A[i, j])a_k}{\sum_{k=1}^L w(a_k - A[i, j])} \quad (4-71)$$

$$\text{avec } w(x) = \begin{cases} 1 & \text{si } |x| \leq t \text{ (t : seuil préfixé)} \\ 0 & \text{sinon} \end{cases}$$

Dans une zone stationnaire, ce filtre agit comme un filtre moyenneur, avec de bonnes performances en terme de lissage si le bruit a une densité assez concentrée (gaussien ou uniforme par exemple). Lorsque la fenêtre d'analyse est située de part et d'autre d'une transition, l'opérateur ne tient compte, dans le calcul de la moyenne, que des pixels situés du même côté du contour que le pixel central. Ainsi une transition de type échelon ne sera pas élargie, sous réserve que son amplitude soit supérieure au seuil t . Dans [POMA84] les auteurs préconisent une valeur de seuil de l'ordre de trois fois l'écart-type du bruit, et obtiennent dans ces conditions de meilleures performances qu'avec un filtre médian de même taille.

L'opérateur de moyenne adaptative est, en fait, une version modifiée du « sigma-filter » [LEE83] pour lequel le seuil préconisé est fixé à deux fois l'écart-type du bruit et dont le principe est rappelé ci-dessous.

Soit N_e le nombre d'échantillons à utiliser dans le calcul de la moyenne.

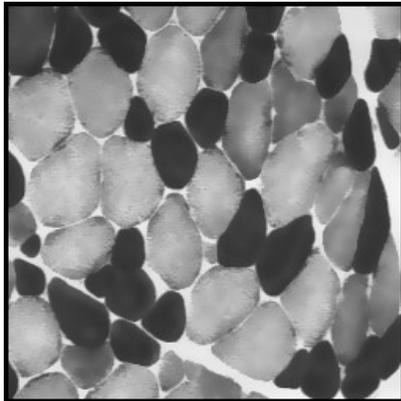
$$N_e = \sum_{k=1}^L w(a_k - A[i, j]) \quad (4-72)$$

Ce nombre est comparé à un seuil préfixé N_{\min} . Si $N_e \geq N_{\min}$ alors la sortie $C[i, j]$ est donnée par la formule (4-71). Sinon, elle est égale à la moyenne arithmétique calculée dans un voisinage immédiat du pixel central. Ceci a pour but d'éviter la trop forte sensibilité du filtre aux impulsions parasites. Le problème est reporté sur le choix de la valeur du seuil N_{\min} . Une valeur trop forte remet en cause l'effet de préservation des petites structures. Lee préconise une valeur limite de 3 pour une fenêtre 5×5 , et 4 pour une fenêtre 7×7 .

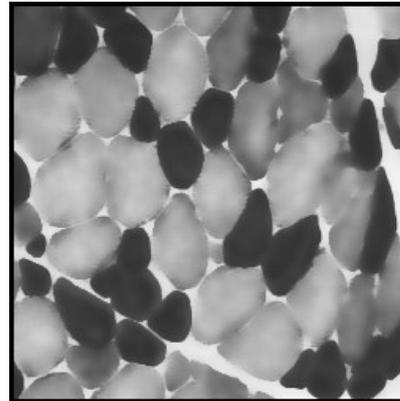
Le bon fonctionnement de ces opérateurs nécessite la connaissance de l'écart-type du bruit. Lorsque cette donnée n'est pas disponible, on peut choisir, pour le seuil t , la médiane des écarts par rapport à la médiane :

$$t = \text{median} \left\{ \left| a_1 - a_{\left(\frac{L+1}{2}\right)} \right|, \dots, \left| a_k - a_{\left(\frac{L+1}{2}\right)} \right|, \dots, \left| a_L - a_{\left(\frac{L+1}{2}\right)} \right| \right\} \quad (4-73)$$

Le filtre réalisé, appelé M-MAD-TM par ses auteurs [CAI86] présente un bon compromis entre performances et complexité.



seuil = 25



seuil = 50

Figure 4.14 : Filtrage par moyennes adaptatives (taille : 7×7)

2.4.2 Filtre de rang adaptatif

Le principe de ce filtre [ZAMP90] consiste à extraire de la fenêtre d'analyse la statistique de rang r , notée $a_{(r)}$, le rang étant calculé à partir des données présentes dans la fenêtre selon la formule (4-74) :

$$r = \text{Ent}[1 + \rho(L-1)] \quad (4-74)$$

$$\text{avec } \rho = \frac{1}{2} + \frac{\sum_{k=1}^L a_{(k)} - a_{(1)} - (a_{(L)} - a_{(1)}) \frac{k-1}{L-1}}{(L-1)(a_{(L)} - a_{(1)})}$$

L'étage de décision utilise l'information sur la symétrie (ou la dissymétrie) de la répartition des statistiques d'ordre $a_{(k)}$. Si la répartition est symétrique, la valeur de ρ est proche de $1/2$ et le filtre adaptatif a le comportement d'un filtre médian. Si cette répartition est décalée du côté des statistiques de rang faible, ce qui se produit en cas de présence d'une impulsion parasite positive, le filtre sélectionne une statistique de rang faible, ce qui élimine le parasite. Un comportement symétrique est obtenu en cas de parasite impulsionnel négatif.

Compte tenu de son principe, cet opérateur élimine les perturbations impulsionnelles et respecte les transitions entre régions (*cf.* figure 4.15). Dans les zones stationnaires, son comportement est celui d'un filtre médian si la distribution du bruit est symétrique. Un léger effet d'amplification du bruit, par sélection des statistiques de rang extrême, peut se produire quand la fenêtre d'analyse inclut une partie de la zone de transition. Cet effet améliore le contraste visuel, mais peut aussi affecter la mesure de l'intensité moyenne d'une région

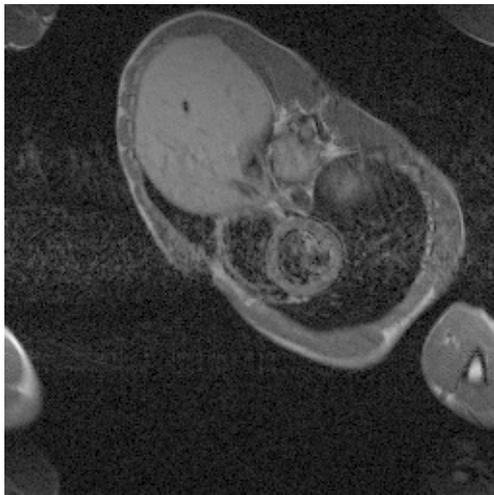


image initiale

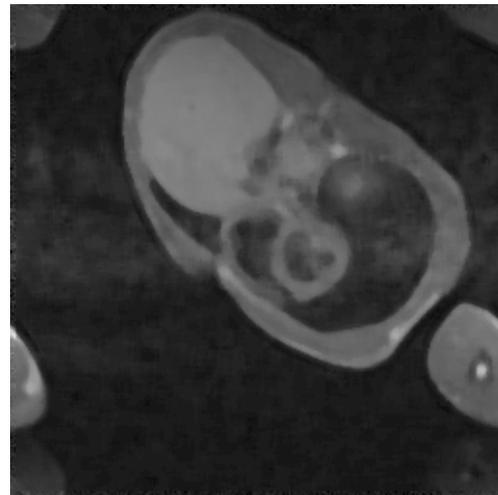


image filtrée

Figure 4.15 : Filtrage de rang adaptatif (taille : 7×7)

2.4.3 Moyenne tronquée adaptative

Ce filtre vise à tenir compte non seulement de la présence éventuelle de contours délimitant des régions, mais encore du caractère plus ou moins impulsionnel du bruit. L'étage de décision comporte deux parties. L'une a pour but de détecter la présence d'une transition à l'intérieur de la fenêtre d'analyse. Elle est basée sur le calcul de la quasi-étendue définie par :

$$Q_p = a_{(L-p)} - a_{(1+p)} \text{ qui est comparée à un seuil } t. \quad (4-75)$$

L'autre partie doit évaluer le caractère plus ou moins impulsionnel du bruit, que l'on atteint en calculant le rapport :

$$V_\alpha s = \frac{H_\alpha s - B_\alpha s}{H_{1/2} s - B_{1/2} s} \quad (4-76)$$

où $H_\alpha[s]$ et $B_\alpha[s]$ représentent les sorties des L-filtres dont les coefficients sont :

$$\text{pour H : } h_k = \begin{cases} 1/\alpha L & \text{pour } L+1 - \text{Ent}(\alpha L) \leq k \leq L \\ 1 - \frac{\text{Ent}(\alpha L)}{\alpha L} & \text{pour } k = L - \text{Ent}(\alpha L) \\ 0 & \text{ailleurs} \end{cases}$$

$$\text{pour B : } b_k = \begin{cases} 1/\alpha L & \text{pour } 1 \leq k \leq \text{Ent}(\alpha L) \\ 1 - \frac{\text{Ent}(\alpha L)}{\alpha L} & \text{pour } k = 1 + \text{Ent}(\alpha L) \\ 0 & \text{ailleurs} \end{cases}$$

La quantité V_α mesure l'écart entre la moyenne des αL plus grandes valeurs dans la fenêtre et la moyenne des αL plus petites valeurs. Si le bruit a une densité qui s'étale vers l'infini, V_α tend à prendre des valeurs élevées. Pour un bruit à distribution assez concentrée (de type uniforme par exemple), V_α prend des valeurs faibles.

L'étage de filtrage est constitué d'un ensemble de cinq L-filtres. On définit la moyenne externe ME comme la moyenne des αL valeurs extrémales :

$$\text{ME}_\alpha s = \frac{1}{2} H_{\alpha/2} s + B_{\alpha/2} s \quad (4-77)$$

La moyenne interne MI est définie comme la moyenne des αL valeurs médianes, c'est-à-dire comme la moyenne tronquée obtenue par un L-filtre dont les coefficients sont, pour L impair :

$$c_k = \begin{cases} 0 & \text{pour } 1 \leq k \leq k_{\min} \\ \frac{\alpha L - 1 - 2Ent\left(\frac{\alpha L - 1}{2}\right)}{2\alpha L} & \text{pour } k = k_{\min} \\ \frac{1}{\alpha L} & \text{pour } k_{\min} \leq k \leq k_{\max} \\ \frac{\alpha L - 1 - 2Ent\left(\frac{\alpha L - 1}{2}\right)}{2\alpha L} & \text{pour } k = k_{\max} \\ 0 & \text{pour } k_{\max} \leq k \leq L \end{cases}$$

avec $k_{\min} = \frac{L-1}{2} - 2Ent\left(\frac{\alpha L - 1}{2}\right)$ et $k_{\max} = \frac{L+3}{2} + 2Ent\left(\frac{\alpha L - 1}{2}\right)$

Le fonctionnement du filtre est présenté dans l'algorithme 4.1.

Dans [REST88], les auteurs préconisent le choix de $p = Ent(L/4)$, et $\alpha = 0.05$ ou 0.20 pour le calcul de la quantité V_α . Les seuils permettant de classifier la distribution du bruit selon son caractère plus ou moins impulsif sont également donnés pour des tailles de filtres allant de 3 à 25. Le tableau 4.2 indique les seuils pour un filtre 5×5 et $\alpha = 0.2$.

filtre	ME ₀	ME _{1/2}	MI ₁	MI _{1/2}	MI ₀
V _{0.2}].., 1.65]]1.65, 1.71]]1.71, 1.76]]1.76, 1.83]]1.83, ...[

Tableau 4.2 : Seuils pour le choix du filtre à appliquer (taille : 5×5).

```

SI transition détectée ALORS
    appliquer  $MI_0$           /*filtrage médian pour conserver les transitions*/
SINON
    SI bruit de type uniforme ALORS
        appliquer  $ME_0$       /*filtre milieu*/
    SINON
        SI bruit à distribution assez concentrée ALORS
            appliquer  $ME_{1/2}$ 
        SINON
            SI bruit de type gaussien ALORS
                appliquer  $MI_1$     /* filtre moyeneur */
            SINON
                SI bruit à distribution assez étalée ALORS
                    appliquer  $MI_{1/2}$ 
                SINON
                    appliquer  $MI_0$ 
                    /*filtre médian pour bruit impulsionnel*/

```

Algorithme 4.1 : Moyenne tronquée adaptative

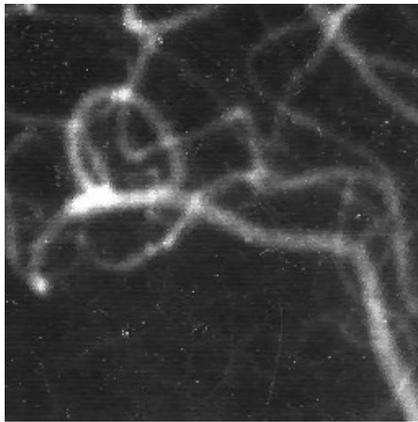


image initiale

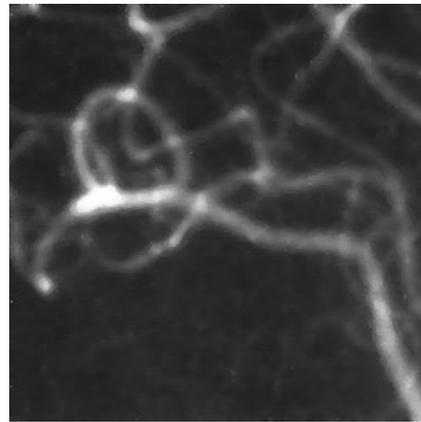


image filtrée (taille 5x5, seuil=30)

Figure 4.16 : Filtrage par moyennes tronquées adaptatives

La méthode ci-dessus peut sembler complexe à mettre en œuvre. Il faut remarquer que toutes les quantités nécessaires s'obtiennent par filtrage d'ordre des données. Une fois l'étape de tri effectuée, seuls les coefficients changent. Le principal problème provient du choix du seuil t qui dépend de la variance du bruit.

2.4.4 Filtrage par fractionnement de la fenêtre d'analyse

Pour ces opérateurs, la sélection des pixels intervenant effectivement dans le calcul de la sortie du filtre est faite en prenant en compte des contraintes géométriques.

L'opérateur de Nagao [NAGA79] travaille sur un voisinage 5x5 centré sur le pixel courant. Dans ce voisinage, 9 domaines sont définis (cf. figure 4.17). Pour chaque domaine D_i , on calcule la moyenne μ_i et la variance $\text{var}(D_i)$. L'intensité du pixel courant est remplacée par la moyenne du domaine qui présente la plus faible variance. Cet opérateur présente l'avantage de réduire le bruit tout en renforçant les contrastes par diminution de la largeur de la zone de transition (cf. figure 4.18). Le domaine central D_9 peut être ignoré [COCQ85] sans altérer notablement les performances. Pour diminuer le temps de calcul sans perte significative de qualité, on peut employer l'étendue des domaines au lieu de la variance.

Une version étendue au cas de filtres de tailles quelconques est proposée dans [SCHU94].

Lorsque les structures à rechercher dans l'image présentent une anisotropie marquée, l'adaptation de la fenêtre de filtrage doit être plus fine et plus directionnelle. Une solution consiste à mesurer localement cette anisotropie et à utiliser cette mesure pour faire varier la largeur de la fenêtre de filtrage [ZAMP92]. Les meilleures performances s'obtiennent en considérant non seulement les directions principales du maillage (horizontale, verticale et diagonales) mais aussi un ensemble de directions définies par les segments de droite discrets passant par le bord de la fenêtre d'analyse [VILA93].

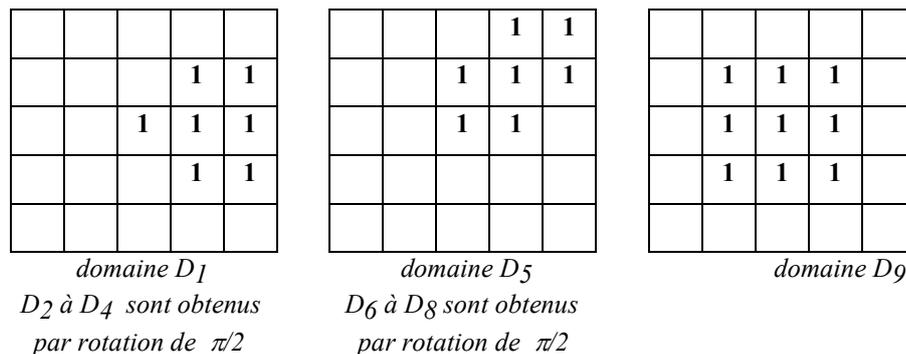


Figure 4.17 : Domaines de Nagao

2.4.5 Remarques

Les méthodes adaptatives doivent permettre de prendre en compte la non stationnarité de l'image en tant que signal. Elles nécessitent l'injection de connaissances *a priori* sur le type de structures à rechercher ou sur les perturbations à éliminer. D'autres méthodes sont disponibles, comme l'adaptation de la taille du filtre [LIN88] [SALE92]. Les filtres quadratiques [RAMP88]

[FONT93] donnent de bons résultats dans le cas d'images à très faible contraste. Cependant, ils ne sont pas invariants par changement d'intensité moyenne, ce qui rend leur utilisation délicate.

Bien que leurs performances en réduction de bruit soient attractives, les méthodes adaptatives issues du traitement du signal, de type LMS par exemple, ne sont pas couramment employées dans la phase de prétraitement. Cela tient peut-être au fait que le mécanisme d'adaptation introduit une dissymétrie spatiale dans le traitement, phénomène que l'on cherche en général à éviter.

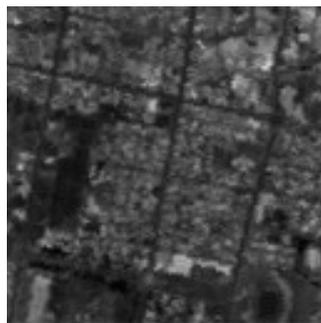
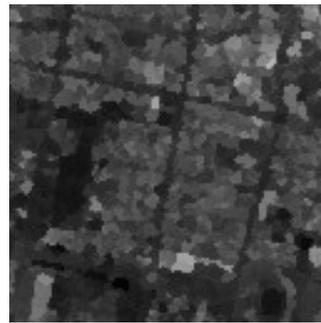
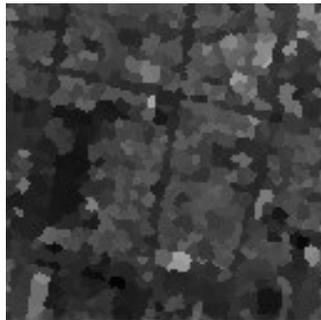


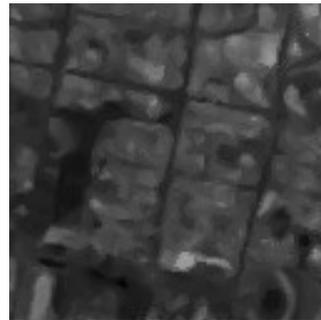
image initiale



filtrage de Nagao



filtrage de Nagao itéré 3 fois



filtrage directionnel (N=5) itéré 3 fois

Figure 4.18 : Filtrage à fractionnement de voisinage

2.5 Bruit multiplicatif

Un certain nombre de situations expérimentales peuvent être caractérisées par un bruit multiplicatif, plutôt que par un bruit additif. C'est, sous certaines conditions, le cas du bruit de speckle dans les images radar à ouverture synthétique, ou bien le bruit de grain dans l'analyse de films radiographiques. Le modèle de l'image est alors :

$$A_{i,j} = g_{i,j} \cdot B_{i,j} \quad (4-78)$$

g représente l'image sans défauts, B est un bruit à valeurs positives, indépendant de g , à variations rapides devant celles de g .

Pour retrouver g , il faut disposer de connaissances sur le bruit B (moments ou densité de probabilité). On peut toujours imposer au bruit d'avoir une moyenne

$E\{B\}$ égale à 1. L'image originale est alors estimée à un coefficient multiplicatif près, ce qui ne change en rien le problème de sa segmentation.

Une famille de filtres, appelés estimateurs linéaires généralisés [HILL90], est donnée par :

$$C[i, j] = \alpha(A[i, j])^p + \beta \quad (4-79)$$

où $p \in]0, 1]$ et α, β dépendent de p

Le filtre optimal est celui qui minimise l'erreur quadratique pour $p \in]0, 1]$.

Pour des valeurs fixées de p , on obtient des filtres sous-optimaux. Ainsi, en prenant $p = 1$, on obtient le filtre affine, appelé également « algorithme » de Lee. Pour $p = \text{var}(\ln g) / \text{var}(\ln A)$ et $\beta = 0$, on obtient une solution de type filtrage homomorphique.

2.5.1 Algorithme de Lee

Les coefficients optimaux sont :

$$\alpha = \frac{\text{var } g}{\text{var } A} = \left(\frac{E\{A^2\}}{E\{B^2\}} - E\{A^2\} \right) \frac{1}{\text{var } A} \quad (4-80)$$

$$\beta = E\{g\} - E\{A\} \frac{\text{var } g}{\text{var } A} = \frac{E\{A\}E\{A^2\} \text{var } B}{E\{B^2\} \text{var } A} \quad (4-81)$$

En pratique les moments d'ordre 2 de B sont supposés connus, et les moments d'ordre 1 et 2 de A sont estimés sur une fenêtre centrée sur le pixel $[i, j]$.

2.5.2 Filtrage homomorphique

En posant $\underline{a} = \ln A[i, j]$, $\underline{g} = \ln g[i, j]$ et $\underline{b} = \ln B[i, j]$, la relation (4-78) devient : $\underline{a} = \underline{g} + \underline{b}$, où \underline{b} est un bruit indépendant de \underline{g} . Le meilleur estimateur affine de \underline{g} est :

$$\underline{c} = c_1 + c_2 \cdot \underline{a}$$

$$\text{avec } c_1 = E\{\underline{g}\} - \frac{\text{var } \underline{g}}{\text{var } \underline{a}} E\{\underline{a}\} \text{ et } c_2 = \frac{\text{var } \underline{g}}{\text{var } \underline{a}} \quad (4-82)$$

La sortie du filtre est alors $C[i, j] = \exp(\underline{c})$

La structure de l'opérateur est donc de type homomorphique, les fonctions φ et Φ étant respectivement le logarithme népérien et l'exponentielle (cf. figure 4.19). La sortie de l'opérateur est bien de la forme : $\alpha \cdot A^p$ avec $\alpha = \exp(c_1)$ et $p = c_2$.

Le choix des valeurs des coefficients c_1 et c_2 requiert la connaissance des lois de g et B . Si seule la loi de B est connue (ce qui est souvent le cas pour le speckle d'une image radar), on peut approcher ces valeurs par une estimation locale de

paramètres dans une fenêtre centrée sur le pixel courant. Yan et Chen [YAN86] approchent la loi de g par une densité triangulaire de moyenne $E\{g\}$ et d'étendue $5(\text{var}(g))^{1/2}$. Compte tenu de la normalisation choisie pour B , on obtient :

$$E\{g\} = E\{A\} \quad \text{et} \quad E\{g^2\} = \frac{E\{A^2\}}{E\{B^2\}} \quad (4-83)$$

Les moments de $A[i, j]$ sont alors estimés localement [CHEH91].

Si la loi de B n'est pas connue, on peut effectuer un filtrage passe-bas de l'image $\ln(A)$, en tenant compte du fait que les variations de B , et donc de son logarithme, sont à échelle spatiale faible devant celle des variations de g .

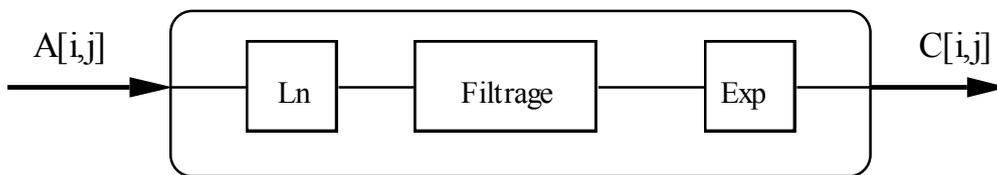


Figure 4.19 : Filtrage du bruit multiplicatif par opérateur homomorphique

2.6 Commentaires

Il existe une grande variété d'opérateurs, linéaires ou non, pour résoudre le problème de réduction du bruit d'image [PITA90]. Cependant, compte tenu de la grande quantité de données à traiter, les opérateurs simples et polyvalents seront privilégiés. Les filtres « à empilement », (« stack filters ») [WEND86] peuvent se décomposer en opérateurs traitant des signaux binaires par des fonctions logiques. Leur structure se prête assez bien à une réalisation VLSI. Il reste encore à choisir les bonnes fonctions logiques et à comparer les performances de tous ces filtres.

3 Rehaussement de contraste

Le but des opérateurs définis dans le paragraphe précédent est de réduire les fluctuations d'intensité à l'intérieur d'une même région, sans perturber, par élargissement, la zone de transition marquant la frontière entre régions. Il arrive cependant que la transition entre régions soit initialement floue, c'est-à-dire, qu'elle s'étale sur plusieurs pixels. Ceci se produit dans le cas de bougés (objets ou caméra mobiles) ou de défocalisation de la caméra. La cause peut également en être le processus physique de formation de l'image (phénomènes de diffraction en radiographie X ou échographie ultrasonore). Rehausser le contraste de l'image consiste à diminuer l'étendue de la zone de transition sans affecter l'intensité moyenne des régions situées de part et d'autre (cf. figure 4.20). On limite ainsi le risque de fusion intempestive de régions distinctes lors de la phase de segmentation.

L'opérateur de rehaussement de contraste devra, si possible, réduire le bruit dans les zones stationnaires et éviter les phénomènes de dépassement. Ce type d'opérateur est réalisable au moyen de méthodes linéaires ou non linéaires.

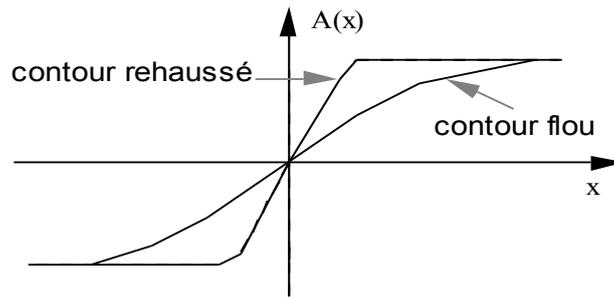


Figure 4.20: Rehaussement de contraste

3.1 Méthodes linéaires

3.1.1 Méthodes basées sur le laplacien

L'image rehaussée C s'obtient en soustrayant à l'image initiale A une certaine proportion de son laplacien ΔA .

$$C(x, y) = A(x, y) - \lambda \cdot \Delta A(x, y) \quad (4-84)$$

Pour présenter cette approche, on se place tout d'abord dans le cas d'un signal monodimensionnel continu décrivant un contour flou. Après les changements d'origine et d'échelle adéquats, le contour peut être modélisé par la fonction $A(x) = \arctan(x)$ (cf. figure 4.20).

La netteté de ce contour peut être caractérisée par la pente à l'origine en $x = 0$: $A'(0) = 1$. La méthode de rehaussement consiste à calculer :

$$C(x) = A(x) - \lambda A''(x) \quad \text{avec } \lambda > 0 \quad (4-85)$$

La pente en $x = 0$ devient :

$$C'(0) = 1 + 2\lambda > A'(0)$$

La zone de transition a été resserrée.

Il faut noter que, pour des images bruitées, cette méthode a tendance à amplifier le bruit. Soit $A(x)$ un modèle monodimensionnel de contour flou bruité.

$$A(x) = g(x) + B(x) \quad (4-86)$$

$$\text{avec : } \begin{cases} g(x) = \arctan(x) \\ B(x) : \text{bruit centré stationnaire indépendant de } g \end{cases}$$

Par filtrage, on obtient :

$$C(x) = g(x) - \lambda g''(x) + B(x) - \lambda B''(x) \quad (4-87)$$

La composante aléatoire $N(x) = B(x) - \lambda B''(x)$ est centrée, car $B(x)$ est centré. La variance de l'image rehaussée est :

$$\text{var}(C) = \text{var}(N) = \text{var}(B) + \lambda^2 \text{var}(B'') - 2\lambda \text{cov}(B, B'') \quad (4-88)$$

Or la covariance de B et B'' vaut :

$$\text{cov}(B, B'') = E\{B(x)B''(x)\} = R_{B''B}(0) = R''_{BB}(0) \quad (4-89)$$

Le terme de covariance, égal à la dérivée seconde de la fonction de corrélation de B, est négatif. Non seulement la dérivation tend à amplifier les hautes fréquences (bruit), mais cet effet est accentué par le terme de covariance. Ce phénomène est d'autant plus accentué que la valeur de λ est forte.

Une expression simple du laplacien bidimensionnel obtenue par approximation discrète des dérivées secondes est donnée au chapitre V (§1.5.1). Le calcul de la dérivée seconde est fait par un filtre linéaire de réponse impulsionnelle $+1, -2, +1$. Le masque du laplacien, somme des dérivées secondes horizontale et verticale, s'obtient donc par un filtrage linéaire dont les coefficients sont :

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

On utilise parfois le masque de convolution suivant :

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

ce qui revient à ajouter les dérivées secondes dans les directions diagonales.

Pour atténuer les effets d'amplification du bruit, on peut utiliser un opérateur laplacien filtré [SHEN85] obtenu en soustrayant à l'image initiale \mathbf{A} une version filtrée passe-bas $\Phi(\mathbf{A})$. L'image rehaussée est donnée par :

$$\mathbf{C} = (1 + \lambda)\mathbf{A} - \lambda\Phi(\mathbf{A}) \quad (4-90)$$

Le choix d'une valeur trop forte pour λ entraînera un phénomène de dépassement.

Pour introduire un effet de réduction de bruit, on estime le laplacien par une combinaison de deux filtrages passe-bas de l'image initiale. Soit Φ_1 et Φ_2 deux filtres linéaires dont les réponses fréquentielles \hat{h}_1 et \hat{h}_2 sont schématisées sur la figure 4.21. Le laplacien de l'image \mathbf{A} est estimé par $\Phi_2(\mathbf{A}) - \Phi_1(\mathbf{A})$. Finalement en utilisant la relation (4-85), l'image rehaussée est donnée par la relation :

$$\mathbf{C} = \mathbf{A} + \lambda\Phi_1(\mathbf{A}) - \lambda\Phi_2(\mathbf{A}) \quad (4-91)$$

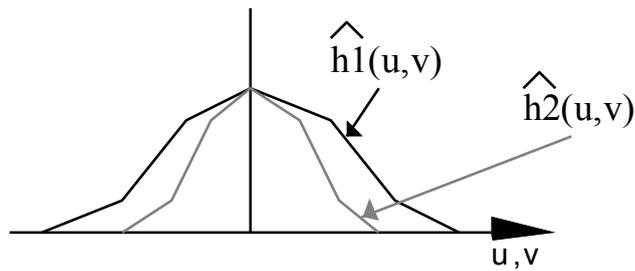


Figure 4.21 : Calcul du laplacien par différence de filtres passe-bas

Ce filtre est plus compliqué à régler puisqu'il y a trois paramètres à ajuster : les fréquences de coupure des filtres Φ_1 et Φ_2 et le coefficient λ . Comme précédemment, une valeur trop forte de λ entraîne, au voisinage des frontières, un phénomène de dépassement pouvant donner naissance à des régions parasites lors de la segmentation.

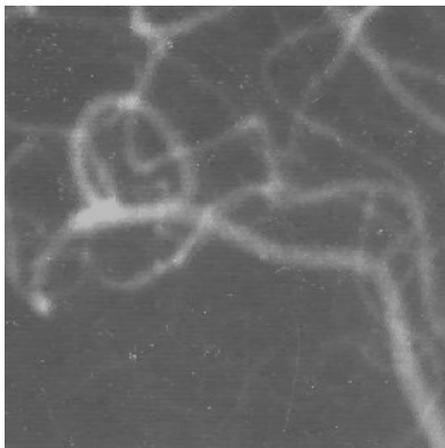


image initiale

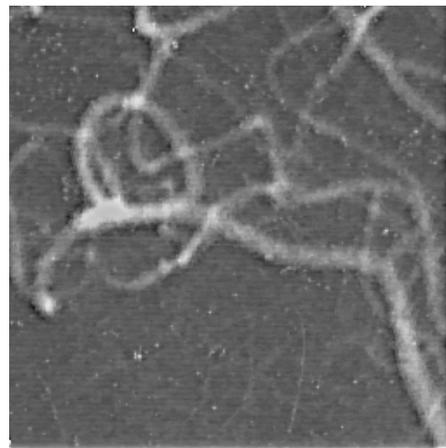


image rehaussée par laplacien obtenu par différence de deux filtres passe-bas

Figure 4.22 : Rehaussement de contraste par laplacien - filtrage exponentiel
 $\beta_1 = 1,33$ $\beta_2 = 0,57$ et $\lambda = 4$

La figure 4.22 donne un exemple de rehaussement de contraste sur une image d'angiographie. Le laplacien est calculé par différence de deux filtres exponentiels Φ_1 et Φ_2 dont les paramètres sont $\beta_1 = 1,33$ et $\beta_2 = 0,57$. Ces paramètres donnent aux filtres exponentiels l'effet de réduction de bruit de filtres moyenneurs de tailles 3×3 et 7×7 respectivement. On peut remarquer le phénomène de dépassement (« overflow ») qui donne naissance à une zone sombre sur les bords des vaisseaux sanguins. On note aussi que le bruit a été renforcé dans les régions correspondant au fond et à l'intérieur des vaisseaux.

3.1.2 Méthodes inverses

Si l'image floue peut être considérée comme le résultat du filtrage d'une image nette par un filtre linéaire Φ de type passe-bas, on peut envisager le rehaussement de contraste comme un problème de déconvolution. Pour une image bruitée, ceci se résout par filtrage de Wiener.

Supposons que l'on dispose de l'image \mathbf{A} , version floue et bruitée de l'image nette \mathbf{f} :

$$\mathbf{A} = \Phi(\mathbf{f}) + \mathbf{B} \quad (4-92)$$

Si h désigne la réponse impulsionnelle du filtre Φ , et \hat{h} sa transformée de Fourier, une version rehaussée de \mathbf{A} est $\mathbf{C} = g(\mathbf{A})$ où le filtre g a comme fonction de transfert :

$$\hat{g} = \frac{\hat{h} \cdot S_{ff}}{|\hat{h}|^2 S_{ff} + S_{BB}} \quad (4-93)$$

S_{ff} et S_{BB} étant les densités spectrales de l'image \mathbf{f} et du bruit \mathbf{B} .

L'usage de cette méthode requiert la connaissance du processus de dégradation Φ , et celle des caractéristiques spectrales de l'image nette.

3.2 Filtrage homomorphique

L'effet de flou peut être créé par des variations locales de l'éclairement qui, se traduisant de manière multiplicative dans l'image analysée, modifient l'aspect des frontières entre régions. L'image observée est donnée par :

$$A_{i,j} = g_{i,j} \cdot B_{i,j} \quad (4-94)$$

g représente l'image contenant des détails nets. \mathbf{B} est l'image de l'éclairement dont les fluctuations sont à échelle spatiale large vis à vis des détails de l'image g . \mathbf{B} agit donc comme un bruit multiplicatif. Contrairement au cas traité au paragraphe 2.5, il s'agit ici d'un processus basse fréquence alors que le signal utile est plutôt haute fréquence.

Un filtrage homomorphique (*cf.* figure 4.23) permet de séparer les contributions de g et de \mathbf{B} . Le filtre linéaire employé doit atténuer les basses fréquences et amplifier les hautes fréquences spatiales.

Le filtrage homomorphique est également envisageable quand l'image \mathbf{A} peut être considérée comme résultant du traitement de l'image g par un filtre linéaire de réponse impulsionnelle h :

$$\mathbf{A} = g * h \quad (4-95)$$

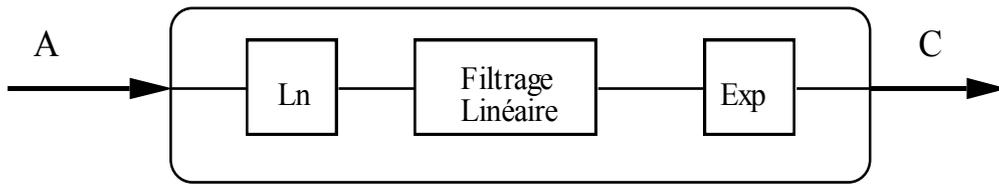


Figure 4.23 : Rehaussement de contraste par filtrage homomorphique

On est ramené à un problème de bruit convolutif (*cf.* § 2.3.2) dont on peut éliminer la contribution si l'information sur h est suffisante [JAIN89].

Pour des images comportant un bruit additif haute fréquence, le filtrage homomorphique est à utiliser avec précaution.

3.3 Filtrage d'ordre adaptatif

Dans cette approche, appelée filtrage par comparaison et sélection [LEE87], on recherche les zones convexes (CV) ou concaves (CC) du signal-image. Une représentation 1D est donnée en figure 4.24.

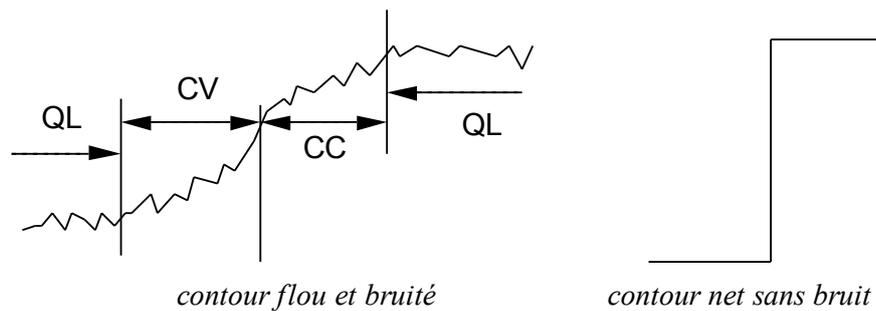


Figure 4.24 : Rehaussement de contraste : zones convexes CV, concaves CC, quasi-linéaires QL

Une zone qui n'est ni convexe ni concave est dite quasi-linéaire QL. Le principe du rehaussement de contraste est le suivant :

- dans une zone convexe, appliquer un filtre d'ordre Φ_1 accentuant les statistiques d'ordre faible (typiquement le minimum)
- dans une zone quasi-linéaire, appliquer un filtre de lissage Φ_2 (par exemple, le filtre d'ordre optimal)
- dans une zone concave, appliquer un filtre d'ordre Φ_3 accentuant les statistiques d'ordre élevé (typiquement le maximum)

L'opérateur comporte un étage de décision et un étage de filtrage. L'étage de décision est réalisé par un filtre d'ordre Φ dont les coefficients f_k vérifient les relations (4-96).

$$\sum_{k=1}^L f_k = 0 \text{ et } f_p = f_{L+1-p} \quad \forall p = 1..L \quad (4-96)$$

Dans une zone quasi-linéaire, la sortie du filtre Φ est nulle. Dans des zones de concavités opposées, les sorties sont de signe opposé.

En imposant aux filtres Φ_1 , Φ_2 et Φ_3 , d'avoir la somme de leurs coefficients respectifs égale à 1, on évite tout phénomène de dépassement et donc d'amplification de l'amplitude du bruit.

Les coefficients du filtre Φ peuvent être choisis de la manière suivante :

$$f_1 = f_L = \frac{1}{2} ; f_{(L+1)/2} = -1 ; f_k = 0 \text{ pour } k \neq 1, \frac{L+1}{2}, L \quad (4-97)$$

Ceci revient à comparer, dans la fenêtre d'analyse, la valeur milieu et la valeur médiane. Dans le cas d'images bruitées, on utilise un seuil pour la détection des concavités (ou convexités).

Si u désigne la sortie du filtre Φ , on a :

$u \geq \text{seuil} :$	zone convexe
$-\text{seuil} < u < +\text{seuil} :$	zone quasi-linéaire
$u \leq -\text{seuil} :$	zone concave

Avec de tels coefficients, on obtient de bons résultats lorsque le bruit d'image est de type gaussien ou uniforme (*i.e.* à distribution assez concentrée), tout en évitant les effets parasites rencontrés dans [LEE87] sur des transitions larges.

Si le bruit a un caractère plus impulsionnel, on peut utiliser la valeur quasi-milieu ; les coefficients sont alors :

$$f_{1+p} = f_{L-p} = \frac{1}{2} ; f_{(L+1)/2} = -1 ; f_k = 0 \text{ pour } k \neq 1+p, \frac{(L+1)}{2}, L-p \quad (4-98)$$

Avec un opérateur de ce type, il est possible d'effectuer un rehaussement parfait (avec 0 ou 1 pixel intermédiaire) en un seul passage, si la largeur de la transition est inférieure à la moitié de la taille du filtre.

Il est à noter qu'un effet similaire, mais moins accentué, est obtenu avec le filtre de rang adaptatif présenté au paragraphe 2.4.2. Par ailleurs, en utilisant un M-filtre de type α dont le paramètre α prend une valeur inférieure à 1, on obtient un effet conjoint de réduction de bruit et de rehaussement de contraste.

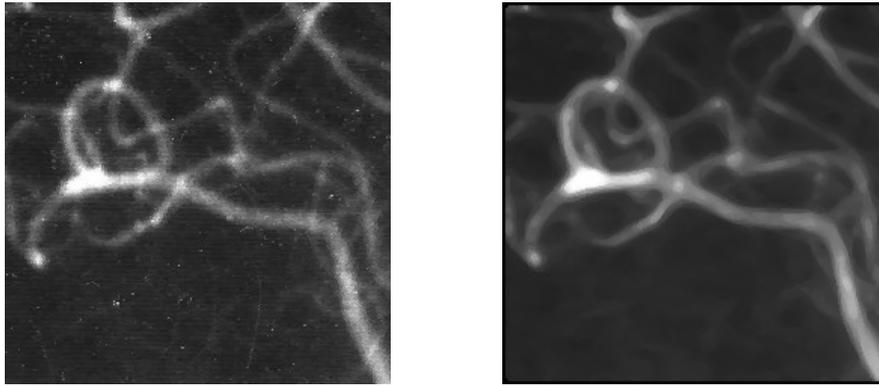


image initiale

image rehaussée (taille 7x7)

Figure 4.25 : Réduction de bruit et rehaussement de contraste par filtrage d'ordre adaptatif

3.4 Méthodes morphologiques

Les transformations morphologiques de rehaussement de contraste exploitent l'information de proximité relative de la fonction initiale avec sa dilatée et son érodée. L'effet recherché est d'affecter à un pixel la valeur, soit de l'image dilatée, soit de l'image érodée, pour créer des discontinuités. Le fonctionnement est donc semblable à celui d'un filtre d'ordre adaptatif mais il ne fait intervenir que des opérations morphologiques.

Pour simplifier, on se place dans l'ensemble des fonctions continues Φ . Le *rehaussement de contraste simple* de Kramer et Bruckner [KRAM75] [SERR82] est défini ci-après par la fonction $\psi_g(f)$.

$\forall f \in \Phi$, pour tout élément structurant g , on a : $f \ominus \overset{\vee}{g} \leq f \leq f \oplus \overset{\vee}{g}$. On pose, pour tout pixel s : $M[s] = 1/2 \left((f \ominus \overset{\vee}{g})[s] + (f \oplus \overset{\vee}{g})[s] \right)$

La fonction $\psi_g(f)$ est définie par :

<p>SI $f[s] \leq M[s]$ ALORS</p> $\psi_g(f)[s] = (f \ominus \overset{\vee}{g})[s]$ <p>SINON</p> $\psi_g(f)[s] = (f \oplus \overset{\vee}{g})[s]$
--

De meilleurs résultats sont obtenus en combinant les sorties d'opérateurs de contraste associés à des éléments structurants de différentes tailles. On définit ainsi un opérateur de *contraste naturel* Ψ_N par :

$$\Psi_N(f) = \frac{1}{N} \sum_{i=1}^N \Psi_i(f) \quad (4-99)$$

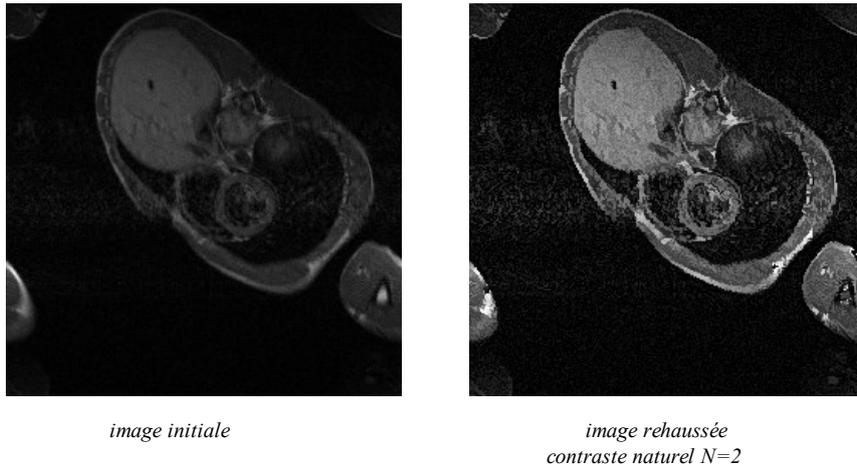


Figure 4.26 : Résultat de l'opérateur de contraste naturel

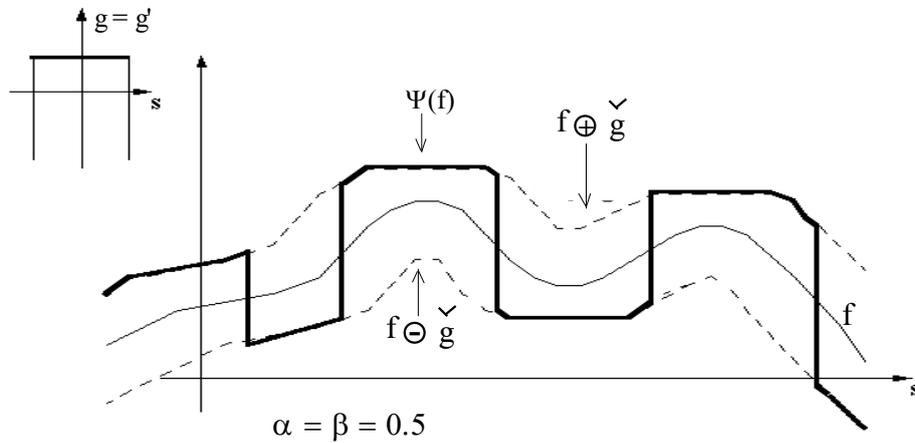


Figure 4.27 : Exemple de rehaussement de contraste par transformations morphologiques

D'une façon générale, pour exploiter au mieux l'information *a priori* des images à rehausser, on introduit le *contraste simple généralisé* [PRET87] défini pour deux éléments structurants g et g' dans \mathcal{D} telles que 0 appartienne aux supports des deux fonctions et deux scalaires α et β dans $[0,0.5]$. L'expression générale est donnée par :

$$\psi(f)[s] = \begin{cases} (f \oplus \check{g}')[s] & \text{si } (f \oplus \check{g}')[s] - f[s] \leq \alpha M[s] \\ (f \ominus \check{g})[s] & \text{si } f[s] - (f \ominus \check{g})[s] \leq \beta M[s] \\ f[s] & \text{sinon} \end{cases}$$

avec $M[s] = (f \oplus \check{g})[s] - (f \ominus \check{g}')[s]$

Il est à noter que des opérateurs similaires de rehaussement de contraste peuvent être définis en considérant l'ouvert et le fermé de l'image initiale à la

place de l'érodé et du dilaté. Les effets alors obtenus sont plus doux et la robustesse au bruit augmentée.

3.5 Observations

Le choix d'une méthode de rehaussement de contraste dépend des informations *a priori* disponibles. Les méthodes de déconvolution nécessitent une connaissance assez fine des processus de dégradation. Les méthodes de type laplacien ont l'avantage d'être simples. Elles sont néanmoins sensibles au bruit d'image. De plus, des dépassements peuvent se produire si le phénomène de dégradation n'est pas stationnaire sur toute l'image.

Les autres méthodes présentées ici sont plus complexes à mettre en œuvre, par leur non linéarité ou par la difficulté du choix d'un jeu de paramètres adaptés. Elles permettent toutefois d'obtenir des résultats sensiblement meilleurs.

4 Conclusion

Dans le processus d'analyse d'une image, la phase de prétraitement a pour but d'améliorer l'homogénéité des régions qui la composent, tout en conservant, voire en renforçant, le contraste entre régions adjacentes. Nous avons considéré ici que la segmentation était basée sur la valeur de l'intensité moyenne des régions, le cas des images texturées étant traité par ailleurs. Naturellement, les techniques présentées peuvent être adaptées pour exploiter tout attribut numérique autre que le niveau de gris. Les filtres de rang et les filtres morphologiques sont même utilisables dès que les valeurs des attributs peuvent être ordonnées.

Les méthodes de filtrage linéaire s'étendent au cas d'images de nature vectorielle (images couleur ou plus généralement multispectrales). Il en va de même pour les traitements par M-filtres. Il faut aussi noter que d'autres approches introduites dans cet ouvrage, telles que la restauration markovienne ou les méthodes multirésolution, peuvent être utilisées en phase de prétraitement. Leur complexité en réserve l'usage à des images difficiles à analyser.

Le nombre d'opérateurs, linéaires ou non linéaires, disponibles pour le prétraitement est très important. Le choix d'une méthode, ou de l'enchaînement de traitements élémentaires, dépendra des informations dont on dispose *a priori* sur le contenu de l'image. Dans le contexte d'applications nécessitant des cadences de traitement élevées, il faut aussi tenir compte de l'existence, ou de la possibilité de réalisation, d'opérateurs câblés ou intégrés.

CHAPITRE V

Approche frontière: méthodes dérivatives, surfaiques, morphologiques

Dans une image, les variations d'intensité représentent des changements de propriétés physiques ou géométriques de la scène ou de l'objet observé correspondant par exemple à :

- des variations d'illumination, des ombres,
- des changements d'orientation ou de distance à l'observateur,
- des changements de réflectance de surface,
- des variations d'absorption des rayons (lumineux, X, *etc.*).

Dans un grand nombre de cas, ces variations d'intensité sont des informations importantes pour les opérations situées en aval de la segmentation. Elles constituent les frontières de régions correspondant à des bords ou parties d'objets de la scène. D'où le nom donné à cette approche de la segmentation.

Intuitivement, dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes ; il s'agit de contours de type « saut d'amplitude ». Un contour peut également correspondre à une variation locale d'intensité présentant un maximum ou un minimum ; il s'agit alors de contour « en toit ». Les descriptions qui précèdent ne couvrent pas tous les cas et ne s'appliquent pas en particulier aux frontières séparant des régions de textures différentes. Cet aspect sera abordé dans les chapitres consacrés à l'approche région de la segmentation (chapitres X, XI et XII). Dans le présent chapitre, nous ne considérons que les méthodes détectant les variations locales d'intensité.

Une manière d'aborder ce problème est de considérer que l'image numérique résulte de l'échantillonnage d'une fonction scalaire $A(x, y)$ à support borné et dérivable en tout point (*cf.* annexe D). Les développements théoriques sont alors présentés dans le domaine continu, en particulier dans les approches *dérivatives* et *surfaiques*.

Les *approches dérivatives* sont les plus immédiates pour détecter et localiser les variations du signal. Les contours sont assimilés aux points de fort gradient ou de dérivée seconde nulle.

Dans les approches *surfacique* et *morphologique*, l'image des intensités est considérée comme une surface.

Dans les méthodes surfaciques, la transition entre deux régions est modélisée par un gabarit utilisable à deux fins :

- un contour est présent quand la mise en correspondance entre le gabarit et une zone de l'image est bonne.
- l'approximation par facettes de la surface fournit une équation analytique locale qui permet de calculer de manière très précise (sub-pixel) la position du contour et ses caractéristiques.

Les méthodes morphologiques travaillent sur les maximums et les minimums des intensités du voisinage de chaque pixel. Un contour de type « saut d'amplitude » sera détecté si la différence entre le maximum et le minimum est importante ou, pour des contours « en toit » par extraction des lignes de crêtes entre bassins versants.

Toutes ces méthodes sont sensibles au bruit. C'est la raison pour laquelle, comme nous l'avons indiqué dans le chapitre précédent, l'image est souvent *prétraitée* avant la segmentation proprement dite.

Une comparaison des performances théoriques des opérateurs de type dérivatifs est faite pour des signaux continus et discrets dans le paragraphe 1.8. Une comparaison plus complète est présentée au chapitre IX sur des images de synthèse et des images de la banque du GDR 134³. Cependant, nous fournissons, dans ce chapitre, quelques résultats d'extraction de contours sur des images de cette banque pour illustrer et exposer les caractères généraux des opérateurs étudiés.

L'implantation de certains opérateurs, présentés dans ce chapitre, est étudiée en détail dans l'annexe E.

1 Méthodes dérivatives

1.1 Généralités

Le principe général des méthodes dérivatives est illustré sur un signal continu monodimensionnel $A(x)$ présentant une transition avec un saut d'amplitude en x_0 . Les allures de la dérivée première et de la dérivée seconde sont données par la figure 5.1 (où $x_0 = 0$). Si on considère que la transition du signal est repérée par son point d'inflexion, sa localisation peut se faire par recherche du maximum local de la valeur absolue de la dérivée première ou par recherche du passage par zéro de la dérivée seconde. On peut définir la zone de transition du signal comme un intervalle comprenant le maximum (ou le minimum) local de la dérivée première ou le passage par zéro de la dérivée seconde. L'identification d'une zone de

³GDR 134 : Groupe de recherche « Traitement du signal et images » du CNRS.

transition du signal peut être faite par seuillage de la norme de sa dérivée première. Si le seuil est trop bas, on détecte aussi des transitions dues au bruit.

Dans le cas bidimensionnel, l'image est représentée par une fonction scalaire continue $A(x, y)$. Le vecteur gradient est défini au point M de coordonnées (x, y) par :

$$\mathbf{n} = \begin{pmatrix} \frac{\partial A}{\partial x} & \frac{\partial A}{\partial y} \end{pmatrix} \quad (5-1)$$

Le vecteur gradient en un point M est normal à la courbe de niveau donnée par $A(x, y) = \text{Constante}$ qui passe par M . La plus grande variation de $A(x, y)$ a lieu quand on se déplace le long de la normale à la courbe de niveau. Le maximum de cette variation est défini en grandeur et en direction par $\nabla A(x, y)$. Localement, la norme du gradient est d'autant plus élevée que les courbes de niveau sont rapprochées et de valeurs « sensiblement » différentes.

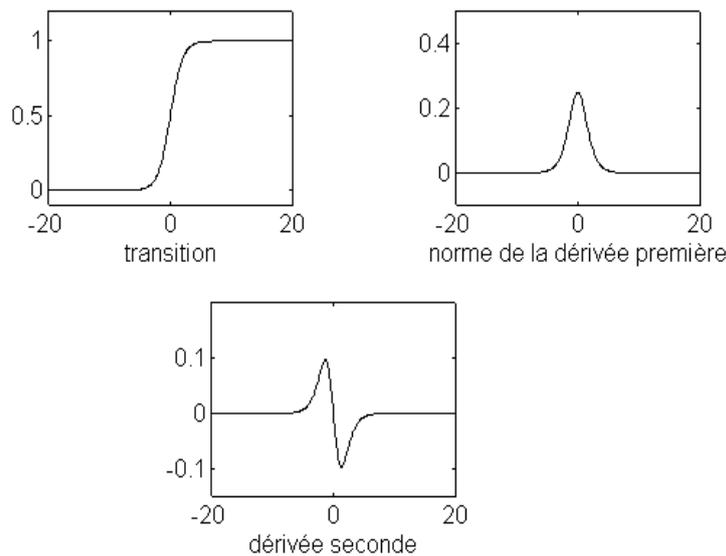


Figure 5.1 : Allure des dérivées première et seconde d'une transition de type « saut d'amplitude ».

Au point $M(x, y)$, l'orientation ϕ du gradient est donnée par :

$$\phi = \arctan\left(\frac{\frac{\partial A}{\partial y}}{\frac{\partial A}{\partial x}}\right) \quad (5-2)$$

et le vecteur unitaire \mathbf{n} normal au contour a pour expression :

$$\mathbf{n} = \begin{pmatrix} \cos \phi & \sin \phi \end{pmatrix} \quad (5-3)$$

L'orientation du contour vaut alors : $\theta = \frac{\pi}{2} - \phi$ (cf. figure 5.2)

La recherche des maximums locaux et des passages par zéro nécessite le calcul des dérivées directionnelles. La dérivée première directionnelle de $A(x, y)$ suivant \mathbf{n} est donnée par :

$$\frac{\partial A}{\partial \mathbf{n}} = \nabla A(x, y) \cdot \mathbf{n} = \frac{\partial A}{\partial x} \cos \phi + \frac{\partial A}{\partial y} \sin \phi \quad (5-4)$$

A un extremum de la dérivée directionnelle suivant \mathbf{n} correspond un passage par zéro de la dérivée seconde directionnelle donnée par :

$$\frac{\partial^2 A}{\partial \mathbf{n}^2} = \frac{\partial^2 A}{\partial x^2} \cdot \cos^2 \phi + 2 \frac{\partial^2 A}{\partial x \partial y} \cos \phi \sin \phi + \frac{\partial^2 A}{\partial y^2} \cdot \sin^2 \phi \quad (5-5)$$

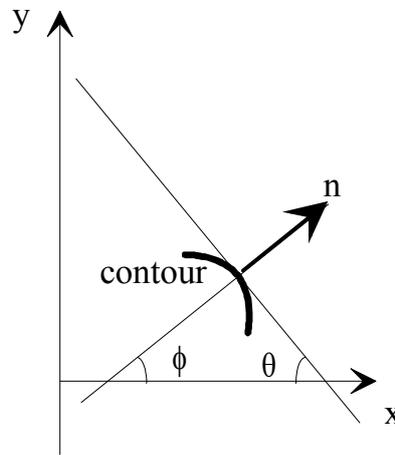


Figure 5.2 : Orientation d'un contour

Pour alléger les calculs, on utilise souvent les passages par zéro du laplacien pour localiser les transitions. Du fait de son isotropie l'opérateur laplacien peut s'exprimer en fonction des dérivées secondes selon deux directions orthogonales quelconques. Pour les directions définies par les axes Ox et Oy , il s'écrit alors :

$$\Delta A(x, y) = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} \quad (5-6)$$

Selon les directions \mathbf{n} et \mathbf{n}_\perp orthogonales, il vaut:

$$\Delta A(x, y) = \frac{\partial^2 A}{\partial \mathbf{n}^2} + \frac{\partial^2 A}{\partial \mathbf{n}_\perp^2} \quad (5-7)$$

Le second terme de l'expression ci-dessus est lié à la courbure du contour et a pour expression :

$$\frac{\partial^2 A}{\partial \mathbf{n}_\perp^2} = \frac{\partial^2 A}{\partial x^2} \cdot \sin^2 \phi - 2 \frac{\partial^2 A}{\partial x \partial y} \cos \phi \sin \phi + \frac{\partial^2 A}{\partial y^2} \cdot \cos^2 \phi \quad (5-8)$$

Il est important de signaler, qu'en général, les passages par zéro de la dérivée seconde directionnelle selon \mathbf{n} ne coïncident pas avec les passages par zéro du laplacien. Il y a coïncidence si les variations d'intensité sont linéaires sur la ligne du passage par zéro et sur des lignes de niveau « parallèles », ceci dans un voisinage réduit. Nous reproduisons ci-après les éléments de la démonstration exposée dans [MARR80].

Supposons que $A(x, y)$ soit deux fois dérivable. Soient : Oxy un repère situé dans l'image, l un segment ouvert porté par l'axe Oy avec $\frac{\partial^2 A}{\partial x^2} = 0$ sur l et $V(l)$ un voisinage ouvert réduit de ce segment (cf. figure 5.3).

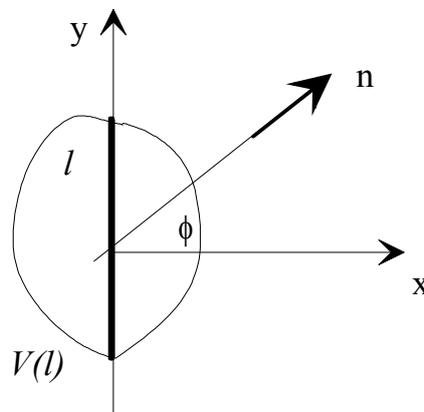


Figure 5.3 : Coïncidence des passages par zéro

Si $A(0, y)$ est constant ou linéaire, la dérivée $\frac{\partial A(0, y)}{\partial y}$ est constante le long du segment l et par conséquent :

$$\frac{\partial^2 A}{\partial x \partial y} = \frac{\partial^2 A}{\partial y^2} = 0$$

La dérivée seconde directionnelle suivant \mathbf{n} s'écrit alors :

$$\frac{\partial^2 A}{\partial n^2} = \frac{\partial^2 A}{\partial x^2} \cdot \cos^2 \phi \quad (5-9)$$

Elle vaut zéro si le segment l correspond à un passage par zéro du laplacien. Par ailleurs, la relation précédente montre que la pente de la dérivée seconde directionnelle est maximale pour $\phi = 0$, donc pour une orientation perpendiculaire à la ligne de passage par zéro.

1.2 Principes généraux d'utilisation des méthodes dérivatives

La présentation du paragraphe précédent permet de définir trois techniques générales utilisant l'approche dérivative. Elles comportent toutes un seuillage en

dernière étape pour éliminer les transitions de faible gradient pouvant correspondre à du bruit.

a) Utilisation du gradient

- Calcul du gradient en chaque point de l'image.
- Création de l'image de la norme du gradient.
- Pour chaque point de l'image de la norme, extraction des maximums locaux (obtention de contours fins), dans la direction exacte du gradient.
- Seuillage par hystérésis de l'image des maximums locaux : on seuille d'abord avec un seuil haut et on conserve ensuite toutes les chaînes ayant au moins un point dont la norme du gradient est supérieure au seuil bas (*cf.* § 4.2).

b) Utilisation de la dérivée seconde

- Calcul de la dérivée seconde dans la direction du gradient.
- Recherche des passages par zéro de la dérivée seconde dans la direction du gradient.
- Création de l'image des passages par zéro affectés de la norme du gradient.
- Seuillage par hystérésis (élimination des passages par zéro non significatifs).

c) Utilisation du laplacien

- Calcul du laplacien.
- Recherche des passages par zéro du laplacien.
- Création de l'image des passages par zéro affectés de la norme du gradient.
- Seuillage par hystérésis (élimination des passages par zéro non significatifs) comme précédemment.

Les techniques b) et c) nécessitent le calcul du gradient.

Par la suite, différentes méthodes d'approximation du gradient et de la dérivée seconde sont présentées.

1.3 Approches dérivatives appliquées aux images numériques

La dérivation d'un signal échantillonné peut être considérée comme équivalente à l'application d'un filtre dérivée de gaussienne au signal original, suivie d'un échantillonnage (*cf.* annexe D).

Poggio [POGG85] aboutit à la même conclusion par une voie différente. L'image numérique $A[i, j]$ est formée par discrétisation d'une fonction réelle $A(x, y)$:

$$A[i, j] = Q.A(x, y)$$

où Q est un opérateur de discrétisation et de codage.

La différentiation numérique est un problème mal posé, car la solution ne dépend pas de façon continue des données. Le problème est donc de trouver une

représentation deux fois différentiable de $A(x, y)$. Ceci peut être fait en utilisant des techniques de régularisation. En pratique, le noyau gaussien permet d'obtenir par convolution, des estimations régularisées de $A(x, y)$ ainsi que de ses dérivées première et seconde. L'estimation régularisée de l'image est alors donnée par :

$$\tilde{A}(x, y) = g * A[x, y] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} A[u, v] g(x-u, y-v) du dv \quad (5-10)$$

où $g(x, y)$ est une fonction gaussienne centrée de deux variables de covariance σ :

$$g(x, y) = \frac{1}{G_0} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (5-11)$$

1.4 Opérateurs dérivatifs du premier ordre.

Dans le cas discret, les dérivées directionnelles suivant les directions horizontale et verticale au site $[i, j]$ sont approchées par de simples différences finies :

$$\frac{\partial A}{\partial y} \approx \frac{\Delta A}{\Delta i} = A_{i, j} = A_{i+1, j} - A_{i, j} \quad (5-12)$$

$$\text{et : } \frac{\partial A}{\partial x} \approx \frac{\Delta A}{\Delta j} = A_{i, j} = A_{i, j+1} - A_{i, j} \quad (5-13)$$

La norme du gradient est alors donnée par :

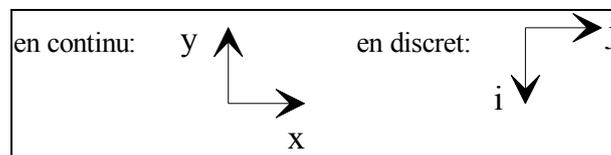
$$|\nabla A_{i, j}| = \sqrt{A_{j, i}^2 + A_{i, j}^2} \quad (5-14)$$

où encore par :

$$|\nabla A[i, j]| = \max \{A_{j, [i, j]}, A_{[i, j]}\} \quad (5-15)$$

Notons que ces opérateurs sont particulièrement sensibles au bruit.

+Rappel des conventions d'orientation des axes :



1.4.1 Opérateurs de Prewitt et de Sobel

Pour ces opérateurs, les dérivées directionnelles horizontale et verticale s'expriment sous la forme :

$$A_{j, i, j} = h_j * A_{i, j} \quad \text{et} \quad A_{i, i, j} = h_i * A_{i, j} \quad (5-16)$$

$$\text{avec : } \mathbf{h}_j = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \text{ et } \mathbf{h}_i = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix} \quad (5-17)$$

Les matrices \mathbf{h}_j et \mathbf{h}_i , appelées aussi masques, sont les noyaux de convolution de filtres à réponse impulsionnelle finie.

Les masques de Prewitt [PREW70] sont définis par $c = 1$ et les masques de Sobel par $c = 2$.

Rappelons que la relation entre convolution et corrélation peut être traduite par l'expression :

$$\begin{aligned} \mathbf{h} * \mathbf{A}_{i,j} &= \sum_{m=-M}^M \sum_{n=-N}^N \mathbf{h}_{m,n} \cdot \mathbf{A}_{i-m,j-n} \\ &= \sum_{m=+M}^{-M} \sum_{n=+N}^{-N} \mathbf{h}_{-m,-n} \cdot \mathbf{A}_{i+m,j+n} \end{aligned} \quad (5-18)$$

où \mathbf{h} est un noyau de convolution de taille $(2M+1) \times (2N+1)$.

Les calculs précédents des deux dérivées directionnelles peuvent donc être considérés comme des corrélations avec les « gabarits » $\mathbf{h}_j[-j]$ et $\mathbf{h}_i[-i]$.

Il est intéressant de remarquer que les 2 masques de Prewitt et Sobel correspondent à la composition de deux convolutions. Par exemple, \mathbf{h}_j représente la réponse impulsionnelle d'un filtre séparable comprenant un lissage suivant la direction verticale à l'aide du noyau $\mathbf{h}_{Lj} = [1, c, 1]$ et une dérivation suivant la direction horizontale à l'aide du noyau $\mathbf{h}_{Dj} = [1, 0, -1]$, soit :

$$\mathbf{h}_j = \mathbf{h}_{Lj} \cdot \mathbf{h}_{Dj} \quad (5-19)$$

$$\text{de même : } \mathbf{h}_i = \mathbf{h}_{Di} \cdot \mathbf{h}_{Lj} \quad (5-20)$$

Il s'en suit que :

$$\mathbf{h}_j * \mathbf{A}_{i,j} = \sum_{m=-M}^M \mathbf{h}_{Lj} \cdot \mathbf{h}_{Dj} \cdot \mathbf{A}_{i-m,j-n} \quad (5-21)$$

Cette idée sera reprise pour l'implémentation de détecteurs de contours plus complexes.

Soient $\mathbf{A}_j = \mathbf{A} * \mathbf{h}_j$ et $\mathbf{A}_i = \mathbf{A} * \mathbf{h}_i$ les images obtenues par convolution de l'image originale \mathbf{A} avec les deux masques \mathbf{h}_j et \mathbf{h}_i . Il est possible comme précédemment de calculer la norme et l'orientation du gradient :

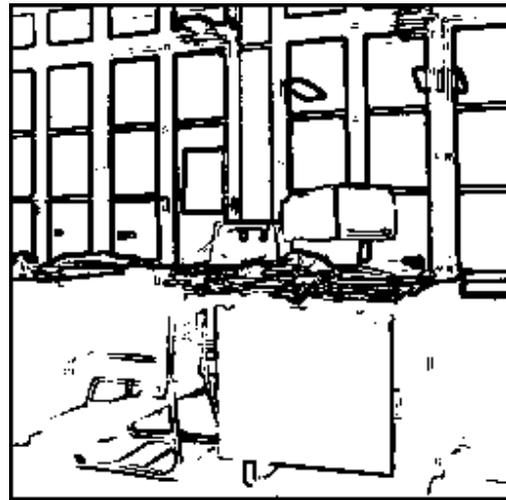
$$|\nabla \mathbf{A}_{i,j}| = \sqrt{\mathbf{A}_j^2(i,j) + \mathbf{A}_i^2(i,j)} \text{ et } \theta = \frac{\pi}{2} - \arctan(\mathbf{A}_i / \mathbf{A}_j) \quad (5-22)$$

Une approche discrète pour la détermination de l'orientation du gradient consiste à utiliser des filtres directionnels adaptés à certaines orientations prédéfinies. Par exemple, le masque suivant donne une réponse maximum pour un contour d'orientation $\pi/4$ (cf. §1.4.2.) :

$$h_1 = \begin{bmatrix} c & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -c \end{bmatrix} \quad (5-23)$$



Image originale



Sobel (seuil 150)

Figure 5.4 : Contours obtenus avec l'opérateur de Sobel

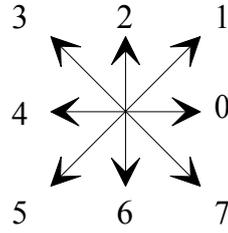
Une extraction de contours faite par l'opérateur de Sobel sur l'image BUREAU est présentée figure 5.4. On observe que les contours obtenus sont épais. Le seuil est réglé empiriquement. Cet opérateur est facile à implanter. Il a été utilisé dans les premières réalisations temps réel de détection de contours dans les années 1980. Les processeurs actuels de traitements du signal autorisent l'implantation d'opérateurs plus performants.

1.4.2 Opérateurs de gradients directionnels de Kirsh

L'opérateur de Kirsh [KIRS71] est un opérateur à huit masques correspondant chacun à une direction préférentielle et obtenu par rotation de $\pi/4$ de l'opérateur de base \mathbf{h}_0 (la numérotation des masques est faite dans l'ordre des directions de Freeman en 8-connexité) :

$$h_0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad h_1 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad (5-24)$$

avec le codage de Freeman suivant :



Le gradient retenu est donné par :

$$\max_i \{ \mathbf{h}_i * \mathbf{A}; i = 0, 7 \} \quad (5-25)$$

et l'orientation sera donnée par : $\frac{\pi}{4} \cdot \operatorname{argmax}_i \{ \mathbf{h}_i * \mathbf{A}; i = 0, \dots, 7 \}$.

1.4.3 Opérateurs MDIF et NAGDIF.

Ces deux opérateurs sont la combinaison d'un lissage et d'une dérivation. Le premier est linéaire et le second non-linéaire.

1.4.3.1 Opérateur MDIF

Il s'agit de la combinaison d'un filtre moyenneur de noyau \mathbf{m} et d'un dérivateur utilisant les masques directionnels de Prewitt \mathbf{h}_0 et \mathbf{h}_2 :

$$\mathbf{m} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{h}_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \mathbf{h}_2 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (5-26)$$

Les approximations des composantes du gradient peuvent être calculées après le lissage ou directement en appliquant les masques suivants de taille 5x5 :

$$\mathbf{m}_j = \mathbf{m} * \mathbf{h}_2 = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix} \quad \mathbf{m}_i = \mathbf{m} * \mathbf{h}_1 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (5-27)$$

Soient $\mathbf{A}_j = \mathbf{A} * \mathbf{m}_j$ et $\mathbf{A}_i = \mathbf{A} * \mathbf{m}_i$ les images obtenues par convolution de l'image originale \mathbf{A} avec les deux masques \mathbf{m}_j et \mathbf{m}_i . Il est possible comme précédemment de calculer la norme et l'orientation du gradient. On extrait ensuite les maximums locaux de la norme dans la direction du gradient (*cf.* figure 5.5).

1.4.3.2 Opérateur NAGDIF

L'opérateur NAGDIF présenté dans [COCQ85] est la combinaison d'un lissage non-linéaire de Nagao [NAGA79] (*cf.* chapitre IV) suivi d'une dérivation effectuée par l'opérateur suivant :

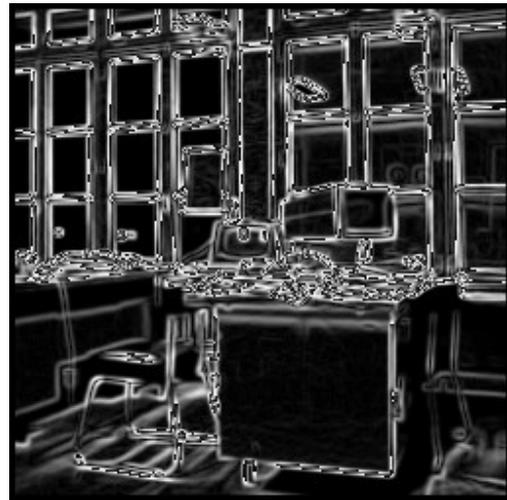
$$\text{CONTOUR SI } \max\{B[i, j] - B[i - \varepsilon_1, j - \varepsilon_2]\} > \text{seuil}$$

où ε_1 et ε_2 sont des valeurs entières de l'intervalle $[-1, 1]$ et \mathbf{B} est l'image lissée.

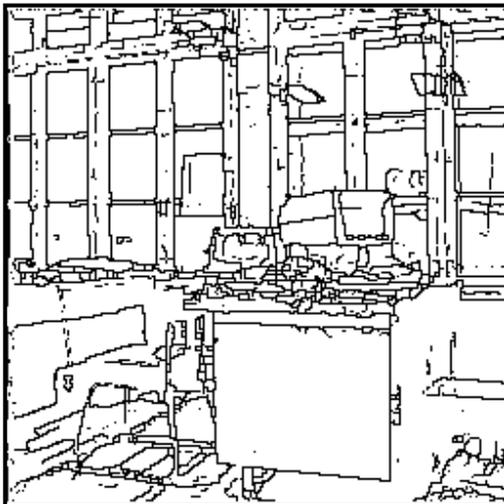
Cet opérateur étend une technique fondée sur un lissage non-linéaire avec des domaines carrés [TOMI77].



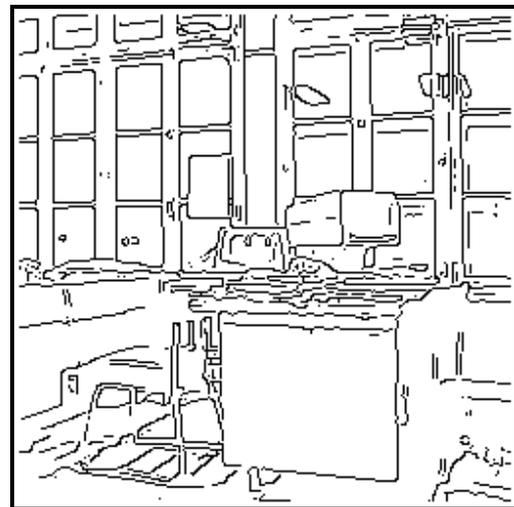
Image originale



Normes des gradients calculées par MDIF



NAGDIF



MDIF

Figure 5.5 : Contours obtenus avec les opérateurs NAGDIF et MDIF

Les contours obtenus par application des opérateurs NAGDIF et MDIF sur l'image BUREAU sont présentés figure 5.5. Les seuils sont réglés empiriquement.

L'opérateur NAGDIF donne des contours fins mais moins réguliers que l'opérateur MDIF. Cependant, il faut signaler que les contours fins obtenus avec MDIF résultent d'une extraction des maximums locaux (*cf.* § 4) dans l'image de la norme du gradient. Les deux opérateurs sont intéressants pour des images peu bruitées comportant des transitions de « largeur » inférieure à 3 pixels. L'opérateur NAGDIF a tendance à « raidir » les transitions. De ce fait, il donne en général des contours fins et ne demande pas à être suivi par une extraction des maximums locaux. Par contre, sur les transitions lentes, il crée un effet « d'escalier » ou de « dentelle ».

1.5 Opérateurs dérivatifs du deuxième ordre

Les contours peuvent être localisés par les passages par zéro du laplacien :

$$\Delta A(x, y) = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2}$$

En coordonnées polaires, cet opérateur s'exprime sous la forme :

$$\Delta A(r, \theta) = \frac{\partial^2 A}{\partial r^2} + \frac{1}{r} \frac{\partial A}{\partial r} + \frac{1}{r^2} \frac{\partial^2 A}{\partial \theta^2} \quad (5-28)$$

On remarque ainsi qu'il est invariant par rotation.

1.5.1 Opérateur laplacien sur voisinage réduit

L'approximation discrète la plus simple du laplacien, calculée sur un voisinage 3x3, correspond au masque suivant :

$$l_4 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (5-29)$$

1.5.2 Opérateur de Marr et Hildreth.

L'image $A(x, y)$ peut être prétraitée en effectuant un lissage par un filtre gaussien $g(x, y)$. Ensuite, on calcule la dérivée seconde dans la direction \mathbf{n} du gradient et on recherche les passages par zéro [MARR80] dans l'image \mathbf{B} définie par :

$$B(x, y) = \frac{\partial^2 (A * g(x, y))}{\partial n^2} \quad (5-30)$$

Pour simplifier, on suppose que les passages par zéro de la dérivée seconde directionnelle coïncident avec ceux du laplacien (*cf.* § 1.1). On peut alors utiliser le laplacien et l'expression (5-30) devient :

$$B(x, y) = \Delta(A * g(x, y)) \quad (5-31)$$

Cela revient (*cf.* annexe D) à appliquer directement sur l'image l'opérateur laplacien d'une gaussienne $\Delta g(x, y)$ qui peut être approché valablement par la

différence de deux gaussiennes (opérateur DOG). En effet, en monodimensionnel, l'opérateur DOG s'écrit :

$$\text{DOG}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} - \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_i^2}} \quad (5-32)$$

En posant $\sigma_i = \sigma + \delta\sigma$, il vient :

$$\sqrt{2\pi}\text{DOG}(x) = \frac{1}{\sigma} e^{-\frac{x^2}{2\sigma^2}} - \frac{1}{\sigma + \delta\sigma} e^{-\frac{x^2}{2(\sigma + \delta\sigma)^2}} \quad (5-33)$$

Il apparaît que :

$$\sqrt{2\pi}\text{DOG}(x) \approx \delta\sigma \frac{\partial}{\partial\sigma} \left(\frac{1}{\sigma} e^{-\frac{x^2}{2\sigma^2}} \right) = - \left(\frac{1}{\sigma^2} - \frac{x^2}{\sigma^4} \right) e^{-\frac{x^2}{2\sigma^2}} \quad (5-34)$$

Or :

$$\left(\frac{1}{\sigma^2} - \frac{x^2}{\sigma^4} \right) e^{-\frac{x^2}{2\sigma^2}} = \frac{\partial^2 g(x)}{\partial x^2} \quad (5-35)$$

DOG(x) approche donc la dérivée seconde de $g(x)$ si σ et σ_i sont voisins.

L'objectif est de réaliser un opérateur différentiel à bande étroite. En étudiant le filtre DOG, on observe que la bande passante est pratiquement constante pour un rapport $\sigma/\sigma_i < 1,6$. Sachant que le gain croît quand σ/σ_i augmente, on choisit pratiquement σ/σ_i voisin de 1,6.

1.5.3 Opérateur de Huertas-Médioni

Huertas et Médioni [HUER86] proposent un filtre qui est une décomposition en filtres séparables de l'opérateur laplacien d'une gaussienne $\Delta g(x, y)$:

$$\frac{1}{G_0} \left(2 - \frac{x^2 + y^2}{\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (5-36)$$

où G_0 est un coefficient de normalisation.

$\Delta g(x, y)$ peut s'exprimer sous la forme :

$$\Delta g(x, y) = g_1(x)g_2(y) + g_2(x)g_1(y) \quad (5-37)$$

avec :

$$g_1(x) = \frac{1}{\sqrt{2G_0}} \left(1 - \frac{x^2}{\sigma^2} \right) e^{-\frac{x^2}{2\sigma^2}} \text{ (dérivée seconde d'une gaussienne)} \quad (5-38)$$

$$\text{et : } g_2(x) = \frac{1}{\sqrt{2G_0}} e^{-\frac{x^2}{2\sigma^2}} \text{ (lissage)} \quad (5-39)$$

L'implantation de ces filtres est faite sous forme de filtres numériques à réponse impulsionnelle finie à coefficients entiers. Par exemple, pour $\sigma = \sqrt{2}$ et $\frac{1}{G_0} = 4232$, les masques suivants sont proposés :

$$\mathbf{g_1} = -1 \ -6 \ -17 \ -17 \ 18 \ 46 \ 18 \ -17 \ -17 \ -6 \ -1$$

$$\mathbf{g_2} = 0 \ 1 \ 5 \ 17 \ 36 \ 46 \ 36 \ 17 \ 5 \ 1 \ 0$$

1.6 Approches par filtrage optimal

Dans les approches suivantes, le contour est modélisé par un échelon d'amplitude U_0 noyé dans un bruit blanc (cf. figure 5.1). Plusieurs méthodes de calcul du filtre optimal pour détecter et localiser la transition sont proposées.

1.6.1 Approche de Canny

L'approche de Canny [CANN86] est présentée en *monodimensionnel*, elle consiste à trouver le filtre optimal de réponse impulsionnelle $h(x)$ satisfaisant les 3 contraintes suivantes pour un signal d'entrée en échelon :

- une bonne détection
- une bonne localisation,
- une faible multiplicité des maximums dus au bruit.

Soit $A(x)$ un signal monodimensionnel représentant un saut d'amplitude U_0 , noyé dans un bruit blanc stationnaire $N(x)$ de moyenne nulle et de d.s.p.(densité spectrale de puissance) N_0^2 :

$$A(x) = U_0 U(x) + N(x) \quad (5-40)$$

$$\text{où } U(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{autrement} \end{cases}$$

Le signal de sortie est donné par l'expression :

$$C(x) = A * h(x) = \int_{-\infty}^{+\infty} A(t) h(x-t) dt \quad (5-41)$$

Le problème posé est de trouver $h(x)$ tel que $C(x)$ soit maximum au point $x = 0$ en respectant les 3 contraintes énoncées précédemment.

bonne détection

Ce critère s'exprime à l'aide du rapport signal sur bruit RSB défini comme le rapport du maximum de la réponse due au signal seul sur la racine carrée de la puissance du bruit en sortie :

$$RSB = \frac{U_0 \int_0^{+\infty} h(x-t) dt}{\left(E \left\{ \left| \int_{-\infty}^{+\infty} N(t) h(x-t) dt \right|^2 \right\} \right)^{\frac{1}{2}}} = \frac{U_0 \int_0^{+\infty} h(x-t) dt}{N_0 \left(\int_{-\infty}^{+\infty} h^2(t) dt \right)^{\frac{1}{2}}} \quad (5-42)$$

Le filtre recherché est un dérivateur, afin d'obtenir une réponse nulle pour un signal d'entrée constant, $h(x)$ est choisie impaire. Ensuite, toujours en se plaçant en $x = 0$, on obtient :

$$RSB = \frac{U_0 \int_{-\infty}^0 h(t) dt}{N_0 \left(\int_{-\infty}^{+\infty} h^2(t) dt \right)^{\frac{1}{2}}} = \frac{U_0 \Sigma}{N_0} \quad (5-43)$$

bonne localisation

Le critère de localisation est mesuré par l'inverse de la variance de la distance entre le maximum de la réponse et la position réelle de la transition (il faudra donc le rendre maximum). Pour le calculer, on cherche à exprimer $E\{x_0^2\}$, x_0 étant la position calculée de la transition. Cette position x_0 correspond au maximum du signal de sortie, donc à un passage par zéro de sa dérivée première puisque le filtre étudié est un dérivateur (cf. §1.1). Nous avons donc :

$$\left[\frac{\partial \mathcal{C}}{\partial x} \right]_{x_0} = \left[\frac{\partial}{\partial x} A * h(x) \right]_{x_0} = [A * h'(x)]_{x_0} = 0 \quad (5-44)$$

$$\begin{aligned} \text{d'où : } \left[\frac{\partial \mathcal{C}}{\partial x} \right]_{x_0} &= U_0 [U * h'(x)]_{x_0} + [N * h'(x)]_{x_0} \\ &= U_0 \int_{-\infty}^{+\infty} U(x_0 - t) h'(t) dt + [N * h'(x)]_{x_0} \\ &= U_0 \int_{-\infty}^{x_0} h'(t) dt + [N * h'(x)]_{x_0} \\ &= U_0 h(x_0) + [N * h'(x)]_{x_0} \end{aligned} \quad (5-45)$$

Ensuite, on exprime le développement limité de la fonction $h(x)$ impaire et continue au voisinage de $x = 0$:

$$h(x_0) \approx h(0) + x_0 h'(0) \approx x_0 h'(0) \text{ puisque } h(0) = 0 \quad (5-46)$$

Il s'en suit que :

$$\left[\frac{\partial \mathcal{C}}{\partial x} \right]_{x_0=0} \approx U_0 x_0 h'(0) + N * h'(0) = 0$$

$$\text{donc : } E\{U_0 x_0 h'(0)\}^2 \approx E\{N * h'(0)\}^2 \quad (5-47)$$

L'expression ci-dessus devient :

$$U_0^2 h'^2(0) E\{x_0^2\} \approx N_0^2 \left(\int_{-\infty}^{+\infty} h'^2(t) dt \right) \quad (5-48)$$

$$\text{ce qui donne : } E\{x_0^2\} \approx \frac{N_0^2 \left(\int_{-\infty}^{+\infty} h'^2(t) dt \right)}{U_0^2 h'^2(0)} \quad (5-49)$$

Le critère de localisation est la racine carrée de l'inverse de l'expression ci-dessus :

$$\frac{U_0}{N_0} \Lambda = \left(\frac{U_0^2 h'^2(0)}{N_0^2 \left(\int_{-\infty}^{+\infty} h'^2(t) dt \right)} \right)^{\frac{1}{2}} = \frac{U_0}{N_0} \frac{|h'(0)|}{\left(\int_{-\infty}^{+\infty} h'^2(t) dt \right)^{\frac{1}{2}}} \quad (5-50)$$

Le produit $\Sigma\Lambda$ est un critère qui combine une bonne détection et une bonne localisation. Il ne dépend pas de l'amplitude U_0 de l'échelon, ni du facteur d'échelle κ . Pour démontrer cette dernière propriété, on pose : $h_\kappa(x) = h\left(\frac{x}{\kappa}\right)$ et on obtient :

$$\Sigma_\kappa = \sqrt{\kappa} \Sigma \quad \text{et} \quad \Lambda_\kappa = \frac{1}{\sqrt{\kappa}} \Lambda \quad (5-51)$$

$$\text{d'où } \Sigma_\kappa \Lambda_\kappa = \Sigma \Lambda. \quad (5-52)$$

non multiplicité des réponses

Pour l'optimisation, Canny propose de *maximiser le produit* $\Sigma\Lambda$ en utilisant une troisième contrainte qui est la minimisation de la densité d_0 de passages par zéro de la réponse due au bruit. Pour un processus gaussien $B(x)$ de moyenne nulle, on démontre que [RICE54] :

$$d_0 = \frac{1}{\pi} \left(\frac{-R_{BB}''(0)}{R_{BB}(0)} \right)^{\frac{1}{2}} \quad (5-53)$$

où $R_{BB}(0)$ est la fonction d'autocorrélation du signal. Posons $B(x) = N * h(x)$, où $N(x)$ est un bruit blanc gaussien, nous avons :

$$R_{BB}(0) = N_0^2 \int_{-\infty}^{+\infty} h^2(t) dt \quad (5-54)$$

$$\text{Compte tenu de la relation } R_{B'B'}(\tau) = -R_{BB}''(\tau) \quad (5-55)$$

Nous pouvons écrire :

$$R_{BB}''(0) = -N_0^2 \int_{-\infty}^{+\infty} h'^2(t) dt \quad (5-56)$$

Nous nous intéressons aux transitions du signal d'entrée qui correspondent aux extremums du signal de sortie du filtre dérivateur étudié. Ces extremums sont aussi les passages par zéro de la dérivée seconde du signal d'entrée, donc de la dérivée du signal de sortie. On exploite alors la formule (5-53) en remplaçant $B(x)$ par $B'(x)$:

$$d_0 = \frac{1}{\pi} \left(\frac{\int_{-\infty}^{+\infty} h'^2(t) dt}{\int_{-\infty}^{+\infty} h^2(t) dt} \right)^{\frac{1}{2}} \text{ et on pose } x_{moy} = \frac{1}{d_0} \quad (5-57)$$

Canny a choisi de prendre un filtre à réponse impulsionnelle finie de « taille » M . Ceci revient à remplacer, dans l'expression des critères, les bornes $-\infty + \infty$ par $-M$ et $+M$. Ensuite, il a imposé comme troisième contrainte que la distance x_{max} entre deux maximums consécutifs de la réponse due au bruit seul soit égale à une fraction de M . Ce critère correspond à la limitation du nombre de maximums locaux détectés en réponse à un seul contour. x_{max} est alors donné par :

$$x_{max} = 2 \cdot x_{moy} = 2\pi \left(\frac{\int_{-M}^{+M} h'^2(t) dt}{\int_{-M}^{+M} h^2(t) dt} \right)^{\frac{1}{2}} = k_m \cdot M \quad (5-58)$$

La maximisation du produit $\Sigma\Lambda$ pour un x_{max} égal à $k_m M$ revient à trouver $h(x)$ qui minimise la fonctionnelle :

$$\Psi(x, h, h', h'') = h^2 + \lambda_1 h'^2 + \lambda_2 h''^2 + \lambda_3 h \quad (5-59)$$

La solution optimale $h(x)$ satisfaisant les conditions de continuité et de dérivabilité est obtenue par calcul variationnel comme solution de l'équation d'Euler correspondant à une fonctionnelle du deuxième ordre :

$$\Psi_h - \frac{\partial}{\partial x} \Psi_{h'} + \frac{\partial^2}{\partial x^2} \Psi_{h''} = 0 \quad (5-60)$$

$$\text{où } \Psi_h = \frac{\partial \Psi}{\partial h}$$

La fonction $h(x)$ est alors solution de l'équation différentielle suivante :

$$2h(x) - 2\lambda_1 h''(x) + 2\lambda_2 h''''(x) + \lambda_3 = 0 \quad (5-61)$$

qui admet comme solution générale :

$$h(x) = a_1 e^{\alpha x} \sin \omega x + a_2 e^{\alpha x} \cos \omega x + a_3 e^{-\alpha x} \sin \omega x + a_4 e^{-\alpha x} \cos \omega x \quad (5-62)$$

avec :

$$\lambda_2 - \frac{\lambda_1^2}{4} > 0; \quad \alpha^2 - \omega^2 = \frac{\lambda_1}{2\lambda_2}; \quad 4\alpha^2\omega^2 = \frac{\lambda_1^2 - 4\lambda_2}{4\lambda_2^2}$$

Pour trouver l'opérateur $h(x)$ sous la forme d'un filtre à réponse impulsionnelle finie (RIF) défini sur l'intervalle $[-M, +M]$ et présentant une pente S à l'origine, Canny a imposé les conditions aux limites suivantes :

$$h(0) = 0; \quad h(M) = 0; \quad h'(0) = S; \quad h'(M) = 0 \quad (5-63)$$

Ces 4 conditions aux limites permettent de déterminer les coefficients a_1 à a_4 . $h(x)$ étant impaire, la solution est étendue aux x négatifs par $h(x) = -h(-x)$.

En utilisant une technique d'optimisation numérique sous contrainte, Canny démontre que l'opérateur le plus performant correspond à un produit $\Sigma\Lambda = 1,12$.

L'opérateur est complexe à mettre en œuvre. Canny, au vu de sa forme, propose une approximation par l'opérateur dérivée première d'une gaussienne qui présente un indice de performance $\Sigma\Lambda = 0,92$. Ce qui dégrade les performances de 20% par rapport à l'opérateur initial.

Une démarche intéressante pour caractériser un filtre dérivateur peut consister à étudier la probabilité d'un maximum « erroné » dans le voisinage de la position exacte de la transition. En considérant que la réponse du filtre dérivateur à un échelon est quasiment linéaire au voisinage de x_0 , on peut conclure qu'il n'y aura qu'un seul passage par zéro si la pente du signal de sortie est supérieure à la pente en sortie due au bruit seul. En exprimant P_s , la probabilité de cet événement et P_f , la probabilité de fausse alarme, Canny a établi la relation suivante :

$$\frac{|h'(0)|}{\left(\int_{-\infty}^{+\infty} h'^2(t) dt\right)^{\frac{1}{2}}} = k_s \frac{\int_{-\infty}^0 h(t) dt}{\left(\int_{-\infty}^{+\infty} h^2(t) dt\right)^{\frac{1}{2}}} \quad (5-64)$$

où k_s est une constante déterminée par les valeurs des probabilités P_s et P_f qui permet de les situer l'une par rapport à l'autre. Pour $k_s=1$, les probabilités sont égales. Le calcul pour le filtre optimal de Canny donne $k_s = 0,58$. Pour l'opérateur dérivée première d'une gaussienne, on obtient $k_s = 0,51$; ce qui correspond à une dégradation d'environ 10%.

1.6.2 Opérateur monodimensionnel de Deriche

Deriche [DERI87], utilisant la même démarche que Canny a cherché une réalisation de l'opérateur sous la forme d'un filtre à réponse impulsionnelle infinie (RII). Il a abouti à la même équation différentielle. Seules les conditions aux limites sont différentes ($M \rightarrow +\infty$) :

$$h(0) = 0; \quad h(+\infty) = 0; \quad h'(0) = S; \quad h'(+\infty) = 0$$

La solution est alors :

$$h(x) = ce^{-\alpha|x|} \sin \alpha x \quad (5-65)$$

En évaluant, pour cet opérateur, les différentes intégrales intervenant dans le calcul des critères de performance, on obtient les résultats suivants :

$$\Lambda = \sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{2\alpha}{\alpha^2 + \omega^2}} \quad (5-66)$$

$$\Sigma\Lambda = \frac{2\alpha}{\sqrt{\alpha^2 + \omega^2}} \quad k_s = \sqrt{\frac{\alpha^2 + \omega^2}{5\alpha^2 + \omega^4 + 6\alpha^2\omega^2}} \quad (5-67)$$

En posant $\alpha = m\omega$, on obtient les 3 cas suivants :

$$a) m \gg 1; \Lambda = \sqrt{2\alpha}; \Sigma = \sqrt{\frac{2}{\alpha}}; \Sigma\Lambda = 2; k_s = 0,44 \quad (5-68)$$

$$b) m = 1; \Lambda = \sqrt{2\alpha}; \Sigma = \sqrt{\frac{1}{\alpha}}; \Sigma\Lambda = \sqrt{2}; k_s = 0,58 \quad (5-69)$$

$$c) m = \sqrt{3}; \Lambda = \sqrt{2\alpha}; \Sigma = \sqrt{\frac{3}{2\alpha}}; \Sigma\Lambda = \sqrt{3}; k_s = 0,5 \quad (5-70)$$

Le dernier cas montre que pour une valeur identique de k_s , l'indice de performance de l'opérateur Deriche est nettement meilleur que celui obtenu avec la dérivée première d'une gaussienne.

Le deuxième cas montre que pour une valeur identique de k_s , l'opérateur de Deriche présente un indice de performance amélioré de près de 25% par rapport à l'opérateur optimal de Canny qui donnait $\Sigma\Lambda = 1,12$ et $k_s = 0,58$.

Le premier cas présente le meilleur indice de performance. Il correspond à la limite de l'opérateur Deriche pour ω tendant vers zéro. Il est facile de vérifier que cette limite correspond à l'opérateur $h(x)$ donné par :

$$h(x) = ce^{-\alpha|x|} \quad (5-71)$$

En fait, cet opérateur est la solution de l'équation différentielle (5-61) dans le cas où le discriminant de son équation caractéristique est nul.

c est une constante de « normalisation » calculée pour avoir un maximum de la réponse égal à 1 en $x = 0$ pour un échelon $U(x)$. On obtient alors :

$$c \int_{-\infty}^0 t e^{-\alpha t} dt = 1 \Rightarrow c = -\alpha^2 \quad (5-72)$$

1.6.3 Opérateurs bidimensionnels de dérivation et de lissage de Deriche

En utilisant l'opérateur optimal de Deriche, on peut mettre en œuvre les techniques générales énoncées précédemment. Ainsi, conformément au §1.2.a, on calcule le gradient en chaque point de l'image. Pour améliorer l'immunité au bruit un lissage préalable est effectué.

1.6.3.1 Lissage

Le filtre utilisé est la combinaison de deux filtres monodimensionnels dans les directions x et y . Le filtre de lissage monodimensionnel retenu par Deriche est l'intégrale $f(x)$ du filtre optimal $h(x) = cx e^{-\alpha|x|}$. Nous avons :

$$f(x) = b(\alpha|x| + 1)e^{-\alpha|x|} \quad (5-73)$$

b est calculé pour donner une réponse constante de valeur 1 pour un signal d'entrée constant de niveau 1; on obtient $b = \frac{\alpha}{4}$.

L'expression du filtre bidimensionnel séparable de lissage est donc de la forme:

$$f(x, y) = b^2(\alpha|x| + 1)e^{-\alpha|x|} (\alpha|y| + 1)e^{-\alpha|y|} \quad (5-74)$$

Si l'image originale est notée $A(x, y)$, l'image lissée aura pour expression :

$$B(x, y) = A * f(x, y) \quad (5-75)$$

1.6.3.2 Calcul du gradient

Le calcul du gradient se fait à partir des dérivées selon x et y du produit de convolution de l'image par le filtre de lissage $f(x, y)$.

Compte tenu des règles de dérivation de l'opération de convolution et de la séparabilité du filtre $f(x, y)$, nous avons par exemple :

$$\frac{\partial B}{\partial x}(x, y) = B_x(x, y) = \frac{\partial \{A * f\}}{\partial x}(x, y) = A * \frac{\partial f}{\partial x}(x, y) \quad (5-76)$$

Les opérateurs de dérivation suivant x et y sont de la forme :

$$f_x(x, y) = \eta x e^{-\alpha|x|} (\alpha|y| + 1)e^{-\alpha|y|} \quad (5-77)$$

$$f_y(x, y) = \eta y e^{-\alpha|y|} (\alpha|x| + 1)e^{-\alpha|x|} \quad (5-78)$$

où η est une constante de normalisation calculée, par exemple pour l'équation (5-77), de façon à fournir un maximum d'amplitude 1 en réponse à une transition verticale unitaire. On trouve alors que $\eta = -\frac{\alpha^3}{4}$; on en conclut que $f_x(x, y) = h(x)f(y)$. Par conséquent, l'image de la dérivée directionnelle en x s'écrit :

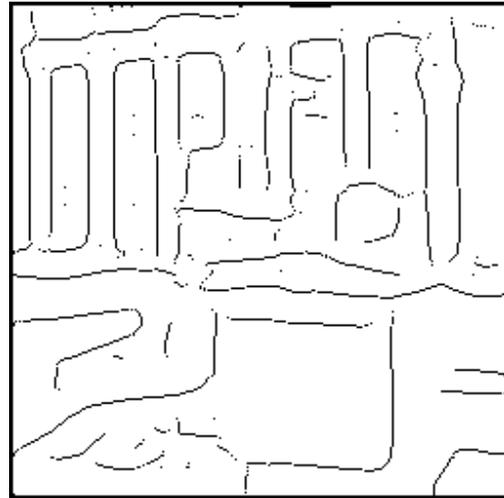
$$B_x(x, y) = (A * h(x)) * f(y) \quad (5-79)$$

C'est à dire que la dérivée directionnelle selon x est le résultat d'un lissage suivant la direction y , suivi par une dérivation suivant x . Les filtres sont monodimensionnels, l'implémentation récursive est aisée. (cf. annexe E)

Pour le filtre décrit par l'équation (5-78), on procède de même avec en entrée une transition horizontale unitaire.



Image originale



$\alpha = 0,2$



$\alpha = 0,5$



$\alpha = 1$

Figure 5.6 : Contours obtenus avec l'opérateur de Deriche pour différentes valeurs de α

Plusieurs résultats de détection de contours effectués à l'aide de l'opérateur dérivée première de Deriche sont présentés figure 5.6. Nous avons employé la démarche décrite au paragraphe 1.2.a. Après le calcul de la norme du gradient,

une extraction des maximums locaux et un seuillage par hystérésis permettent d'obtenir des chaînes de contours fins. On observe que la délocalisation est d'autant plus perceptible que le paramètre α est petit.

1.6.3.3 Opérateur laplacien de Deriche

Il est possible de calculer le laplacien pour la détection de contours. En effectuant le calcul sur l'image lissée, on obtient :

$$\Delta B(x, y) = A * \Delta f(x, y) \quad (5-80)$$

$$\text{Comme } \Delta f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (5-81)$$

en utilisant les formules (5-77) et (5-78), l'opérateur laplacien de Deriche s'exprime sous la forme d'un opérateur bidimensionnel dérivée seconde qui est la différence de deux fonctions de transfert séparables :

$$\Delta f(x, y) = e^{-\alpha|x|} \cdot e^{-\alpha|y|} - k\alpha|x|e^{-\alpha|x|} \cdot \alpha|y|e^{-\alpha|y|} \quad (5-82)$$

Le premier terme correspond à un filtre de lissage, le deuxième, à un filtre dérivateur.

Le calcul du laplacien d'une image $A(x, y)$ est obtenu par une simple soustraction entre deux images « traitées ». Il peut alors être mis en œuvre de manière totalement récursive suivant la démarche présentée en annexe E. Cette technique est très efficace car elle permet d'obtenir simplement le laplacien d'une image à n'importe quelle résolution, cette dernière étant spécifiée par le paramètre α (cf. figure 5.7).

Pour l'opérateur laplacien, la démarche utilisée est celle du paragraphe 1.2.c. On peut observer, dans certaines zones de l'image présentée sur la figure 5.7, que les contours sont moins réguliers que ceux fournis par l'opérateur dérivée première pour un même coefficient α (cf. figure 5.6). Ce phénomène apparaît souvent pour les opérateurs exploitant des dérivées secondes.



Image initiale BUREAU



Contours : passages par zéro du laplacien

Figure 5.7 : Passages par zéro du laplacien calculé à partir de l'opérateur de Deriche avec $\alpha = 1$

1.6.4 Opérateurs monodimensionnels de Shen

Shen et Castan [SHEN90], [CAST89] ont adopté la même démarche que celle exposée précédemment pour la détection de contour : l'image est lissée par des filtres passe-bas avant l'application d'un opérateur différentiel pour mettre en évidence les transitions. Ils déterminent donc, dans un premier temps, les caractéristiques d'un filtre optimal de lissage. Ensuite, en utilisant les fonctions de transfert de ce filtre, de ses dérivées première et seconde, ils proposent plusieurs solutions (présentées au paragraphe 1.6.5) pour la détection de contours.

Le calcul du filtre de lissage optimal de réponse impulsionnelle $f(x)$ se fait avec un critère qui combine la détection et la localisation. Pour cela, une approche voisine de celle de Canny est utilisée. La frontière est modélisée par un échelon d'amplitude U_0 noyé dans un bruit blanc stationnaire additionnel de moyenne nulle et de d.s.p. N_0^2 :

$$A(x) = U_0 U(x) + N(x)$$

Le signal en sortie du filtre est donné par :

$$B(x) = A(x) * f(x) = U_0 U(x) * f(x) + N(x) * f(x) \quad (5-83)$$

Le signal de sortie correspondant au bruit seul a pour valeur : $N(x) * f(x)$. Sa puissance moyenne est donnée par :

$$P_N^b = N_0^2 \int_{-\infty}^{+\infty} f^2(t) dt \quad (5-84)$$

Comme on souhaite exploiter la dérivée du signal lissé (prise au sens des distributions) pour détecter la transition, analysons son expression donnée par :

$$C(x) = \frac{dB}{dx}(x) = U_0 \delta(x) * f(x) + N(x) * \frac{df}{dx}(x) = U_0 f(0) + N(x) * f'(x) \quad (5-85)$$

où δ est la distribution de Dirac.

Il est possible de dégager de cette expression les contributions provenant du signal et du bruit seul et de calculer leur puissance :

$$P_U^c = U_0^2 f^2(0) \quad (5-86)$$

$$P_N^c = N_0^2 \int_{-\infty}^{+\infty} f'^2(t) dt \quad (5-87)$$

Pour bien détecter le signal, les auteurs ont choisi de maximiser P_U^c et de minimiser P_N^c et P_N^b ; ceci revient à minimiser l'expression suivante :

$$CR = \frac{\sqrt{\frac{P_N^b P_N^c}{(P_N^c)^2}}}{\sqrt{\frac{\left(\int_{-\infty}^{+\infty} f^2(t) dt\right) \left(\int_{-\infty}^{+\infty} f'^2(t) dt\right)}{f^2(0)}}} \quad (5-88)$$

Un certain nombre de considérations présentées dans [SHEN92] amènent à choisir la fonction $f(x)$ paire et à analyser les performances du filtre sur une fenêtre de taille $2M$. Le critère s'exprime alors sous la forme :

$$CR^2 = \frac{\left(\int_0^{+M} f^2(t) dt\right) \left(\int_0^{+M} f'^2(t) dt\right)}{f^4(0)} \quad (5-89)$$

Minimiser ce critère revient à minimiser la fonctionnelle :

$$\Psi(x, f, f') = f'^2 + \lambda f^2$$

Pour cela, on exploite l'équation d'Euler du premier ordre :

$$\Psi_f - \frac{\partial \Psi_{f'}}{\partial x} = 0$$

$$\text{où } \Psi_f = \frac{\partial \Psi}{\partial f}$$

pour aboutir à l'équation différentielle :

$$2f'(x) - 2\lambda f(x) = 0$$

dont la solution est de la forme :

$$f(x) = b_1 e^{\beta x} + b_2 e^{-\beta x} \quad (5-90)$$

En tenant compte des conditions aux limites $M \rightarrow +\infty$, $\lim_{x \rightarrow +\infty} f(x) = 0$ et en considérant que $f(x)$ est une fonction paire, le filtre optimal est donné par :

$$f(x) = b e^{-\beta|x|} \quad (5-91)$$

où b est une constante de normalisation définie par :

$$\int_{-\infty}^{+\infty} f(t) dt = 1 \text{ soit } b = \frac{\beta}{2} \quad (5-92)$$

La transformée de Fourier de $f(x)$ vaut :

$$TFf(\omega) = \frac{\beta^2}{\beta^2 + \omega^2} = \frac{\beta}{\beta + j\omega} \cdot \frac{\beta}{\beta - j\omega} \quad (5-93)$$

On en déduit que $f(x)$ peut se mettre sous la forme :

$$f(x) = f_1 * f_2(x) \quad (5-94)$$

avec :

$$f_1(x) = \begin{cases} \beta e^{-\beta x} & \text{pour } x \geq 0 \\ 0 & \text{pour } x < 0 \end{cases} \quad (5-95)$$

$$f_2(x) = \begin{cases} 0 & \text{pour } x > 0 \\ \beta e^{\beta x} & \text{pour } x \leq 0 \end{cases} \quad (5-96)$$

1.6.4.1 Opérateurs de dérivation de Shen

Le problème est de détecter un contour après ce lissage, ceci peut être fait, comme nous l'avons déjà énoncé, en cherchant les maximums de la dérivée première ou les passages par zéro de la dérivée seconde.

Calculons les dérivées première et seconde :

$$f'(x) = \begin{cases} -\frac{\beta^2}{2} e^{-\beta x} & \text{pour } x > 0 \\ \frac{\beta^2}{2} e^{\beta x} & \text{pour } x < 0 \end{cases} \quad (5-97)$$

$$f''(x) = \frac{\beta^2}{2} (\beta e^{-\beta|x|} + 2\delta(x)) \quad (5-98)$$

Avec ces relations, nous pouvons écrire que :

$$f'(x) = \frac{\beta}{2} (f_2(x) - f_1(x)) \quad (5-99)$$

$$f''(x) = \beta^2 (f(x) - \delta(x)) \quad (5-100)$$

Si $A(x)$ est l'image originale et $B(x)$ l'image lissée, nous avons :

$$B(x) = A * f(x) = (A * f_1(x)) * f_2(x) \quad (5-101)$$

$$B'(x) = \frac{\beta}{2} (A * f_2(x) - A * f_1(x)) \quad (5-102)$$

$$B''(x) = \beta^2 (A * f(x) - A(x)) \quad (5-103)$$

Il s'ensuit un calcul simple des dérivées première et seconde.

Remarque

L'opérateur $f'(x)$ ci-dessus n'est pas « normalisé ». Si l'on souhaite une réponse de valeur 1 en $x = 0$ pour une entrée de type « échelon unitaire », $f'(x)$ a pour nouvelle expression :

$$f'(x) = f_2(x) - f_1(x) \quad (5-104)$$

1.6.4.2 Performances théoriques de l'opérateur dérivateur de Shen

Les critères définis par Canny ne sont pas tous applicables au filtre dérivateur de Shen du fait que sa réponse impulsionnelle est discontinue en 0. Seul le rapport signal sur bruit est calculable :

$$RSB = \frac{U_0 \int_{-\infty}^0 f'(t) dt}{N_0 \left(\int_{-\infty}^{+\infty} f'^2(t) dt \right)^{\frac{1}{2}}}$$

$$RSB = \frac{U_0 \int_{-\infty}^0 e^{-\beta|x|} dx}{N_0 \left(\int_{-\infty}^0 e^{2\beta x} dx + \int_0^{+\infty} e^{-2\beta x} dx \right)^{\frac{1}{2}}} = \frac{U_0}{N_0} \frac{1}{\sqrt{\beta}} \quad (5-105)$$

1.6.5 Opérateurs bidimensionnels de Shen

Le filtre séparable de lissage bidimensionnel est une extension du filtre monodimensionnel précédent, il s'écrit :

$$f(x, y) = \frac{\beta^2}{4} e^{-\beta(|x|+|y|)} = f(x).f(y) \quad (5-106)$$

1.6.5.1 Calcul du gradient

Du fait de la séparabilité, les opérateurs normalisés de dérivation première s'expriment simplement en utilisant l'équation (5-104) :

$$\frac{\mathcal{F}}{\partial x}(x, y) = f(y)(f_2(x) - f_1(x)) \quad (5-107)$$

$$\frac{\mathcal{F}}{\partial y}(x, y) = f(x)(f_2(y) - f_1(y)) \quad (5-108)$$

Si A(x, y) est l'image d'entrée, les images de la dérivée première suivant x et suivant y auront pour expression :

$$C_x(x, y) = [(A * f(y)) * f_2(x) - (A * f(y)) * f_1(x)] \quad (5-109)$$

$$C_y(x, y) = [(A * f(x)) * f_2(y) - (A * f(x)) * f_1(y)] \quad (5-110)$$

1.6.5.2 Calcul du laplacien

En appelant B(x, y) l'image lissée par le filtre f(x, y), nous avons :

$$\Delta B(x, y) = A * \Delta f(x, y) = A * \frac{\partial^2 f}{\partial x^2}(x, y) + A * \frac{\partial^2 f}{\partial y^2}(x, y) \quad (5-111)$$

Les opérateurs de dérivation seconde ayant pour expression :

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2}(x, y) &= \beta^2 f(y)(f(x) - \delta(x)) \\ \frac{\partial^2 f}{\partial y^2}(x, y) &= \beta^2 f(x)(f(y) - \delta(y))\end{aligned}\tag{5-112}$$

On peut alors écrire que :

$$\Delta B(x, y) = \beta^2 (2.A * f(x, y) - A * f(x) - A * f(y))\tag{5-113}$$

La démarche utilisée pour obtenir les résultats de la figure 5.8 est la même que pour l'exploitation de l'opérateur de Deriche. Quand β diminue des contours disparaissent si on travaille à seuils constants. A l'encontre de l'opérateur de Deriche, la délocalisation ici est beaucoup moins perceptible.

On observe sur la figure 5.9, comme pour l'opérateur de Deriche, que le mécanisme de dérivation seconde fait apparaître des contours irréguliers.

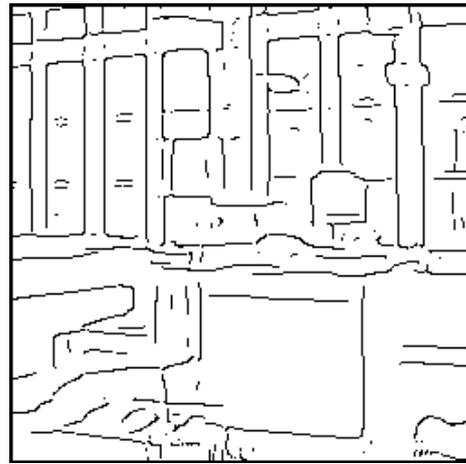
Remarque : l'implantation récursive des différents opérateurs de Shen est décrite en annexe E.

1.7 Comparaison des performances théoriques de quelques opérateurs de dérivation

Nous avons choisi d'exploiter les critères définis par Canny et présentés au §1.6.1. Cette comparaison est faite en exploitant, dans un premier temps, les expressions continues des réponses impulsionnelles des filtres, puis dans un deuxième temps, les expressions discrètes. Pour ce dernier cas, nous proposons une extension au cas discret des critères de Canny [DEMI94] présentée en annexe F. Il convient de placer les opérateurs dans les « mêmes conditions ». Nous avons décidé de fixer leur paramétrage de telle sorte que les puissances des réponses impulsionnelles : $\int_{-\infty}^{+\infty} h^2(t) dt$ soient égales. Il s'en suit que les rapports signal à bruit en sortie, calculés pour une entrée de type échelon, sont identiques.



Image originale



$\beta=0,14$



$\beta=0,35$



$\beta=0,7$

Figure 5.8 : Contours obtenus avec l'opérateur de Shen

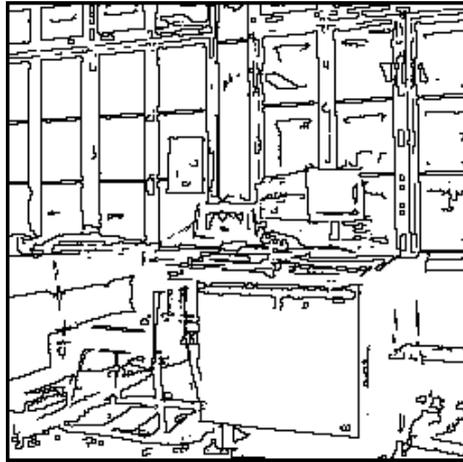


Figure 5.9 : Passages par zéro du laplacien calculé à partir de Shen avec $\beta = 0,5$

1.7.1 Critères continus de comparaison

Nous rappelons les expressions des filtres de dérivation première de Deriche, de Shen, de la dérivée première du filtre gaussien et nous donnons l'expression de leur puissance :

opérateur de Deriche

$$h_D(x) = -\alpha^2 x e^{-\alpha|x|} \text{ et } \int_{-\infty}^{+\infty} h_D^2(t) dt = \frac{\alpha}{2} \quad (5-114)$$

opérateur de Shen

$$h_S(x) = f_2(x) - f_1(x) \text{ et } \int_{-\infty}^{+\infty} h_S^2(t) dt = \beta \quad (5-115)$$

$$f_1(x) = \begin{cases} \beta e^{-\beta x} & \text{pour } x \geq 0 \\ 0 & \text{pour } x < 0 \end{cases}$$

$$f_2(x) = \begin{cases} 0 & \text{pour } x > 0 \\ \beta e^{\beta x} & \text{pour } x \leq 0 \end{cases}$$

dérivée première du filtre gaussien

$$h_G(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \text{ et } \int_{-\infty}^{+\infty} h_G^2(t) dt = \frac{\sqrt{\pi}}{2\sigma} \quad (5-116)$$

$$\text{L'égalité des puissances implique que } \alpha = 2\beta = \frac{\sqrt{\pi}}{\sigma} \quad (5-117)$$

Dans les conditions ci-dessus, les rapports signal sur bruit sont identiques. La comparaison des opérateurs se fait alors sur les deux derniers critères de Canny. Ceux-ci ne sont calculables que pour les opérateurs de Deriche et dérivée d'une gaussienne car l'expression de la fonction de filtrage de Shen n'est pas dérivable en $x = 0$. Nous avons alors les valeurs suivantes pour le critère de localisation :

$$\Lambda_D = \sqrt{2\alpha} \text{ et } \Lambda_G = \left(\frac{4}{3\sigma\sqrt{\pi}} \right)^{\frac{1}{2}} \quad (5-118)$$

$$\text{ce qui donne : } \left(\frac{\Lambda_G}{\Lambda_D} \right)^2 = \frac{2}{3\sigma^2} \quad (5-119)$$

Le rapport signal sur bruit de l'opérateur de Deriche croît quand α diminue (cf. relation (5-66)). Pour $\sigma > \sqrt{\frac{2}{3}}$, l'opérateur de Deriche localise mieux la transition que l'opérateur dérivée première d'une gaussienne (cf. relation (5-119)). Le rapport signal sur bruit de ce dernier a pour expression :

$$\Sigma_D = \left(\frac{2\sigma}{\sqrt{\pi}} \right)^{\frac{1}{2}} \quad (5-120)$$

il croît quand σ augmente mais alors la localisation devient mauvaise (cf. relation (5-118)). Pour les deux opérateurs, il faut trouver le compromis entre une bonne détection et une bonne localisation.

1.7.2 Critères discrets de comparaison

Si les critères de Canny constituent une bonne base de comparaison des différents opérateurs de dérivation, ils ne sont pas applicables aux filtres dont la réponse impulsionnelle est discontinue en zéro (Shen) ou aux filtres directement synthétisés dans le domaine discret (par exemple, de type DOB : *Difference Of Boxes*). Les critères de Canny dans leur forme discrète (cf. annexe F) permettent de comparer tous les filtres linéaires du type dérivée première dans leur contexte réel d'utilisation : le domaine des images et des opérateurs échantillonnés.

Dans le paragraphe 1.7.1, il a été précisé que pour régler de manière relative les différents paramètres des filtres en vue de leur comparaison, on pouvait chercher à égaliser leur puissance. D'autre part, les opérateurs de type dérivée première sont normalisés pour que le maximum de la réponse à un échelon unitaire soit d'amplitude unité. Ces deux conditions donnent :

$$\sum_{-\infty}^{+\infty} h^2(n) = \text{constante} \text{ et } \sum_{-\infty}^0 h(n) = 1 \quad (5-121)$$

Il en résulte que la comparaison des opérateurs se fait à critère de bonne détection (Cd1) constant.

Mis à part pour le filtre DOG (dérivée de gaussienne), les valeurs de Cd1 peuvent être calculées analytiquement :

$$Cd1_{Shen} = \left(\frac{1 + e^{-\beta}}{2 \cdot (1 - e^{-\beta})} \right)^{\frac{1}{2}} ; Cd1_{Deriche} = \frac{(1 + e^{-\alpha})^2}{(2 \cdot (1 - e^{-4\alpha}))^{\frac{1}{2}}} \quad (5-122)$$

$$Cd1_{DOB} = \frac{(N-1)^{\frac{1}{2}}}{2} \quad (5-123)$$

Quatre filtres sont comparés : Shen, Deriche, dérivée de gaussienne (ou DOG) et DOB. Ce dernier filtre a une réponse impulsionnelle RIF du type : $-1 \quad -1 \quad 0 \quad 1 \quad 1$. On note N le nombre total de coefficients (ici N = 5).

Le tableau 5.1 précise pour l'évaluation effectuée, les valeurs de Cd1 et les paramètres correspondant de réglages des filtres.

Cd1	0.71	1.00	1.41	2.00	2.83	4.00	5.65
(N) DOB	3	5	9	17	33	65	129
(β) Shen	25	1.10	0.51	0.25	0.125	0.0625	0.0311
(α) Deriche	25	1.66	0.90	0.48	0.25	0.125	0.0625
(σ) DOG	0.14	1.05	1.86	3.60	7.10	14.2	28.4

Tableau 5.1 : Paramètres de réglage des filtres en fonction du critère Cd1

1.7.2.1 Faible multiplicité des réponses

Le critère Cd3 correspond à la distance moyenne qui sépare deux maximums du bruit en sortie, lorsque le bruit d'entrée est blanc et gaussien (cf annexe F).

$$Cd3 = \frac{2\pi}{\arcsin(\rho)} \quad (5-124)$$

Analytiquement on obtient :

- pour le filtre DOB : $\rho = 0.25$ si N = 3, et $\rho = -0.25$ pour N > 3.
- pour le filtre de Shen : $\rho = \frac{1 - 3e^{-\beta}}{4}$
- pour le filtre de Deriche : $\rho = \frac{(1 - e^{-\alpha}) \cdot (5 + 4e^{-\alpha} + e^{-2\alpha})}{4} - 1$

Sur la figure 5.10 sont représentées les variations de Cd3 en fonction de Cd1. On peut remarquer les différences de performance pour ce critère entre les filtres continus (DOG et Deriche) et les filtres discontinus en zéro (Shen et DOB). Pour

ces derniers, une borne supérieure égale à 6 limite les valeurs possibles de $Cd3$. Il apparaît clairement que DOG est le meilleur filtre vis-à-vis de ce critère.

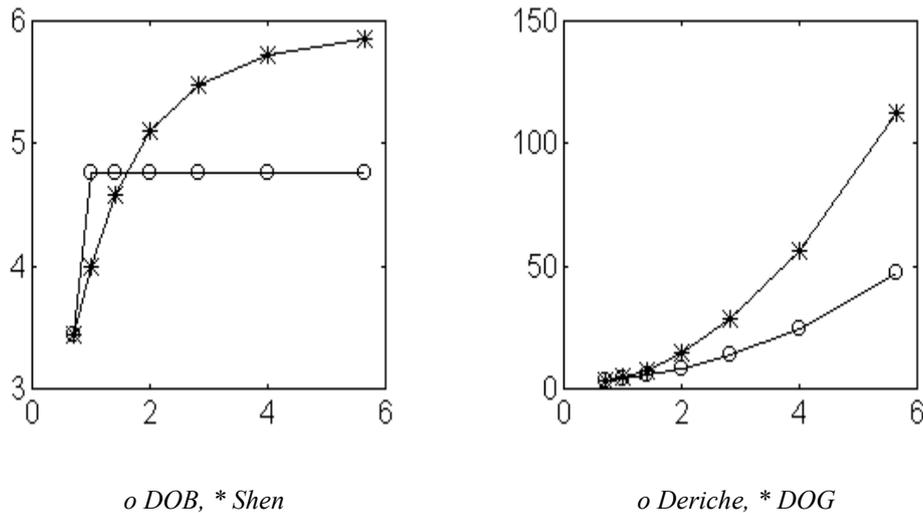


Figure 5.10 : Comparaison de filtres dérivateurs suivant le critère $Cd3=f(Cd1)$

1.7.2.2 Bonne localisation ($Cd2$)

Le critère de localisation dépend du rapport signal à bruit (RSB) à l'entrée. Pour notre étude nous avons retenu les RSB suivants :

RSB en entrée	1.5	3	6	12	24
---------------	-----	---	---	----	----

Lorsque le RSB augmente, la délocalisation devient quasiment nulle. Pour un RSB de 24, la distance quadratique moyenne ($1/Cd2$) est inférieure à $1/1000$, ce qui signifie dans le cas discret que le contour est détecté décalé d'un pixel dans un cas sur mille ! Pour un RSB de 12, seul le filtre DOG présente une délocalisation significative (cf. tableau 5.3).

La figure 5.11 montre les valeurs de $1/Cd2$ en fonction de $Cd1$ pour des RSB de 1.5, 3 et 6.

On constate nettement que la contrepartie d'une valeur élevée de $Cd3$ est une perte en localisation pour les filtres de réponse impulsionnelle discontinue en zéro. Par contre, les filtres Shen et DOB ont une délocalisation largement indépendante du facteur d'échelle. Dans ce cas, en première approximation, la délocalisation ne dépend que du RSB.

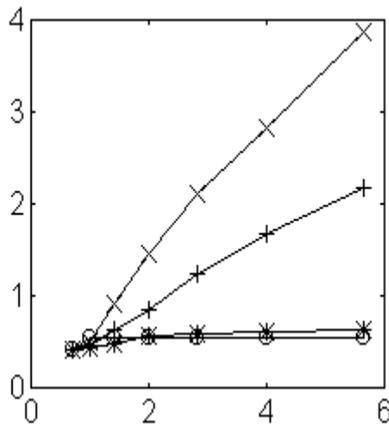
1.7.2.3 Filtrés lisseurs

Le critère C11, amélioration du rapport signal à bruit (réduction en amplitude du bruit) a une signification naturelle pour les filtres de lissage. Il est cependant moins fréquent d'appliquer à ces filtres, le critère Cd3 : distance moyenne entre maximums du bruit, qui pourtant traduit mieux le terme « lissage ». La démarche est la même que pour les filtres dérivateurs. C11 fixe de manière relative les paramètres des filtres (cf. tableau 5.2). Le *boxcar* est le filtre RIF dont les N coefficients de la réponse impulsionnelle sont égaux à 1.

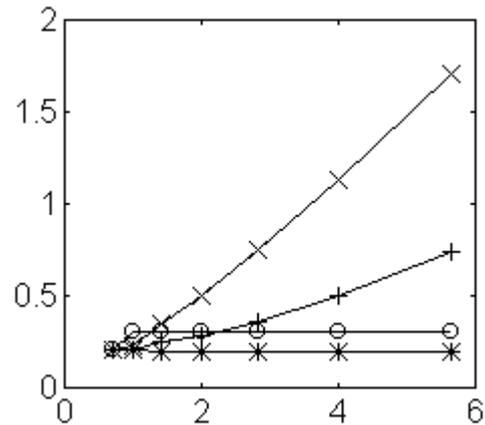
C11	1.00	1.73	2.23	3.00	4.12	5.75	8.06
(N) Boxcar	1	3	5	9	17	33	65
(β) Shen	50	1.16	0.74	0.43	0.23	0.12	0.061
(α) Deriche	50	2.04	1.27	0.71	0.38	0.19	0.099
(σ) gaussienne	0.10	0.84	1.41	2.54	4.80	9.31	18.33

Tableau 5.2 : Paramètres de réglages des filtres de lissage en fonction du critère Cd1.

Toutes les réponses impulsionnelles des lisseurs étant continues en zéro, Cd3 augmente avec C11 (cf. figure 5.11). On constate la nette supériorité du filtre gaussien pour Cd3.



a) $1/Cd2 \text{ RSB} = 1.5$
o DOB, * Shen, + Deriche, x DOG



c) $1/Cd2 \text{ RSB} = 3$
o DOB, * Shen, + Deriche, x DOG

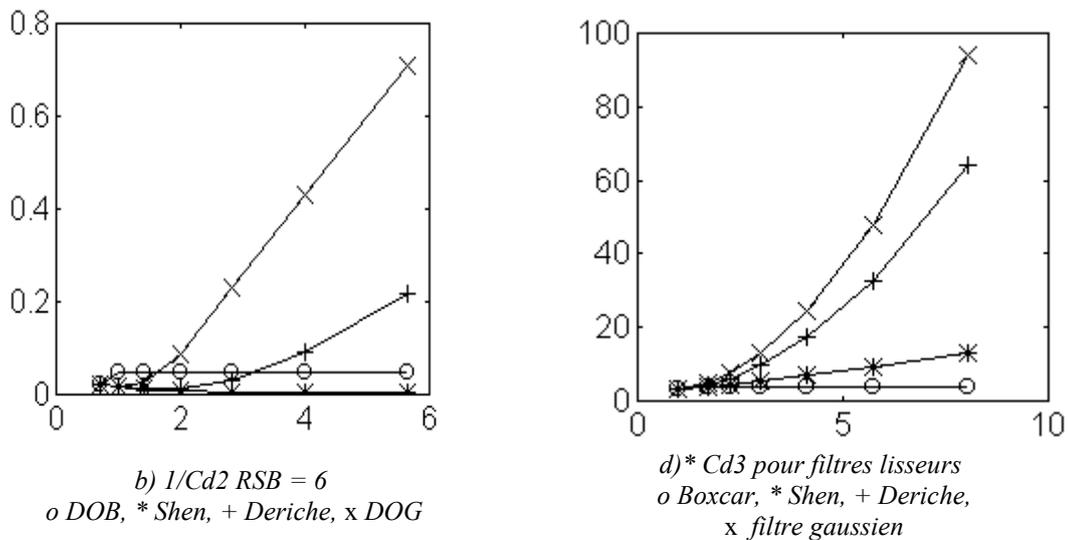


Figure 5.11 : Valeurs de la délocalisation $1/Cd2$ pour les filtres dérivateurs
Valeurs du critère Cd3 pour les filtres de lissage

Cd1	0.71	1.00	1.41	2.00	2.83	4.00	5.65
DOG : $1/Cd2$ (RSB=12)	0.000	0.000	0.000	0.000	0.010	0.076	0.226

Tableau 5.3 : valeur du critère $1/Cd2$ pour le filtre DOG

2 Modèle Surfique

Dans l'exposé des méthodes suivantes, l'image des intensités $A(x, y)$ est considérée comme une surface. Plusieurs approches sont envisageables.

Il est possible de définir un modèle surfique de transition $S(x, y)$, par exemple : un échelon bidimensionnel sur un voisinage donné. Un contour sera détecté en (x, y) s'il y a une bonne corrélation entre le modèle et la fenêtre de l'image centrée en ce point (méthode de Hueckel).

Huertas et Médioni utilisent l'image $D(x, y)$ résultant de l'application du laplacien d'une gaussienne sur l'image initiale $A(x, y)$. Au voisinage d'un passage par zéro, l'image $D(x, y)$ est approximée par un polynôme cubique. Les passages par zéro identifiant les contours sont détectés de manière analytique par interpolation à partir de l'équation du polynôme. L'objectif est de localiser la frontière avec une précision inférieure au pixel. Cette approche est traitée au paragraphe 5.2.2 relatif à la localisation.

Une autre méthode consiste à approcher la surface représentant une transition par un polynôme dont l'équation sera utilisée pour obtenir l'orientation du contour (approche proposée par Haralick).

2.1 Opérateur de Hueckel

Le principe de l'approche proposée par Hueckel [HUEC71] est de se donner un modèle idéal de transition représentée par un échelon bidimensionnel U dans un voisinage circulaire \mathcal{D} .

$$U(x, y, \theta, \rho, b, h) = \begin{cases} b & \text{si } x \cos \theta + y \sin \theta < \rho \\ b + h & \text{si } x \cos \theta + y \sin \theta \geq \rho \end{cases} \quad (5-125)$$

Soit $A(x, y)$ l'image des niveaux de gris. Le problème est de calculer les paramètres du modèle qui approche au mieux l'image en un point donné (x, y) centre d'un domaine circulaire \mathcal{D} . Pour cela, on minimise l'écart quadratique entre l'image et le modèle :

$$\varepsilon^2 = \iint_{\mathcal{D}} (U(x, y, \theta, \rho, b, h) - A(x, y))^2 dx dy \quad (5-126)$$

Pour réaliser cette minimisation, Hueckel décompose U et $A(x, y)$ sur une base de fonctions orthogonales définies dans le domaine de Fourier. Afin de diminuer le temps de calcul requis par cette technique assez lourde, la présence ou l'absence de contours dans le disque de recherche est d'abord testée par un opérateur de type gradient. Si un contour potentiel est détecté, l'algorithme de Hueckel est mis en œuvre sur une base limitée à seulement huit termes. Une étape finale utilise alors un opérateur de décision basé sur des techniques de seuillage afin de choisir les éléments de contour valables.

2.2 Opérateur de Haralick

2.2.1 Rappel sur l'approximation polynômiale

Le problème est de faire correspondre un polynôme $P(x)$ avec n points. Pour cet ensemble de points, il est possible de trouver n polynômes $P_{nk}(x)$ orthogonaux de degré $k \leq n$ et caractérisés par la propriété :

$$\sum_{x=0}^{n-1} P_{nu}(x) P_{nv}(x) = 0 \text{ avec } u \neq v \quad (5-127)$$

Tout polynôme $P(x)$ de degré $k \leq n-1$ peut être exprimé suivant une combinaison linéaire de polynômes orthogonaux $P_{nk}(x)$:

$$P(x) = \sum_{k=0}^m a_m P_{nk}(x) \quad (5-128)$$

On démontre en minimisant l'erreur quadratique moyenne :

$$\varepsilon^2 = \sum_{x=0}^{n-1} \left(P(x) - \sum_{k=0}^m a_m P_{nk}(x) \right)^2 \quad (5-129)$$

que les coefficients a_m sont donnés par la relation :

$$a_k = \frac{\sum_{x=0}^{n-1} P(x)P_{nk}(x)}{\sum_{x=0}^{n-1} P_{nk}^2(x)} \tag{5-130}$$

2.2.2 Polynômes monodimensionnels discrets de Tchebycheff

Pour un ensemble symétrique I de n points tel que $i \in I \Rightarrow -i \in I$, on définit une base de $m+1$ polynômes orthogonaux :

$$P_0 \ i = 1$$

..... (5-131)

$$P_m \ i = k_0 + k_1 i + k_2 i^2 + \dots + k_{m-1} i^{m-1} + i^m$$

avec $m \leq n-1$ et tels que :

$$\sum_{i=0}^{n-1} P_u \ i \ P_v \ i = 0 \text{ avec } u \neq v$$

Par exemple, pour $n=5$, $i \in \{-2,-1,0,1,2\}$, on trouve :

$$P_0 \ i = 1 \qquad P_2 \ i = i^2 - 2 \qquad P_4 \ i = \frac{i^4 + 210i^2 - 144}{160}$$

$$P_1 \ i = i \qquad P_3 \ i = i^3 - \frac{17}{5}i$$

(5-132)

2.2.3 Approximation polynômiale bidimensionnelle

Dans l'approche d'Haralick [HARA84], la portion de l'image $A[i, j]$ située dans une fenêtre de taille $N \times N$ centrée en $[i, j]$ est approximée par le polynôme dont le degré maximum en i ou j est $N-1$.

On dispose donc d'un ensemble de N^2 points pour lesquels on construit une base de N^2 polynômes résultant du produit cartésien des deux bases de polynômes monodimensionnels discrets :

$$\{P_0[i], \dots, P_{N-1}[i]\} \times \{P_0[j], \dots, P_{N-1}[j]\} = \{P_{00}[i, j], P_{01}[i, j], \dots, P_{(N-1)(N-1)}[i, j]\} \tag{5-133}$$

Sur le domaine de taille $N \times N$, $A[i, j]$ peut alors être approchée par :

$$\hat{A} \ i, j = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} k_{nm} P_{nm} \ i, j \tag{5-134}$$

Les expressions du paragraphe 2.2.1 sont généralisables au cas bidimensionnel et les coefficients k_{nm} sont donnés par l'expression :

$$k_{mn} = \frac{\sum_{i=-M}^M \sum_{j=-M}^M A_{i,j} P_{mn}(i,j)}{\sum_{i=-M}^M \sum_{j=-M}^M P_{mn}^2(i,j)} \quad (5-135)$$

$$\text{avec } M = \begin{cases} \frac{N-1}{2} & \text{si } N \text{ impair} \\ \frac{N}{2} & \text{si } N \text{ pair} \end{cases}$$

Le coefficient k_{mn} peut être interprété comme le résultat de la corrélation de l'image avec un masque H_{mn} dont les coefficients sont donnés par :

$$H_{mn}(u,v) = \frac{P_{mn}(i,j)}{\sum_{i=-M}^M \sum_{j=-M}^M P_{mn}^2(i,j)} \quad u = i + M + 1 \text{ et } v = j + M + 1$$

A partir de la décomposition sur une base orthogonale, il est possible de calculer les dérivées directionnelles par rapport à i et j données par :

$$\hat{A}_i(i,j) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} k_{nm} \frac{\partial P_{nm}}{\partial i}(i,j)$$

$$\hat{A}_j(i,j) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} k_{nm} \frac{\partial P_{nm}}{\partial j}(i,j)$$

2.2.4 Exemple de calcul de masques pour l'opérateur d'Haralick

Pour un voisinage 3x3 avec $[i,j] \in \{-1,0,1\} \times \{-1,0,1\}$ nous avons les polynômes suivants :

$$\begin{aligned} P_{00}[i,j] &= 1 & P_{02}[i,j] &= j^2 - \frac{2}{3} \\ P_{10}[i,j] &= i & P_{12}[i,j] &= i \left[j^2 - \frac{2}{3} \right] \\ P_{01}[i,j] &= j & \text{et} & \\ P_{11}[i,j] &= ij & P_{21}[i,j] &= j \left[i^2 - \frac{2}{3} \right] \\ P_{20}[i,j] &= i^2 - \frac{2}{3} & P_{22}[i,j] &= \left[i^2 - \frac{2}{3} \right] \left[j^2 - \frac{2}{3} \right] \end{aligned} \quad (5-136)$$

On obtient les 9 masques suivants :

$$\begin{aligned}
 \mathbf{H}_{00} &= \frac{1}{9} \begin{bmatrix} +1 & +1 & +1 \\ +1 & +1 & +1 \\ +1 & +1 & +1 \end{bmatrix} \mathbf{H}_{10} = \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} \mathbf{H}_{01} = \frac{1}{6} \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \\
 \mathbf{H}_{20} &= \frac{1}{6} \begin{bmatrix} +1 & +1 & +1 \\ -2 & -2 & -2 \\ +1 & +1 & +1 \end{bmatrix} \mathbf{H}_{11} = \frac{1}{4} \begin{bmatrix} +1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & +1 \end{bmatrix} \mathbf{H}_{02} = \frac{1}{6} \begin{bmatrix} +1 & -2 & +1 \\ +1 & -2 & +1 \\ +1 & -2 & +1 \end{bmatrix} \\
 \mathbf{H}_{12} &= \frac{1}{4} \begin{bmatrix} -1 & +2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix} \mathbf{H}_{21} = \frac{1}{4} \begin{bmatrix} -1 & 0 & +1 \\ +2 & 0 & -2 \\ -1 & 0 & +1 \end{bmatrix} \mathbf{H}_{22} = \frac{1}{4} \begin{bmatrix} +1 & -2 & +1 \\ -2 & +1 & -2 \\ +1 & -2 & +1 \end{bmatrix}
 \end{aligned} \tag{5-137}$$

Calculons $\frac{\partial A}{\partial i}$ 0,0 à l'aide de la formule :

$$\hat{A}_i \ i, j = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} k_{nm} \frac{\partial}{\partial i} P_{mn} \ i, j$$

Il s'avère que pour le cas étudié, toutes les valeurs $\frac{\partial P_{mn}}{\partial i}$ 0,0 sont nulles sauf $\frac{\partial P_{10}}{\partial i}$ 0,0 qui vaut 1 et $\frac{\partial P_{12}}{\partial i}$ 0,0 qui vaut $-\frac{2}{3}$. Il s'en suit l'expression de la dérivée directionnelle selon i :

$$\hat{A}_i \ 0,0 = k_{10} \frac{\partial P_{10}}{\partial i} \ 0,0 + k_{12} \frac{\partial P_{12}}{\partial i} \ 0,0 \tag{5-138}$$

qui fournit le masque de l'opérateur de dérivation correspondant :

$$\frac{1}{2} \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & +1 & 0 \end{bmatrix}$$

2.2.5 Estimation de la direction du gradient

Il est important de remarquer que les masques utilisés pour estimer la dérivée première dépendent de la taille du voisinage, de l'ordre de la fonction d'approximation et de sa nature. Haralick montre qu'il faut un modèle du troisième ordre pour obtenir une bonne estimation des dérivées premières :

$$P(x, y) = k_1 + k_2x + k_3y + k_4x^2 + k_5xy + k_6y^2 + k_7x^3 + k_8x^2y + k_9xy^2 + k_{10}y^3 \tag{5-139}$$

Les coefficients des polynômes sont obtenus par identification avec la décomposition de l'image suivant une base de polynômes orthogonaux limitée au degré 3 :

$$P_{i,j} = \hat{A}_{i,j} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} k_{nm} P_{nm} \quad i,j$$

En reprenant les expressions du paragraphe 1.1 :

$$\frac{\partial A}{\partial n} = \nabla A(x,y) \cdot \mathbf{n} = \frac{\partial A}{\partial x} \cos \phi + \frac{\partial A}{\partial y} \sin \phi \quad (5-140)$$

$$\frac{\partial^2 A}{\partial n^2} = \frac{\partial^2 A}{\partial x^2} \cdot \cos^2 \phi + 2 \frac{\partial^2 A}{\partial x \partial y} \cos \phi \sin \phi + \frac{\partial^2 A}{\partial y^2} \cdot \sin^2 \phi \quad (5-141)$$

Il est possible de calculer l'orientation du gradient au centre du voisinage :

$$\cos \phi = \frac{k_2}{\sqrt{k_2^2 + k_3^2}} \quad \sin \phi = \frac{k_3}{\sqrt{k_2^2 + k_3^2}} \quad (5-142)$$

En posant $x = \rho \cos \phi$ et $y = \rho \sin \phi$ la dérivée seconde directionnelle peut être mise sous la forme :

$$\frac{\partial^2 A}{\partial n^2} = a\rho + b \quad (5-143)$$

où a et b sont fonction des coefficients k_i et de l'orientation ϕ .

Un pixel $[i, j]$ est alors déclaré comme point de contour s'il existe ρ tel que :

$$|\rho| < \rho_0, \quad \frac{\partial^2 A}{\partial n^2} = 0, \quad \frac{\partial A}{\partial n} \neq 0 \quad (5-144)$$

Ceci indique qu'il existe un point de passage par zéro dans la direction du gradient et dans un voisinage de taille ρ_0 autour du pixel considéré.

3 Morphologie Mathématique

3.1 Gradient morphologique

Le gradient morphologique donne une approximation du module du vecteur gradient en chaque point. Pour une fonction f continûment différentiable (adoptée ici par souci de simplification des propriétés topologiques du gradient morphologique qui est, dans ce cas, une application continue) :

$$|\nabla f| = \lim_{\lambda \rightarrow 0} \frac{(f \oplus \overset{\vee}{g}_{\lambda K}) - (f \ominus \overset{\vee}{g}_{\lambda K})}{2\lambda} \quad (5-145)$$

où $\overset{\vee}{g}_{\lambda K}$ est l'élément structurant associé au disque fermé de rayon λ (*cf.* annexe C)

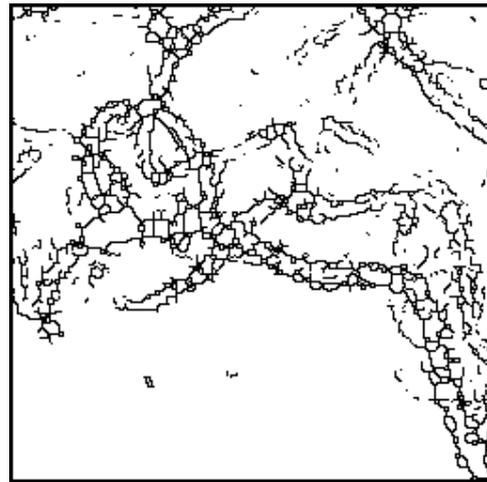
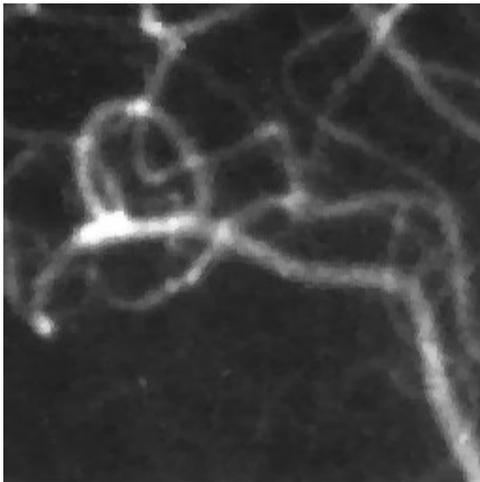
Dans le cadre discret, pour une image $A[s]$, la norme du gradient s'obtient par la transformation :

$$|\nabla A| = \frac{(A \oplus \overset{\vee}{g}_K) - (A \ominus \overset{\vee}{g}_K)}{2} \quad (5-146)$$

Il suffit donc de dilater et d'éroder l'image en niveaux de gris puisque le gradient s'interprète comme l'étendue, c'est-à-dire la différence entre le « max » et le « min » sur le disque unité.

Ce calcul de la norme du gradient est très sensible au bruit, il ne peut s'appliquer que sur une image lissée. De plus, les contours obtenus après un simple seuillage sont épais, il faut donc les amincir, par exemple par un amincissement morphologique homotopique, suivi éventuellement d'une ébarbulation.

La figure 5.12 présente un traitement morphologique complet : l'image a été lissée par un filtre alterné séquentiel (*cf.* chapitre IV) comportant une fermeture suivie d'une ouverture (avec le disque unité). Le gradient morphologique a ensuite été calculé en tout point puis l'image a été seuillée et amincie avec l'élément structurant L dans toutes les directions jusqu'à stabilisation.



*Image ANGIOGRAPHIE lissée par filtre
alterné séquentiel*

Gradient morphologique seuillé puis aminci

Figure 5.12 : Contours par l'opérateur gradient morphologique

3.2 Ligne de partage des eaux

Si l'on considère l'image comme une surface dans l'espace de dimension 3, on peut utiliser la terminologie de la géographie qui définit la ligne de partage des eaux comme la crête qui forme la limite entre deux bassins versants. Celle-ci s'obtient par amincissement morphologique effectué avec l'élément structurant L ⁴ (et ses rotations) itéré jusqu'à stabilisation. En effet, l'amincissement avec L, en

⁴L'élément structurant L est décrit dans l'annexe C

maille carrée, consiste à remplacer le niveau de gris de $A[i, j]$ par la valeur A_1 si $A_1 < A[i, j] \leq A_2$ sachant que :

$$A_1 = \max\{A[i-1, j-1], A[i-1, j], A[i-1, j+1]\}$$

$$A_2 = \min\{A[i+1, j-1], A[i+1, j], A[i+1, j+1]\}$$

L'amincissement effectué dans toutes les directions (8 en maille carrée) abaisse donc les niveaux de gris des pixels situés sur les flancs des pics, sans toucher aux sommets, ni aux vallées. En itérant jusqu'à stabilisation, les lignes de crêtes ont gardé leurs niveaux de gris d'origine et entourent des régions dont le niveau de gris est uniforme et égal au niveau de gris le plus bas de la région. Un simple seuillage suffit alors à extraire la ligne de partage des eaux.

Sur une image non texturée comportant des objets sombres sur un fond clair (ou vice-versa), la ligne de partage des eaux fournit un partitionnement en zones d'influence des objets, même si la frontière entre les objets est très faiblement marquée [COST89].

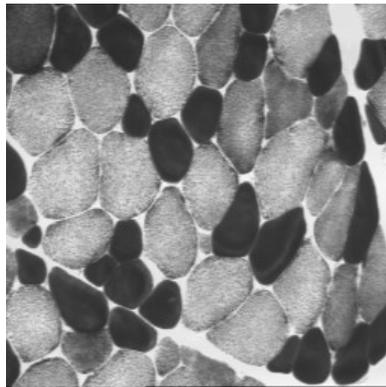
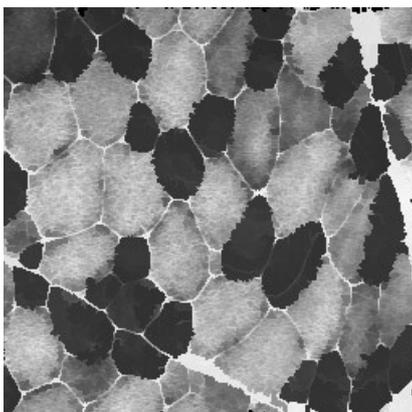


Image originale MUSCLE



Ligne de partage des eaux obtenue par amincissement



Fibres noires extraites par seuillage

Figure 5.13 : Utilisation de la ligne de partage des eaux pour extraire des frontières peu marquées entre zones homogènes

Ce traitement est particulièrement adapté pour séparer des objets très proches, pour lesquels la frontière ne se manifeste que par quelques niveaux de gris de différence par rapport aux objets (*cf.* figure 5.13). Sur l'image MUSCLE, les fibres noires voisines sont très difficiles à détacher, car les frontières qui les séparent sont à peine marquées dans l'image. Cette méthode permet d'extraire la plupart de ces régions.

Pour séparer des objets pour lesquels la frontière est légèrement plus foncée que les objets eux-mêmes, on opérera une inversion préalable des niveaux de gris de l'image (négatif).

4 Localisation des contours et seuillage

Nous avons exposé dans les paragraphes précédents différentes méthodes pour extraire les frontières entre régions. Pour certaines d'entre elles, les contours extraits peuvent être ni minces (d'épaisseur supérieure à 1 pixel) ni fermés. Les méthodes dérivatives, surfaciques et morphologiques calculent une approximation des dérivées première ou seconde qu'il faut ensuite seuiller pour obtenir les contours proprement dits.

Les nombreuses techniques de seuillage adaptatif, de suivi de contours, restent lourdes dans leur mise en œuvre sur des contours sur-segmentés difficilement exploitables.

Pourtant, quand un contour peut s'interpréter comme une ligne de crête dans l'image de la norme du gradient, il est caractérisé par :

- des niveaux de gris toujours élevés,
- de faibles dénivelés le long de ces lignes,
- de fort dénivelés dans les autres directions.

Les points de contour correspondent donc aux maximums locaux dans l'image de la norme du gradient.

4.1 Extraction des maximums locaux de la norme du gradient

L'algorithme d'extraction des maximums locaux consiste à comparer la norme du gradient au site $M[i, j]$ à celle de ses deux voisins M_1 et M_2 situés dans la direction du gradient à une distance unitaire de part et d'autre de M .

Soient A l'image de départ, A_i et A_j les images des composantes verticale et horizontale du gradient.

Soient Gr la norme du gradient du point courant $M[i, j]$, Gr_1 et Gr_2 les normes des gradients en M_1 et M_2 .

Le point M est un maximum local si : $Gr > Gr_1$ et $Gr > Gr_2$

Gr_1 et Gr_2 peuvent être calculés par interpolation linéaire. Posons $u_i = A_i[i, j]$ et $u_j = A_j[i, j]$. Suivant les valeurs relatives de u_i et u_j , on peut distinguer différents

cas de figure pour l'interpolation. Ainsi, dans le cas de la figure 5.14 où $u_i > u_j$, nous avons :

$$Gr_1 = \frac{u_j}{u_i} Gr_{i-1, j} + \frac{u_i - u_j}{u_i} Gr_{i-1, j+1} \quad (5-147)$$

$$Gr_2 = \frac{u_j}{u_i} Gr_{i+1, j-1} + \frac{u_i - u_j}{u_i} Gr_{i+1, j} \quad (5-148)$$

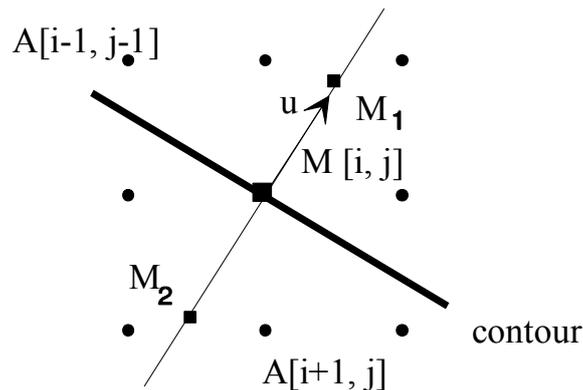


Figure 5.14 : Exemple d'interpolation

4.2 Seuillage par hystérésis

Les fluctuations provenant du bruit et apparaissant en sortie d'un opérateur de détection de contours font que, quand un seuil unique est utilisé sur l'image des maximums locaux pour obtenir les contours, il y a un risque de création de lacunes. On peut limiter ce défaut en employant un seuillage par *hystérésis* avec deux valeurs de seuil sh et sb ($sh > sb$). Les maximums locaux dont la valeur est supérieure à sh sont conservés, ainsi que tous ceux dont les valeurs sont supérieures à sb et qui appartiennent à une composante connexe comprenant au moins une valeur supérieure à sh . Le risque de coupure d'un contour est réduit car il correspondrait à une valeur de sortie de l'opérateur qui fluctuerait au dessus du seuil haut et en dessous du seuil bas [CANN86].

4.3 Suivi de ligne de crête

La morphologie mathématique propose des outils adaptés au suivi de ligne de crête dans une image de norme de gradient selon les critères définis au début du paragraphe 4 :

- faibles dénivelés le long de lignes de crête,
- forts dénivelés dans les autres directions.

L'idée est de déterminer des « marqueurs » du contour dans l'image de la norme de gradient (obtenue par une méthode quelconque) et à les propager dans cette image, conditionnellement à ces critères.

Les marqueurs (binaires) sont obtenus par extraction des maximums locaux et constituent donc les germes des contours.

La propagation s'effectue itérativement à partir de cette image \mathbf{M}_0 des marqueurs, en ajoutant à chaque étape k les pixels vérifiant le critère de dénivelé. Ceux-ci sont déterminés dans le voisinage V_k des marqueurs propagés à l'étape $k-1$ en ne conservant que ceux ayant un dénivelé faible. Plus précisément :

$$V_k = (\mathbf{M}_{k-1} \oplus \overset{\vee}{g}_C) \setminus \mathbf{M}_{k-1} \quad (5-149)$$

où $\overset{\vee}{g}_C$ est un élément structurant associé à un compact C (par exemple, un hexagone) et où « \setminus » représente la différence ensembliste.

Les valeurs des pixels de V_k sont comparées au maximum a_0 de l'image de départ ; on ne conserve que ceux pour lesquels la différence est faible, *i.e.* ceux retenus par un seuillage à la valeur du dénivelé moyen δ_k sur V_k .

Comme à chaque itération on ajoute des points, il est nécessaire de définir un critère d'arrêt. Le contour est obtenu lorsque, dans le voisinage V_k , il n'y a plus de direction de faible dénivelé. Le critère d'arrêt porte donc sur le dénivelé minimal d_k dans V_k : si sa valeur est supérieure à un pourcentage β de a_0 , la propagation s'arrête.

Soit \mathbf{A} l'image de la norme du gradient et \mathbf{D} l'image des dénivelés

$$a_0 := \max \{A[s]\}$$

\mathbf{M}_0 := image des marqueurs (maximums locaux seuillés à α % de a_0).

$$\forall s, D[s] := a_0 - A[s]$$

$$d_k := 0; k := 1;$$

TANT QUE $d_k < \beta \cdot a_0$ FAIRE

DEBUT

$$V_k := (\mathbf{M}_{k-1} \oplus \overset{\vee}{g}_C) \setminus \mathbf{M}_{k-1}$$

$$\delta_k := \text{dénivelé moyen sur } V_k$$

$$d_k := \text{dénivelé minimal sur } V_k$$

$$\mathbf{M}_k := \mathbf{M}_{k-1} \cup \{s \in V_k, D[s] < \delta_k\}$$

$$k := k+1$$

FIN

Algorithme 5.1 : Suivi de ligne de crête

L'algorithme 5.1 dépend de 3 paramètres. Beaucoup de pixels, bien qu'ayant une norme du gradient élevée n'appartiennent pas au contour recherché ; le premier paramètre, α , en élimine le plus grand nombre par un simple seuillage. L'ajustement fin de α rend la procédure de propagation d'autant plus rapide.

Le deuxième paramètre est l'élément structurant $\overset{\vee}{g}_C$ qui définit la taille du voisinage dans lequel les critères de dénivelés sont calculés. Son ajustement permet, d'une part de moduler le degré de contrainte sur les dénivelés et, d'autre part d'accélérer la convergence. Toutefois, plus $\overset{\vee}{g}_C$ est grand, moins il y a d'itérations mais plus il y a risque de sur-segmentation.

Enfin le paramètre β traduit les variations maximales de la norme du gradient autorisées. Une valeur élevée de ce paramètre entraîne la formation de boucles parasites difficiles à éliminer par la suite.

5 Conclusion

Ce chapitre présente de manière approfondie les opérateurs dérivatifs. Les critères continus et surtout discrets de comparaison sont utiles au traicteur d'images pour faire un choix en fonction de la propriété (rapport signal sur bruit, délocalisation, unicité de la réponse) qu'il considère comme la plus importante d'exploiter dans son application. Il est intéressant de noter que l'approche surfacique (celle d'Haralick) débouche sur des opérateurs de type filtre à réponse impulsionnelle finie semblables aux opérateurs de Prewitt ou à l'opérateur MDIF. L'approche morphologique est adaptée à la prise en compte de critères topologiques pour définir les frontières de régions dans une image. Enfin, ces approches sont peu adaptées aux images texturées pour lesquelles il est préférable d'exploiter les approches régions ou les approches markoviennes.

Ce chapitre, auquel se rattachent les annexes D, E et F, est complété par le chapitre IX relatif à une comparaison des résultats obtenus sur des images de synthèse et des images issues de la banque du GDR 134

Fermeture de contours, contours virtuels

Les opérateurs de détection de contours comportent souvent un seuil dont le réglage résulte d'un compromis pour extraire au mieux le contour « vrai » du bruit. Quelle que soit la technique employée, les contours obtenus présentent des lacunes ayant pour origine la présence de bruit mais aussi des occultations. La fermeture de ces lacunes revêt un grand intérêt pour la segmentation d'image car elle permet d'obtenir des régions fermées pouvant être interprétées comme des projections d'objets ou parties d'objets à reconnaître dans la scène. Plusieurs approches sont possibles, on peut distinguer :

- les méthodes exploitant les techniques d'exploration de graphe,
- les méthodes d'inspiration neurobiologique,
- les méthodes exploitant les contours déformables.

La première classe de méthodes s'inspire des techniques d'intelligence artificielle et de programmation dynamique. Elle sera présentée au paragraphe 1.

Nous abordons dans le paragraphe 2 les méthodes d'inspiration biologique. A cette occasion nous détaillons la modélisation des mécanismes de la vision préattentive chez les vertébrés. L'intérêt de l'approche neuromimétique est qu'elle est globale et qu'elle permet de simuler des systèmes complets de vision allant de l'extraction de primitives à l'interprétation dans lesquels coopèrent les processus de bas et de haut-niveau [GAUS91].

Enfin, les « snakes » à extrémités fixes peuvent être employés pour combler des lacunes. Dans ce cadre, l'opérateur doit choisir les deux extrémités. Cette approche est exposée dans le chapitre VIII.

Nous présentons en fin de ce chapitre, un des mécanismes d'extraction de points caractéristiques qui explique certains phénomènes d'illusions visuelles.

1 Fermeture des contours

1.1 Recherche du meilleur chemin entre 2 extrémités

Comme nous l'avons exposé dans le chapitre II, l'image numérique peut être associée à un maillage où chaque sommet est un site ou pixel. Le maillage est un graphe, la fermeture de contour peut donc s'apparenter à la recherche d'un chemin entre les deux sommets du graphe associés aux deux sites s_0 et s_f qui sont les

extrémités de la lacune à combler (cf. figure 6.1). Les algorithmes d'énumération tels que A* [NILS71] sont applicables. Pour tout site s situé sur un chemin solution, on définit une fonction d'évaluation :

$$\hat{f}(s) = \tilde{g}(s) + \hat{h}(s) \quad (6-1)$$

où :

- $\tilde{g}(s)$ représente le coût du chemin de s_0 à s
- $\hat{h}(s)$ est une estimation du coût du chemin de s à s_f

Pour le chemin solution de coût minimum, on a la relation :

$$f(s) = g(s) + h(s) \quad (6-2)$$

L'algorithme A* permet de trouver le chemin de coût minimum si :

$$\hat{h}(s) \leq h(s)$$

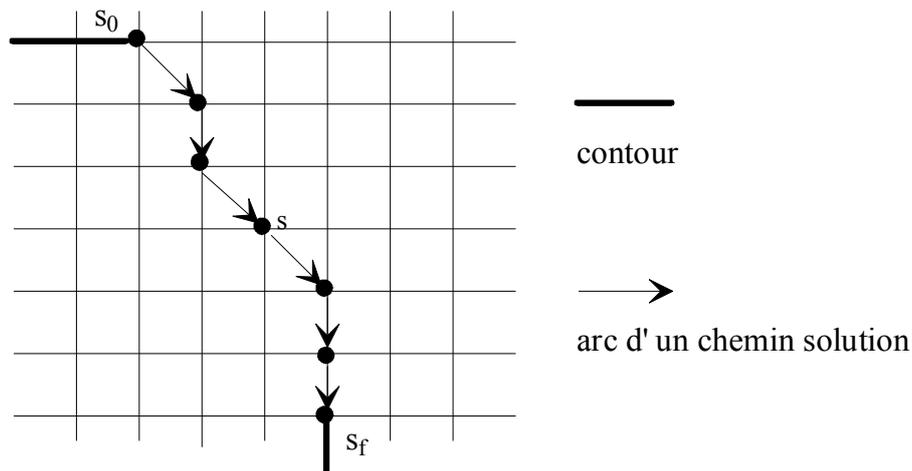


Figure 6.1 : Recherche de chemin sur un maillage

Cet algorithme n'est applicable que si l'on dispose de deux points extrêmes et si il existe des données dans l'image permettant de calculer une fonction d'évaluation. Voici quelques grandeurs pouvant intervenir dans le calcul de $\hat{h}(s)$:

- *norme du gradient* : si \mathbf{B} est l'image sur laquelle on applique l'algorithme, la contribution de l'ajout d'un site s au chemin en cours d'élaboration peut être mesurée par :

$$\max_{t \in \mathbf{B}} \|\nabla A(t) - \nabla A(s)\| \quad (6-3)$$

- *courbure* : si t est le prédécesseur de s dans le chemin en cours d'élaboration et si l'on recherche des trajets de faible courbure, la contribution de s est mesurable par une fonction monotone décroissante de $|\theta(s) - \theta(t)|$ où $\theta(s)$ est l'orientation du gradient en s .

- *distance à l'objectif*: si le chemin est de faible courbure, il est possible de prendre en compte la distance de s à l'objectif dans le calcul de $\hat{h}(s)$.

1.2 Recherche du meilleur chemin à partir d'une extrémité

Ce principe est exposé dans [DERI88], il suppose une extraction préalable des contours par un opérateur donnant des contours d'une « largeur » d'un pixel et fournissant la norme du gradient en chaque pixel. L'idée de base est d'extraire les contours « sûrs » en fixant un seuil suffisamment élevé pour ne pas détecter le bruit. En procédant ainsi, les contours présentent souvent des lacunes. A partir de chaque extrémité de contour, il y a élaboration d'un chemin qui suit la meilleure ligne de crête dans l'image de la norme du gradient.

1.2.1 Identification des extrémités

Comme les contours sont fins, il est possible d'identifier une extrémité par l'étude du voisinage 3x3 de chaque point de contour dont les huit voisins x_i sont désignés suivant l'ordre indiqué dans la figure 6.2. Si le pixel voisin est un point de contour, x_i vaut 1, il vaut 0 sinon. Ainsi, chaque voisinage peut être codé sur un octet :

$$V = \sum_{i=1}^7 x_i 2^i \quad (6-4)$$

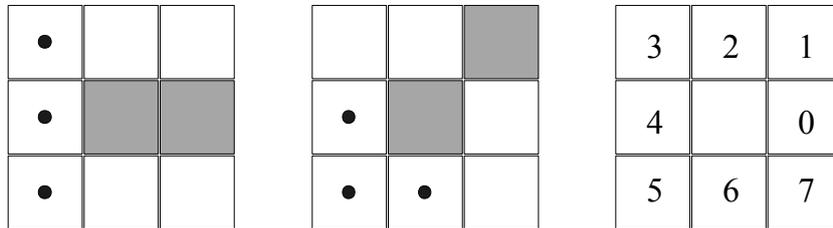


Figure 6.2 : Exemples de configurations extrémités :
 les pixels hachurés représentent les contours,
 les points noirs désignent les candidats à la fermeture
 Le carré de droite indique la numérotation des voisins pour le calcul du code V

On construit alors une table T de 256 éléments (« look-up table ») dont la fonction d'adressage est le code V du voisinage. Si le pixel examiné $s = [i, j]$ est une extrémité, l'élément $T[V]$ adressé par le code V du voisinage contiendra un 1 et les coordonnées relativement à s des 3 pixels candidats à la fermeture (cf. figure 6.3).

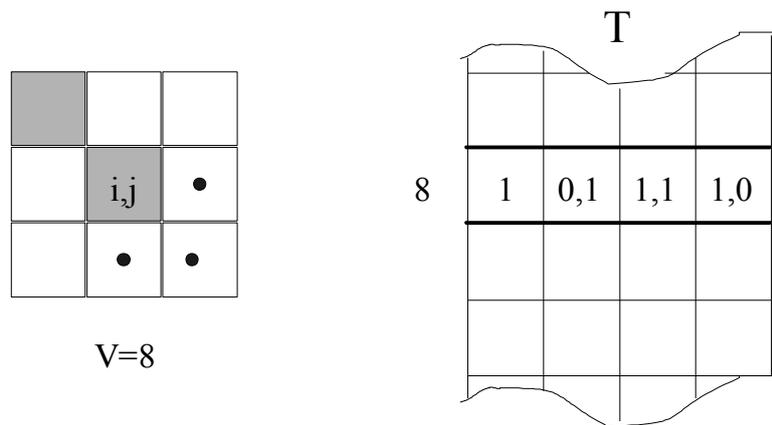
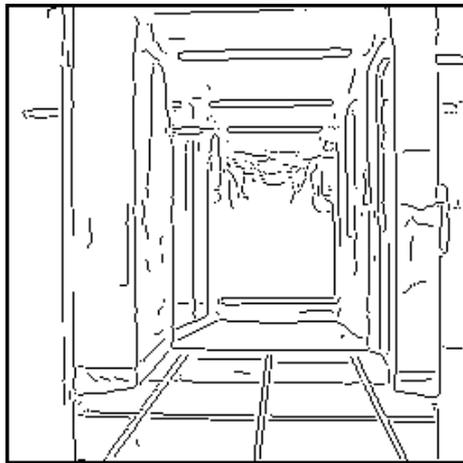
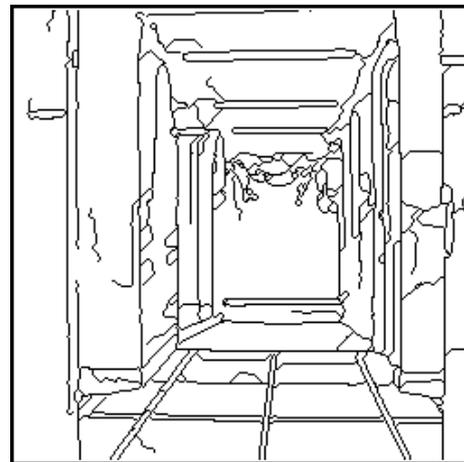


Figure 6.3 : Codage d'une extrémité et table d'examen contenant les coordonnées relatives des candidats à la fermeture



Contours obtenus avec l'opérateur de Deriche, $\alpha = 1$



Contours prolongés le long des lignes de crête du gradient ; $\text{prof}=3$, $\text{longmax}=7$

Figure 6.4 : Fermeture de contour à partir de l'image des gradients calculés par Deriche

1.2.2 Algorithme de fermeture

Pour mettre en œuvre cette méthode, l'image des contours est balayée ligne par ligne du haut vers le bas. Quand une extrémité est rencontrée, la procédure de recherche du meilleur candidat à la fermeture est activée. Quand la condition d'arrêt est atteinte, le balayage de l'image reprend là où il s'était arrêté.

Pour trouver le meilleur candidat à la fermeture, à partir de l'extrémité détectée, la procédure « recherche » de l'algorithme 6.1 élabore l'arborescence de tous les chemins possibles de profondeur (prof) donnée et attribue à chacun de ces chemins un coût égal à la somme des normes du gradient des points constituant le chemin.

Le contour est alors prolongé par le premier point du chemin ayant le meilleur coût. L'algorithme est ensuite itéré à partir de ce nouveau point. L'algorithme s'arrête soit quand un point de contour est rencontré, soit au bout d'un nombre d'itérations fixé (longmax).

Signalons que deux mises en œuvre de la fermeture de contour, dont l'une utilise des « automates cellulaires » sont proposées dans [DEMI91] et [QUES91].

```

POUR i := 1 A M FAIRE
  POUR j := 1 A N FAIRE
    SI (A[i, j] est un contour) ALORS
      DEBUT
        V := codage(A[i,j]);
        SI (T[V][1] = 1) ALORS
          DEBUT
            continu := 0;
            k := longmax;
            TANT QUE (( k > 0 ) ET (continu = 1)) FAIRE
              DEBUT
                k--
                /* recherche des coordonnées [m, n] du meilleur candidat à la
fermeture */
                [m, n] := recherche( i, j, V, prof);
                SI (A[m, n] = contour) ALORS
                  continu := 0;
                  SINON
                    DEBUT
                      A[m, n] est marqué comme
contour;
                      V := codage(A[m, n]);
                    FIN
                  FIN
                FIN
              FIN
            FIN
          FIN
        FIN
      FIN
    FIN
  FIN
FIN

```

Algorithme 6.1 : Fermeture de contour

2 Approche neuromimétique de la vision

Le but de ce paragraphe est de proposer une simulation de certaines propriétés de la vision naturelle (humaine et animale). Les recherches dans ce domaine utilisent des résultats obtenus aussi bien en psychologie qu'en neurobiologie. Les variations d'intensité lumineuse sont des caractéristiques importantes exploitées par notre système visuel. Elles correspondent souvent à des contours d'objets. Or, les premières étapes du traitement visuel, qui sont rapides et quasi automatiques, sont très difficiles à caractériser car les contours que nous percevons ne sont pas forcément corrélés avec les données optiques.

Il est difficile d'expliquer notre perception continue des contours et de l'ensemble du monde extérieur car la rétine est parcourue par des veines qui ont pour effet d'altérer l'image reçue. Grossberg et Mingolla suggèrent donc l'existence de « contours virtuels » qui ont un effet important dans les processus de reconnaissance en réglant partiellement le problème de la fermeture des contours [GROS85a], [BONN91]. La rapidité de l'extraction de contours exclut toute analyse complexe de la scène. Il s'agit donc d'un mécanisme *préattentif* opérant localement et de manière massivement parallèle. La possibilité d'améliorer le contraste d'une image et de rendre l'analyse indépendante des variations locales de luminosité (partie d'une image sur ou sous exposée) semble aussi être liée aux toutes premières couches de traitement de l'image. Skrzypek a ainsi implémenté un réseau du même type que celui de Grossberg afin d'améliorer la qualité des images avant de leur faire subir d'autres traitements [SKRZ90].

Nous exposons donc le modèle de la vision préattentive de Grossberg et la réalisation d'un système de vision bas niveau à base de réseaux de neurones comprenant l'extraction de contours fins et la fermeture des contours. Nous présentons aussi la réalisation d'un mécanisme d'extraction de points caractéristiques dans une image de contours.

Signalons aussi les travaux de Von der Heigh et Heitger [VOND84] qui proposent un modèle sans rétro-action pour la fermeture des contours à partir de points caractéristiques.

2.1 Modèles de réseaux de neurones pour la vision de bas-niveau

2.1.1 Système d'analyse d'images par réseau de neurones

L'approche neuronale de la vision présente des similarités avec le traitement des images « classique » : elle comporte des aspects bas-niveau et haut-niveau. Cependant, la différence fondamentale vient du fait que structurellement les différentes couches de neurones communiquent. Ainsi, il ne faut pas chercher à comparer directement les systèmes neuronaux d'extraction de primitives à certains opérateurs du traitement d'images car l'extraction de caractéristiques n'est pas une fin en soi dans l'approche connexionniste. Elle n'est qu'un traitement intervenant à un niveau donné et intégré dans une structure globale d'interprétation où la remise en cause est possible à chaque étape.

Grossberg propose un modèle général pour l'interprétation d'images. Il distingue d'abord un premier niveau de prétraitement des données lié aux neurones photosensibles. A partir de ces données, l'extracteur de contours (BCS : *Boundary Contour System*) fabrique une image de contours qui est exploitée par l'extracteur de primitives (FCS : *Feature Contour System*). Ces données de bas-niveau sont ensuite utilisées par la partie interprétation (ORS : *Objet Recognition System*) (cf. figure 6.5).

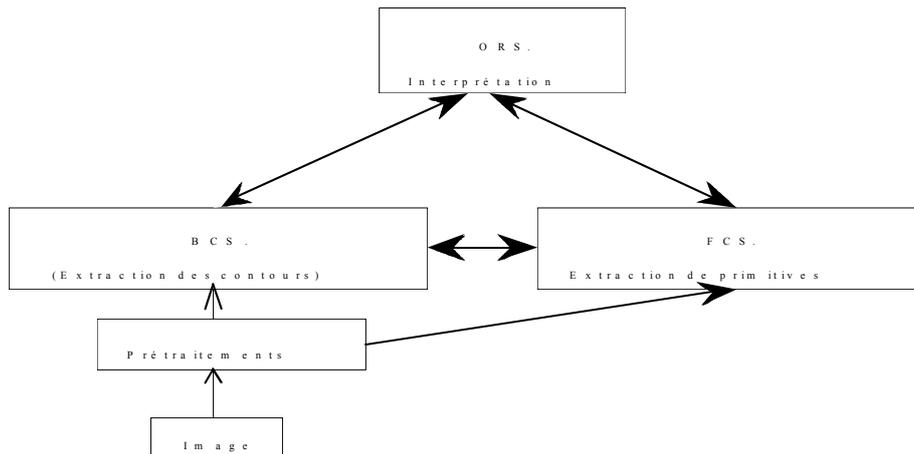


Figure 6.5 : Organisation du système complet

Le système BCS comprend deux blocs de traitements. Le premier, composé de plusieurs couches, réalise une compétition entre différents contours possibles dans un voisinage donné alors que le second fait coopérer les résultats fournis par le premier bloc afin de prolonger les contours (cf. figure 6.7). Les résultats obtenus sont alors injectés vers le premier bloc au travers d'une boucle de rétroaction pour qu'ils soient pris en compte au même titre que des contours vrais (cf. utilisation pour la segmentation de nuages [LEHA90]).

Des règles de fermeture des contours et un principe d'extraction de contours « fins » par compétition ont été imaginés pour expliquer des illusions visuelles célèbres [GROS85a], [GROS85b], [GROS87], [BONN91] (cf. figure 6.6).

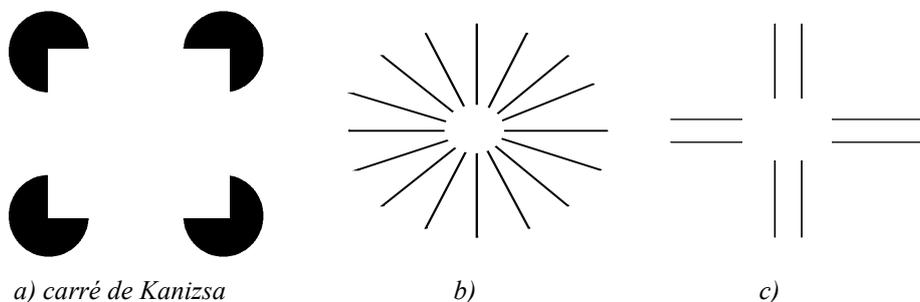


Figure 6.6 : Exemples d'illusions visuelles

Notre système visuel prolonge les contours dans la direction perpendiculaire aux fins de lignes des contours réels. Ainsi, sur la figure 6.6b nous devinons un cercle et sur la figure 6.6c nous percevons un rectangle au centre de la figure. Les deux étages de compétition que nous allons proposer créent ces contours illusoires. En revanche, dans la figure 6.6a les lignes de contours sont prolongées en ligne droite. Il y aura donc une compétition entre les deux possibilités de fermeture des contours : soit dans le prolongement, soit perpendiculairement. La direction choisie pour prolonger les contours ne sera pas locale mais globale (localement les deux hypothèses sont formulées, mais se sont les interactions avec le reste de l'image qui renforceront l'une des deux possibilités). C'est donc à un niveau supérieur que la décision sera prise. La figure 6.7 représente le schéma général du système d'extraction de contour avec les différentes couches correspondant à des niveaux de compétition qui sont exposés dans les paragraphes suivants.

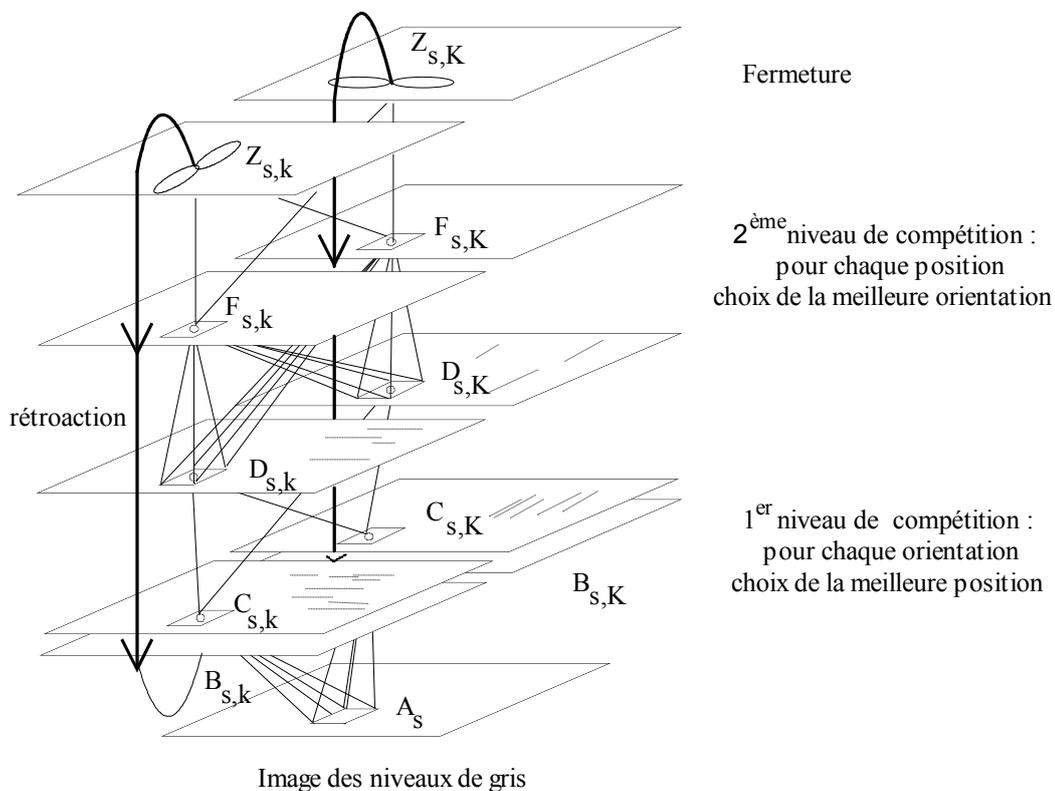


Figure 6.7 : Les différentes couches du système BCS

2.1.2 Masques orientés pour l'extraction de contours

La première couche de neurones correspond à l'image de départ **A** (cf. figure 6.7). La deuxième couche est assimilable à une image **B** de même taille que **A**, dont chaque site de coordonnées $s = [i, j]$ comporte un neurone prenant en compte les sites de **A** situés dans une fenêtre **F** centrée en s et d'orientation k (cf. figure 6.9). Les éléments d'entrée des neurones de cette deuxième couche

d'extraction de contours sont donc des masques directionnels simplifiés. On définit par $\mathbf{Ag}_{s,k}$ et $\mathbf{Ad}_{s,k}$ les parties gauche et droite de la fenêtre \mathbf{F} ; $\mathbf{Ag}_{s,k}$ et $\mathbf{Ad}_{s,k}$ représentent la somme des activités des neurones situés dans chaque partie de la fenêtre.

$$\mathbf{Ag}_{s,k} = \sum_{s \in \mathbf{Ag}_{s,k}} A_s \quad \text{et} \quad \mathbf{Ad}_{s,k} = \sum_{s \in \mathbf{Ad}_{s,k}} A_s \quad (6-5)$$

La sortie de chaque neurone de cette deuxième couche est définie par :

$$B_{s,k} = [\mathbf{Ag}_{s,k} - \alpha \mathbf{Ad}_{s,k}]^+ + [\mathbf{Ad}_{s,k} - \alpha \mathbf{Ag}_{s,k}]^+ \quad (6-6)$$

avec : $[w]^+ = \max\{w, 0\}$

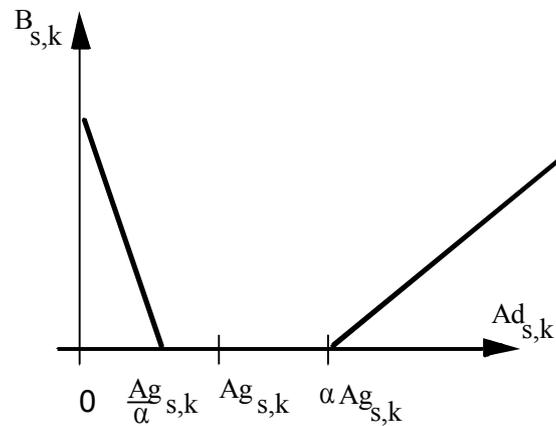


Figure 6.8 : Représentation de $B_{s,k} = g(\mathbf{Ag}_{s,k}, \mathbf{Ad}_{s,k})$

La valeur de α joue le rôle d'un seuil. Si α est grand, on réalise un seuillage important qui va se traduire par une perte d'information définitive (cf. figure 6.8). Si $\alpha = 1$ alors l'expression (6-6) se réduit à :

$$B_{s,k} = |\mathbf{Ag}_{s,k} - \mathbf{Ad}_{s,k}| \quad (6-7)$$

En général, la sortie d'un extracteur de contours est de la forme $|\mathbf{Ag}_{s,k} - \mathbf{Ad}_{s,k}| > \text{seuil}$. Ici, le paramètre α permet d'effectuer un seuillage sur le rapport entre $\mathbf{Ag}_{s,k}$ et $\mathbf{Ad}_{s,k}$, le comportement est donc asymétrique.

Les masques orientés permettent d'initialiser le processus de formation des contours. Ces opérateurs sont plutôt des détecteurs de variations de contraste dont des implantations ont été proposées par [LEHA90]. Nous présentons la méthode de Grossberg en nous limitant à 4 directions : horizontale, verticale et les deux diagonales. Les 4 masques sont décrits dans la figure 6.9.

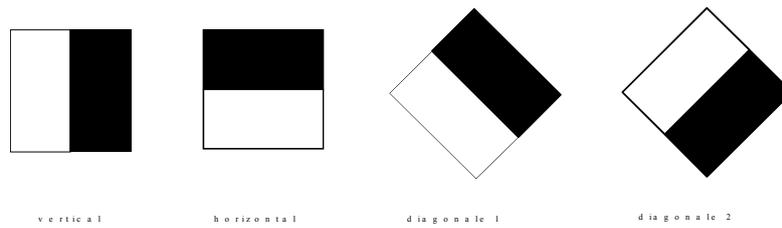


Figure 6.9 : Masques orientés

Les dimensions de la fenêtre sont paramétrables. Plus la fenêtre est grande, plus le moyennage est important (*cf.* chapitre IV). Lorsque la distance entre deux contours est inférieure à la taille du masque, les étapes de compétitions qui suivent entraîneront le réseau à ne garder que le contour correspondant au contraste le plus fort.

2.1.3 Premier niveau de compétition

Le principe de compétition est illustré avec des masques verticaux et horizontaux. Il en est de même pour les masques en diagonale. Cette première compétition concerne les neurones représentant une même orientation. Dans un voisinage donné, il ne doit rester qu'un neurone actif lié à une direction donnée, les autres doivent être inhibés. Il s'agit donc d'une compétition du type *On center-Off surround* [GROS82] entre un neurone et ses voisins. Chaque neurone $C_{s,k}$ excite un neurone de la couche supérieure et cherche à inhiber les neurones $C_{s',k}$ de sa couche appartenant à un voisinage donné v_s ($s \neq v_s$) (*cf.* figure 6.10).

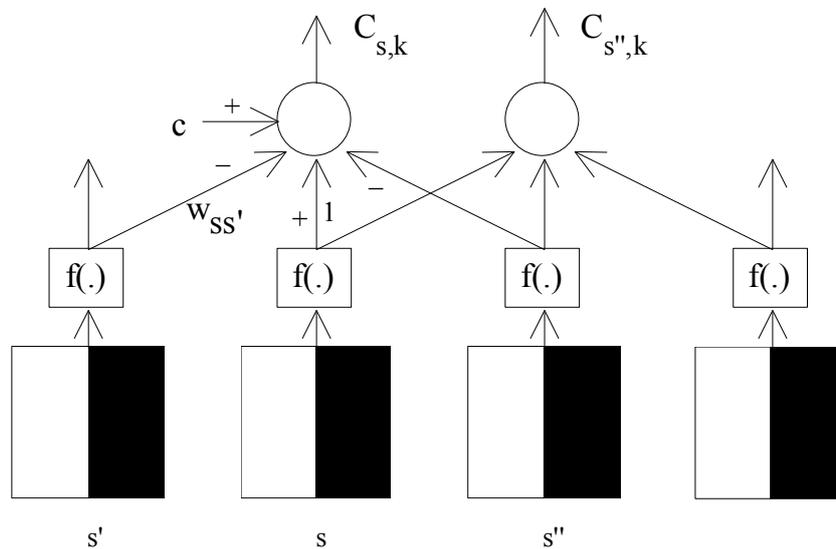


Figure 6.10 : Compétition de premier niveau pour contours verticaux
ici, s' et $s'' \in v_s$
(pour simplifier le dessin, tous les neurones C ne sont pas représentés)

L'équation différentielle décrivant le fonctionnement des neurones de la couche **C** s'écrit, pour chaque direction k , de la manière suivante (pour simplifier, nous employons la même notation pour désigner un neurone et son activité) :

$$\frac{d}{dt} C_{s,k} = -C_{s,k} + c + f(B_{s,k}) - C_{s,k} \cdot \sum_{s' \in \nu_s} f(B_{s',k}) \cdot w_{s's} \quad (6-8)$$

où :

- $w_{s's}$ est le poids de l'inhibition entre les neurones s et s' ,
- $f(B_{s,k})$ est la fonction de sortie des neurones de la couche **B**, elle est choisie égale à $\gamma \cdot B_{s,k}$ avec $\gamma > 0$,
- c est une entrée constante qui permet la suppression de l'inhibition :
 - pour amorcer des contours virtuels dans les mécanismes de compétition entre orientations
 - pour la fermeture de contour lors de la rétroaction.
- $c + f(B_{s,k})$ est un terme d'excitation,
- $C_{s,k} \cdot \sum_{s' \in \nu_s} f(B_{s',k}) \cdot w_{s's}$ est un terme d'inhibition.

Il a été démontré qu'un réseau décrit par cette équation converge vers un état d'équilibre [GROS85b]. La sortie des neurones correspond alors à une normalisation. Les sorties $C_{s,k}$ tendent vers la valeur :

$$c_{s,k} = \frac{c + \gamma \cdot b_{s,k}}{1 + \gamma \cdot \sum_{s' \in \nu_s} b_{s',k} \cdot w_{s's}} \quad (6-9)$$

$$\text{On peut prendre : } w_{s's} = \begin{cases} w_0 s' \text{ si } s' \in \nu_s \\ 0 \text{ sinon} \end{cases}$$

Si la taille des masques est supérieure à celle du voisinage de compétition alors il n'est pas possible de localiser les zones de transitions. Il faut donc que le voisinage de compétition soit plus grand que celui des masques orientés. Si les variations de niveaux de gris sur l'image d'entrée sont définies par une fonction rampe (*cf.* figure 6.11), seules les variations de pentes seront perçues et localisées correctement si elles sont éloignées d'une distance supérieure à la taille des masques. Le comportement est le même que celui des opérateurs à moyenne mobile utilisés en prétraitement (*cf.* chapitre IV). La première compétition fournit des contours fins directement à partir d'une image de type « gradient ».

L'équation (6-9) montre qu'à l'équilibre il y a un effet de normalisation. Cette dernière est nécessaire pour éviter, d'une part une croissance démesurée qui perturberait la compétition et d'autre part une diminution excessive du niveau de

sortie qui conduirait à la disparition de la totalité des contours. La sortie est nulle si elle est inférieure à un seuil.

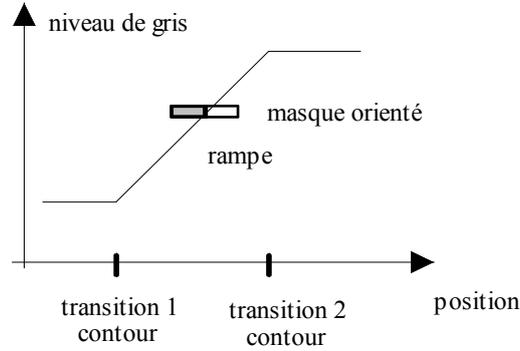


Figure 6.11 : Contours détectés sur une fonction « rampe » si la taille du masque est inférieure à la largeur de la transition

En résumé, dans un voisinage de compétition donné ayant les mêmes dimensions que la fenêtre, seuls les contours verticaux correspondant au plus fort « saut d'amplitude » et pris sur la même horizontale sont conservés (*vice-versa* pour les contours horizontaux). L'implémentation de cette formule sera donnée dans le paragraphe 2.2.1.

2.1.4 Deuxième niveau de compétition

Il s'agit d'une compétition entre les neurones associés aux différentes orientations possibles d'un pixel de contour donné (*cf.* figure 6.12). Si un neurone attaché à une direction donnée est inhibé, le neurone de la direction perpendiculaire pourra alors être activé faiblement (ce mécanisme est utile pour la fermeture des contours).

On notera k une direction donnée et K sa direction perpendiculaire. La sortie du neurone avant normalisation est donnée par :

$$D_{s,k} = d \cdot C_{s,k} - C_{s,K} \quad \text{et} \quad D_{s,K} = d \cdot C_{s,K} - C_{s,k} \quad (6-10)$$

où d est une constante positive.

Pour la direction donnée k , chaque sortie $D_{s,k}$ agit par une liaison excitatrice sur le neurone $F_{s,k}$ de la couche suivante et par une liaison inhibitrice sur chaque neurone $F_{s,k'}$ avec $k' \neq k$. On obtient ainsi un fonctionnement de type *On center-Off surround* décrit par l'équation différentielle :

$$\frac{d}{dt} F_{s,k} = -\chi F_{s,k} + (\varphi - F_{s,k}) D_{s,k} - F_{s,k} \sum_{k' \neq k} D_{s,k'} \quad (6-11)$$

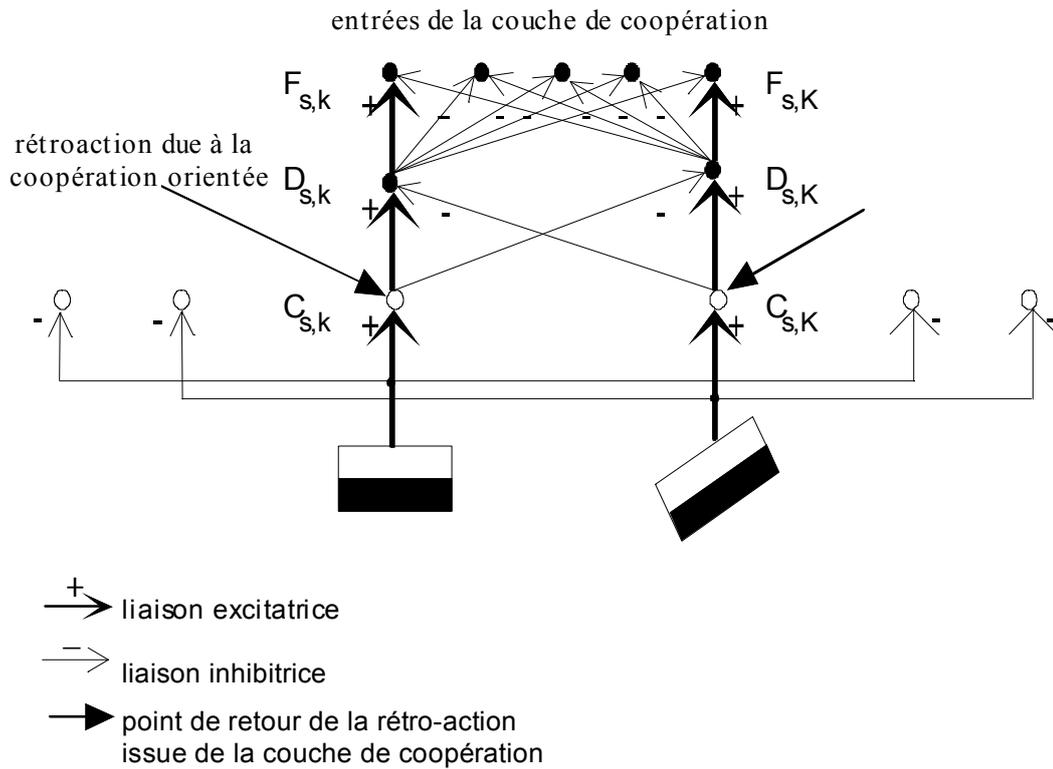


Figure 6.12 : Interaction entre les compétitions. Seule la compétition entre 2 orientations k et K est représentée. La partie non dessinée correspond à la compétition avec le voisinage.

L'équation (6-11) revient à ne laisser actif que le neurone attaché à une orientation donnée ayant un niveau de sortie maximum ; ceci pour chaque site de l'image. A l'équilibre, on a $\frac{dF_{s,k}}{dt} = 0$, il s'ensuit que l'activité d'un neurone est donnée par :

$$f_{s,k} = \frac{\varphi \cdot d_{s,k}}{\chi + \sum_{k'} d_{s,k'}} \quad (6-12)$$

Si l'on choisit χ négligeable par rapport à $\sum_{k'} d_{s,k'}$, (par exemple $\chi = 0.01$)

alors :

$$\sum_{k'} f_{s,k'} \approx \varphi \quad (6-13)$$

On obtient donc, en sortie, une opération équivalente à une normalisation. Dans le cas où l'on utilise uniquement deux orientations perpendiculaires k et K alors l'équation (6-11) correspond à une compétition entre les neurones correspondant à k et K . Donc si $F_{s,k}$ est excité alors $F_{s,K}$ est bien inhibé (cf. figure 6.13).

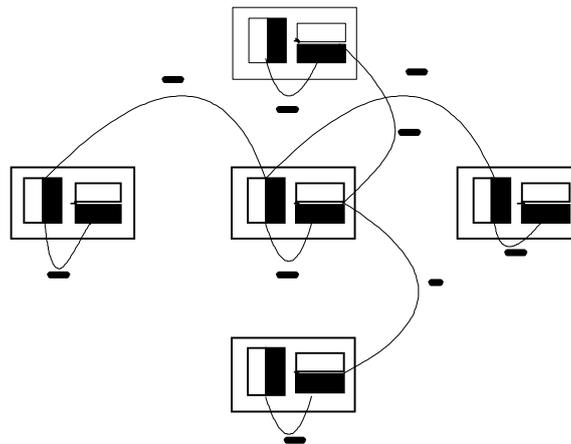


Figure 6.13 : Schéma général de la compétition entre voisins et entre différentes directions pour un même point

La compétition entre orientations étant décrite, nous pouvons revenir sur le mécanisme décrit par l'équation (6-8) qui peut être « câblé » suivant le schéma de la figure 6.14.

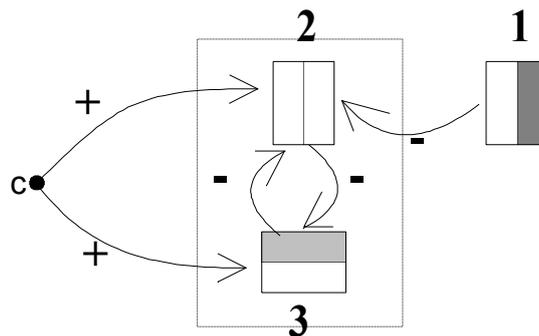


Figure 6.14 : Influence de l'entrée désinhibitrice sur la compétition entre orientations et positions

Lorsqu'une direction est désactivée par l'action d'un neurone correspondant à un contour voisin ou à cause de la rétroaction, la direction perpendiculaire pour le même pixel est activée grâce à l'introduction de l'entrée « désinhibitrice » c . Cette entrée constante c est nécessaire car, sans elle, il n'y aurait aucun moyen de changer l'état d'un neurone inactif ou faiblement actif. Cette situation est illustrée dans la figure 6.14 qui correspond à une situation initiale où le neurone 1 est activé et où les neurones 2 et 3 sont très faiblement activés. Il s'ensuit que le neurone 1 entraîne une inhibition importante du neurone 2 qui lui-même autorise une activation du neurone 3, cette dernière est rendue possible grâce à l'entrée c . Le neurone 3 devient alors plus actif qu'au départ ; il révèle un contour virtuel induit par l'activité importante du neurone 1.

2.1.5 Coopération orientée

La mise en place de pixels de fermeture s'effectue grâce à une boucle de rétroaction. Elle utilise les contours fins produits par le deuxième niveau de compétition.

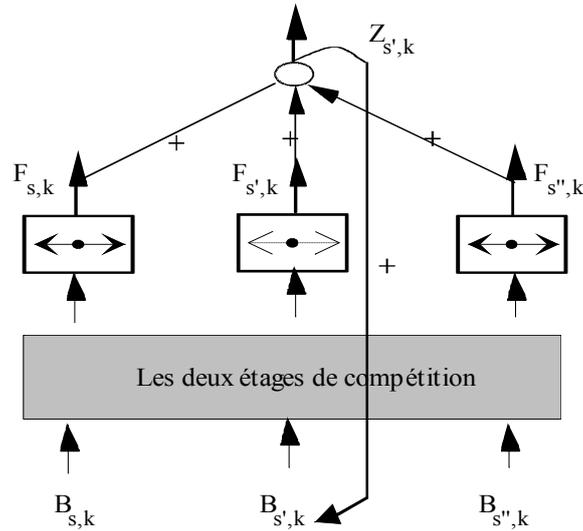


Figure 6.15 : Boucle de coopération, schéma de principe

Les neurones de la couche Z enclenchent (cf. figure 6.15) un mécanisme de fermeture de la manière suivante : un neurone sera actif si et seulement si il reçoit suffisamment d'excitations positives d'entrées alignées appartenant au deuxième étage de compétition. Une lacune pourra être comblée si, dans une direction donnée, il existe des portions de contours qui pourraient se raccorder en formant une droite et si cette droite n'est pas coupée par des neurones actifs dans une direction perpendiculaire. Cela peut être traduit par l'équation différentielle :

$$\frac{dZ_{s,k}}{dt} = -Z_{s,k} + g \left(\sum_{s' \in \mathcal{V}_s} (f(F_{s',k}) - f(F_{s',K})) h(s,k) \right) \quad (6-14)$$

- k est la direction associée au neurone $Z_{s,k}$ sur la couche de coopération,
- K est la direction perpendiculaire à k ,
- g est une fonction de seuillage,

$h(s,k)$ est la fonction donnant les poids des liaisons entre le neurone lié au pixel candidat à la fermeture et les neurones de son voisinage \mathcal{V}_s dans la deuxième couche de compétition,

- f est la fonction de sortie des neurones du deuxième étage de compétition.

Dans l'équation (6-14), $(f(F_{s',k}) - f(F_{s',K}))$ représente l'utilisation antagoniste de deux directions perpendiculaires. La sortie du neurone $Z_{s,k}$ sera d'autant plus grande que l'activité des neurones $F_{s',k}$ sera grande et que celle des neurones $F_{s',K}$ sera petite (cf. figure 6.16). Ces influences sont modulées par la fonction $h(s,k)$ dont une forme possible (cf. figure 6.16) est :

$$h(s,k) = \exp\left(-2\left(\frac{\rho}{s} - 1\right)^2\right) \cdot \cos^p(\theta) \quad (6-15)$$

avec $\rho = (x^2 + y^2)^{\frac{1}{2}}$ et $\theta = \arctan\left(\frac{y}{x}\right)$

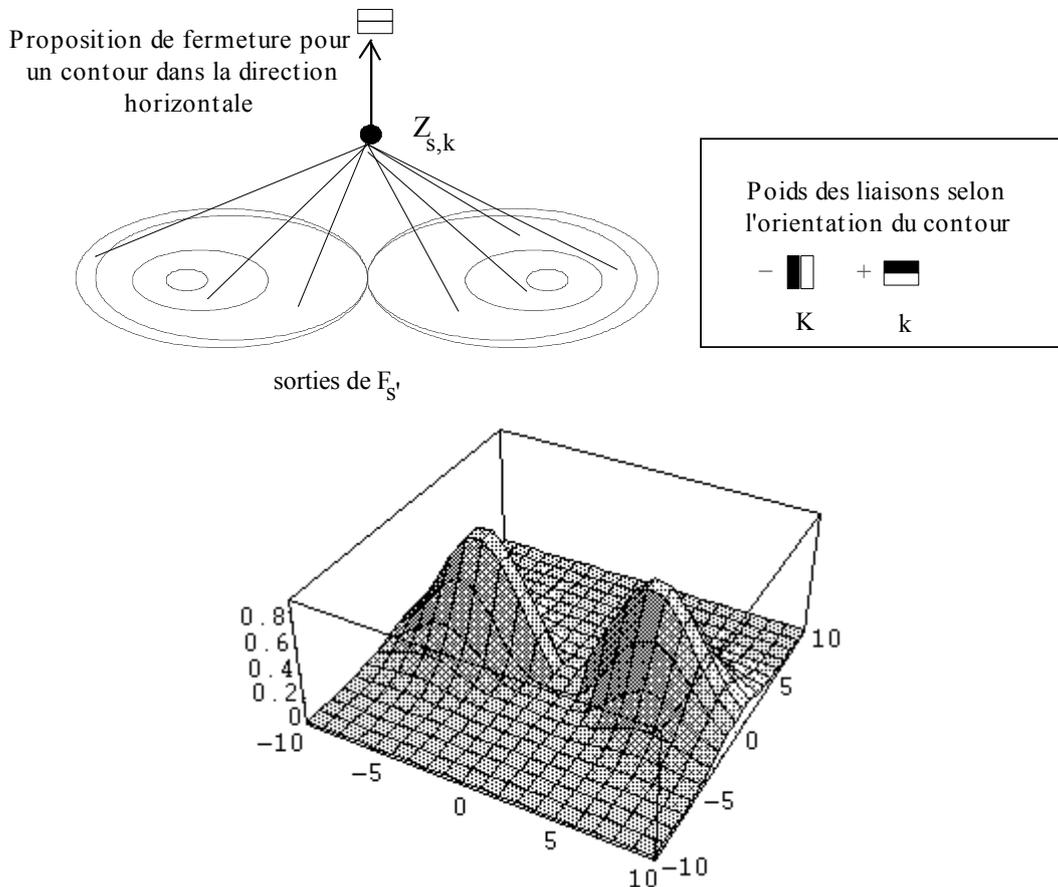


Figure 6.16 : Liaisons synaptiques d'un neurone de fermeture et graphe de la fonction $h(s,k)$ de pondération

2.1.6 Rétroaction

Les sorties des neurones de fermeture sont renvoyées en entrée des neurones du premier niveau de compétition. Elles renforcent l'activité des neurones de la première couche attachés aux pixels candidats à la fermeture. Cette influence agit

dans le cadre de chaque orientation et met encore en œuvre un mécanisme du type *On center-Off surround*. Les potentiels résultant $V_{s,k}$ vérifient l'équation :

$$\frac{d}{dt} V_{s,k} = -V_{s,k} + g(Z_{s,k}) - V_{s,k} \cdot \sum_{s' \in \nu_s} g(Z_{s',k}) w_{s's} \quad (6-16)$$

où $g(\cdot)$ est une fonction de seuillage.

Cette équation est de la même forme que l'équation (6-8). Le potentiel $V_{s,k}$ est injecté à l'entrée du neurone $B_{s,k}$ de la couche d'entrée. Pour la première couche, en introduisant $V_{s,k}$ dans l'équation (6-8), l'équation (6-9) devient :

$$c_{s,k} = \frac{c + \gamma \cdot b_{s,k} + v_{s,k}}{1 + \gamma \cdot \sum_{s' \in \nu_s} \gamma \cdot b_{s',k} \cdot w_{s's}} \quad (6-17)$$

La stabilité de ces équations différentielles est étudiée en détail dans [COHE83], [GROS88]. Les travaux présentés dans [GROS91] montrent que le comportement oscillatoire d'une telle boucle de rétroaction non linéaire pouvait expliquer les expériences qui mettent en évidence une activité oscillatoire synchrone (ou résonance à une fréquence de 40-60 Hz) dans les premières aires du cortex visuel (aires 17 et 18).

2.2 Réalisation d'un système d'extraction et de fermeture des contours avec rétroaction

Les cartes de neurones précédentes ont fait l'objet d'une implantation synchrone. On suppose pour cela que le temps de convergence des différents sous-réseaux est très court par rapport aux temps de modification de l'information en entrée (dans le cas contraire, la théorie ne serait pas applicable). Ainsi, il n'est pas nécessaire de simuler directement les équations différentielles décrivant les réseaux de neurones. Il suffit d'utiliser les expressions représentant l'état d'équilibre des neurones sur les différentes couches.

2.2.1 Description de l'implantation

Dans l'implantation proposée dans [GAUS92] deux directions de masques ont été utilisées (verticale et horizontale), la structure générale du dispositif est celle de la figure 6.7. Les contours sont construits en 4-connexité et peuvent être épais lorsqu'ils sont en diagonale. Les tentatives de fermeture ne pourront se faire que dans les deux directions horizontale et verticale. Ce problème n'est pas gênant pour l'interprétation. Pour simuler une carte particulière, l'algorithme consiste à déplacer une fenêtre sur toute l'image d'entrée ou plus généralement sur toute la couche d'entrée. En effet, tous les neurones d'une couche donnée ont les mêmes entrées et sont décrits par les mêmes équations. L'ensemble des neurones du réseau pourra ainsi être considéré comme autant de tableaux ou d'images qu'il y a de couches dans le réseau.

Afin de simplifier l'algorithme et d'accroître la vitesse d'exécution, la *première étape*, qui est la construction des cartes « gradient horizontal » et « gradient vertical », utilise des masques carrés. Ils permettent d'avoir une mesure du contraste entre deux régions associées à des pixels adjacents. Les points de contours éventuels seront placés entre deux pixels.

Les cartes gradient sont évaluées une seule fois. Les résultats sont enregistrés dans deux tableaux disponibles ensuite pour toutes les itérations de la fermeture des contours. Les sorties sont normalisées entre 0 et 1. Pour éviter les effets de bords, les contours ne sont recherchés que sur les parties où l'information donnée pour la convolution est complète. Ainsi, aucun contour n'est calculé sur une bande de la largeur du masque à partir du bord de l'image. La fonction de seuillage choisie est une extension de celle proposée en (6-6) :

$$|Ag_{s,k} - \alpha Ad_{s,k}| > \text{seuil} \quad (6-18)$$

On retrouve l'équation de Grossberg en prenant le seuil égal à 0. Si α et la valeur du seuil ne sont pas correctement choisis, on peut réaliser un seuillage important qui va se traduire par une perte d'information pour la fermeture des contours. Il y a un compromis difficile à trouver. Une solution satisfaisante pour ces problèmes de réglage se situe dans le couplage avec la partie « interprétation » du système de vision proposé dans [GAUS91].

La *deuxième étape* est une compétition entre les résultats de la première étape de façon à obtenir des contours fins (*cf.* figure 6.17d). Il faut que la zone de compétition soit supérieure ou égale aux dimensions des masques de détection des contours. Pour simplifier, une zone de compétition de même taille que les masques a été choisie. Quand on augmente la taille de la zone de compétition, on réduit la densité de contours extraits dans une image.

Il est nécessaire de normaliser l'activité des neurones pour éviter qu'elle ne croisse trop par rapport à la compétition. La normalisation est aussi indispensable pour éviter le phénomène inverse à savoir la disparition de la totalité des contours.

Pour accélérer l'exécution du mécanisme de compétition, la solution retenue a consisté à prendre directement la sortie la plus grande parmi les neurones en compétition. Les résultats sont plus précis que ceux de Grossberg où il était difficile de choisir les coefficients pour obtenir des contours fins tout en ayant un temps d'exécution acceptable. Une valeur négative est associée aux points non sélectionnés par la compétition. Cette valeur est inversement proportionnelle à la distance au maximum local responsable de l'inhibition. On obtient ainsi une approximation globale et rapide du mécanisme d'inhibition latérale.

La figure 6.17 présente les résultats obtenus sur l'image COULOIR. Les contours obtenus après la compétition et avant la première étape de fermeture apparaissent sur l'image 6.17.d. La figure 6.17.b montre les neurones « candidats » à la fermeture de contours et correspondant au deuxième terme de

l'équation (6-14). Le gris clair est employé pour les neurones actifs de la carte verticale, le noir pour les neurones actifs de la carte horizontale.

La figure 6.17e montre les contours obtenus après une première rétro-action de fermeture (*cf.* figure 6.15). En supprimant le seuillage de l'activité, on observe bien, dans le voisinage des contours vrais, une activité non nulle des neurones de direction orthogonale (*cf.* figure 6.17b). Ces neurones facilitent les tentatives de fermeture dans le mécanisme de rétro-action. Après la deuxième rétro-action, les améliorations ne sont plus significatives, le système se stabilise.

2.2.2 Conclusion

Nous avons choisi d'exploiter les états d'équilibre des équations différentielles car l'un des problèmes majeurs de la méthode Grossberg est lié au choix des constantes (coefficient d'amplification pour le retour, taille des fenêtres...). Elles influent sur la rapidité de convergence et la qualité des résultats (obtention de contours réellement fins et fermeture des contours). Ces difficultés proviennent de la nature du modèle.

Il est possible d'utiliser plusieurs tailles de masques (multi-échelle) pour supprimer le bruit et les structures de taille inférieure à celles liées à la résolution étudiée. L'opérateur proposé par Grossberg a l'avantage de ne pas trop déformer les contours lorsque la résolution diminue. Il préserve les angles.

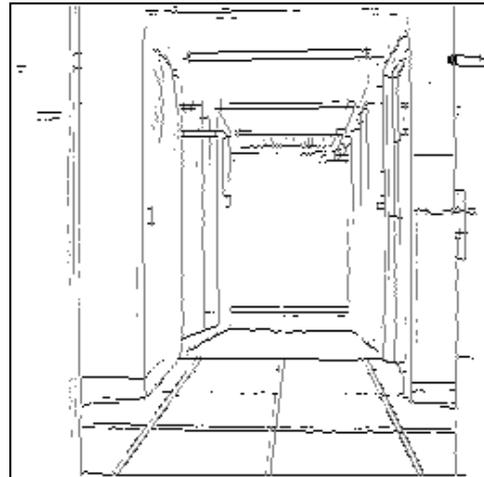
Pour généraliser ce modèle, il faut introduire plus de souplesse dans l'architecture du réseau (le choix des liaisons) et les valeurs des coefficients synaptiques. L'un des intérêts majeurs du bouclage est surtout de permettre un seuillage local adaptatif de l'image qui agit en fonction de l'éclairement moyen sur un voisinage de l'image. Enfin, il ne faut pas oublier qu'il ne s'agit que d'une fermeture préattentive des contours ne mettant pas en œuvre de mécanisme d'interprétation. Le dispositif présenté ne concurrence pas les extracteurs de contours comme ceux présentés au chapitre V. Il s'agit d'un modèle de structure pouvant facilement se raccorder à un système d'interprétation et dans lequel la remise en cause des décisions prises au bas-niveau pourra s'effectuer de manière élégante et naturelle en modifiant les contours suivant le procédé utilisé par la couche de coopération pour fermer les contours.

3 Extraction de points caractéristiques

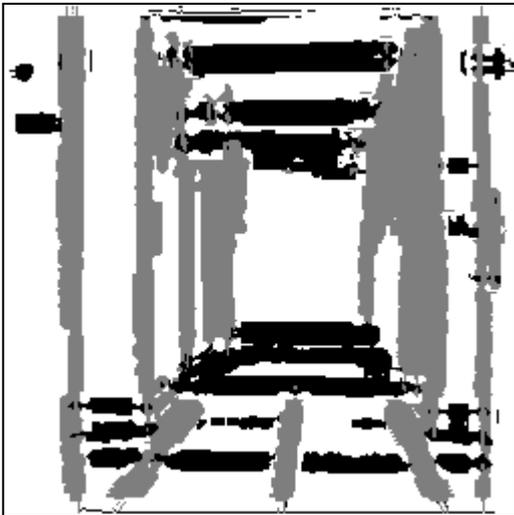
Les points caractéristiques se définissent, en général, à partir des contours. Ils comprennent les fins de lignes et les points de forte courbure. Ces points sont exploités pour la fermeture de contours et l'élaboration de contours virtuels. Ils constituent aussi des points de focalisation de l'attention en interprétation d'images.



a) original



d) neurones contours actifs après compétition



b) neurones candidats à la fermeture

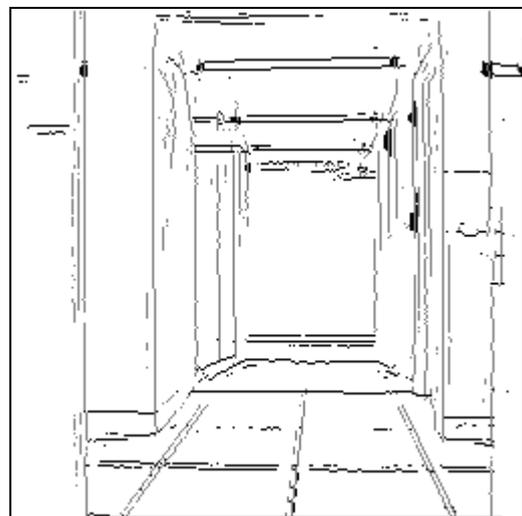
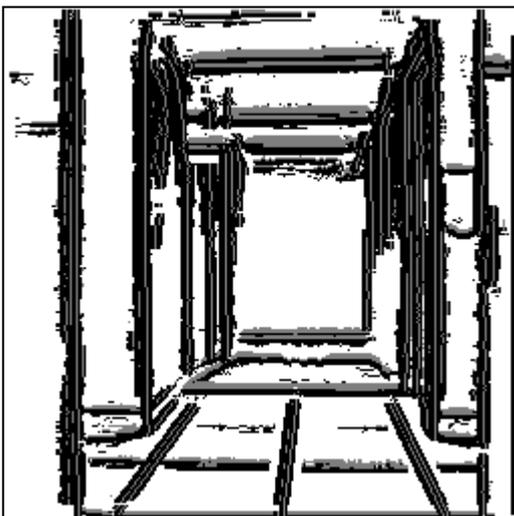
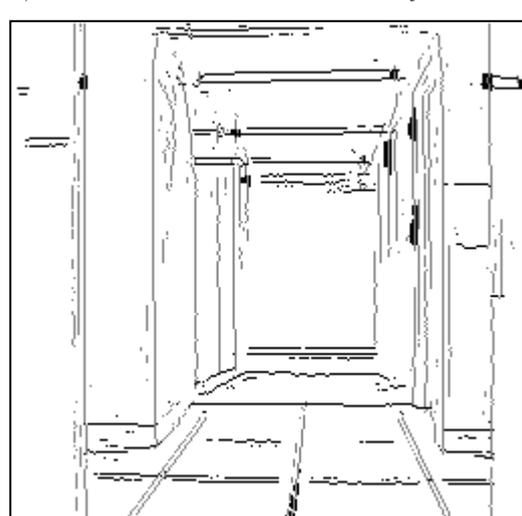
e) résultat de la 1^{ère} rétro-action de fermeturec) neurones contours et neurones contours \perp f) résultat de la 2^{ème} rétro-action de fermeture

Figure 6.17 : Contours extraits par le système neuronal de la figure 6.7.

Il existe de nombreuses méthodes d'extraction de points caractéristiques. Bien que certaines utilisent des masques pour travailler directement sur une image en niveaux de gris, la plupart exploite les images de contours. Les méthodes les plus répandues consistent à parcourir un contour et à détecter les variations de directions du contour. Quand celles-ci sont supérieures à un certain seuil, l'algorithme détecte un angle. Ces algorithmes nécessitent de définir *a priori* au moins deux paramètres, le premier correspond à la distance sur laquelle on doit intégrer la courbe (polygonalisation) et le deuxième correspond à un seuil à partir duquel on considère qu'il y a un angle [RAME70], [PAVL74], [SLAN80] [CHAS91]. On peut aussi détecter les angles en les représentant dans un espace paramétrique (transformée de Hough [BALL81]) ou en utilisant un apprentissage par compétition [RUME85].

Nous présentons une approche exploitant des mécanismes de diffusion et de compétition qui permet d'expliquer les illusions visuelles.

3.1 Utilisation de la diffusion pour l'extraction de points caractéristiques

Cette approche exploite les conclusions de différents travaux sur la localisation des points anguleux prenant en compte des études faites sur la modélisation d'illusions visuelles [GROS87], [SEIB 89] et [OZAW 90].

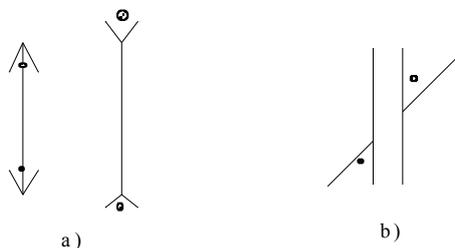


Figure 6.18 : Illusions visuelles illustrant le mécanisme d'extraction de points caractéristiques (illusions de Muller-Lyer)

Sur le dessin de la figure 6.18a, les deux flèches semblent avoir des longueurs différentes, nous percevons les extrémités des deux segments verticaux à des endroits différents. De la même manière, pour le dessin 6.18b, la ligne oblique qui coupe les deux segments verticaux paraît être composée de deux segments non alignés. On peut ici aussi supposer que la mauvaise localisation des extrémités des segments explique l'erreur.

A partir de cette constatation, plusieurs modèles ont été imaginés pour expliquer ces illusions. Le principe consiste à faire diffuser les contours et à ne garder que les maximums locaux. L'équation différentielle de la diffusion de la chaleur a pour solution une gaussienne. Un neurone, ayant ses poids fixés pour correspondre à une gaussienne en trois dimensions, donnera ainsi le résultat de la diffusion en un point particulier (masque d'entrée). Les points appartenant à un contour sont les entrées du réseau mises à 1. Pour tous les autres pixels de l'image,

l'entrée associée est fixée à 0. Si l'image est une droite isolée, les neurones correspondant à cette droite auront tous la même valeur. Après la compétition, tous les points correspondant à la droite seront supprimés sauf les deux extrémités. Lorsque l'image comporte un angle formé par 2 segments, la partie aiguë de l'angle voit se superposer l'influence de la diffusion des contours des deux segments. Il apparaît un maximum qui est supérieur aux résultats obtenus pour les points des segments (*cf.* figure 6.19). Après la compétition, ce point sera donc conservé. Plus l'angle est aigu, plus son intensité sera grande. Par ailleurs, plus le domaine de diffusion sera important plus le point caractéristique sera éloigné de l'angle.

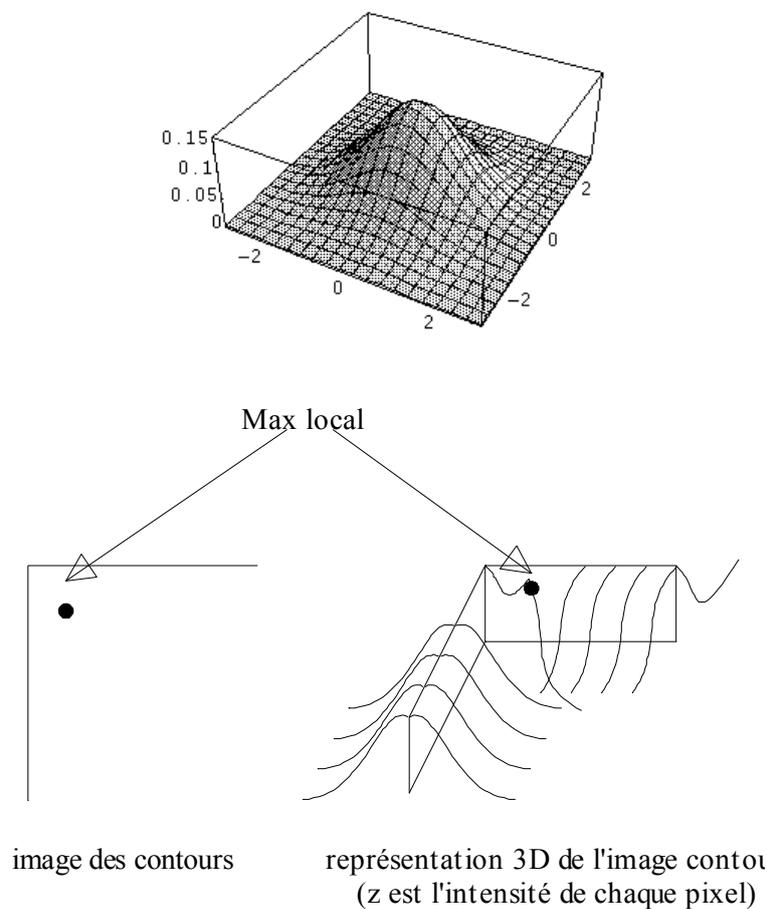


Figure 6.19 : Forme de la gaussienne et point caractéristique extrait pour un angle droit.

Malheureusement, sur des images bruitées, les pixels correspondant au bruit sont responsables de la présence de points caractéristiques correspondant à des contours anguleux de très faibles dimensions. Pour les supprimer, il faut introduire un mécanisme permettant de sélectionner une résolution à laquelle ces points ne sont plus présents mais où les points caractéristiques « intéressants » apparaissent malgré tout. Il faut donc utiliser un nouveau type de masque accordé sur une fréquence spatiale particulière correspondant à celle de l'objet à analyser.

3.2 Détection des points caractéristiques grâce à des cellules *Center-Off*

Il est possible de sélectionner uniquement les points caractéristiques présents à partir d'une résolution spatiale donnée grâce à une modification de la forme du masque précédent de manière à ce que, dans un voisinage donné des points de contours servant de support à la diffusion, aucun point caractéristique ne puisse s'activer. Cela revient donc à introduire une zone d'inhibition. Le nouveau masque pourra donc être facilement réalisé à partir de la différence de deux gaussiennes de même centre (ou laplacien d'une gaussienne). La forme du masque sera donc celle d'une cellule *Center-Off* [GROS82]

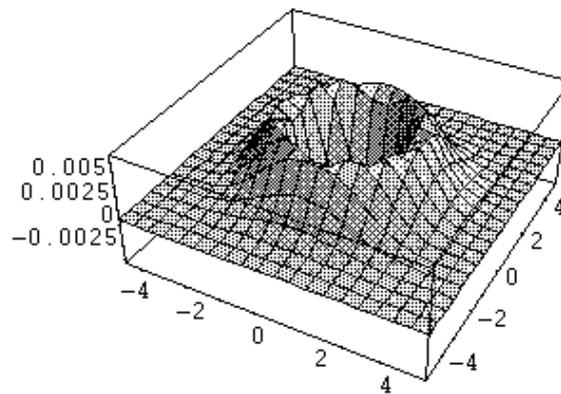


Figure 6.20 : Forme du masque d'extraction de points caractéristiques (cellule *Center-Off*)

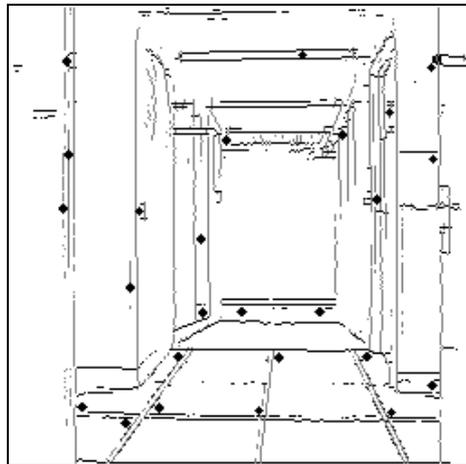


Figure 6.21 : Exemple de contours et de points caractéristiques extraits

Selon la taille du masque et le choix des paramètres de diffusion, le filtre peut être accordé sur des fréquences spatiales particulières. Afin de limiter le décalage des points caractéristiques obtenus lorsque la taille du masque augmente, on peut se servir d'un masque mettant en évidence des points caractéristiques à une basse résolution pour sélectionner les zones où l'on extraira des points caractéristiques à

une meilleure résolution. De cette manière, la localisation des points caractéristiques reste précise et n'est pas perturbée par les zones de contours bruités.

Sur une image de contours, le résultat de la convolution avec ce masque correspond à une érosion en morphologie mathématique. De même, on peut « dilater » les contours en utilisant la fonction opposée. Cette succession d'opérations permet de faire disparaître certains contours et d'en fermer d'autres (lissages spatiaux-fréquentiels).

4 Conclusion

L'essentiel de ce chapitre est consacré à la présentation d'une approche d'inspiration neurobiologique pour la détection et la fermeture de contours et l'extraction de points caractéristiques. Il apparaît que cette approche repose sur deux aspects : le parallélisme massif et l'interconnexion forte entre les différentes couches de traitements. Ce dernier point est important car il règle structurellement le problème de la communication entre les traitements de haut et de bas niveau et il procure des voies de recherche pour la conception de systèmes complets de vision.

Approche frontière : méthodes markoviennes

1 Introduction

L'utilisation des champs de Markov [KIND80], [AZEN87] en vision par ordinateur s'est développée depuis quelques années. Les principales applications de ce type de modélisation sont la détection de contours [BLAK87], [GEIG90], [SIMC89a], [GEMA90], [ZERU90] la restauration ou le lissage d'image [BLAK87], [GEMA84], [MARR87], [SIMC89b], [ZERU93], la stéréovision [YUIL90] et la classification de textures [GEMA87a], [GEMA87b], *etc.*

Le principal avantage des champs de Markov est de fournir un modèle mathématique simple permettant de prendre en compte les différents niveaux d'interprétation. Le plus souvent utilisé dans un cadre bayésien dans les premiers travaux, cette approche consiste alors à construire un modèle *a priori* portant sur le processus à reconstruire (ou à estimer) et un modèle de déformation ayant conduit à l'observation faite. Ces modèles sont ensuite combinés, le cadre markovien ayant justement l'intérêt d'être suffisamment large pour effectuer tous ces travaux de modélisation.

Une objection souvent faite à l'utilisation des champs de Markov est le coût de calcul relatif aux algorithmes permettant de calculer les estimateurs des processus recherchés. Ces temps de calcul sont très fortement liés à la forme de l'énergie utilisée et à la complexité de la tâche à effectuer plus qu'à l'approche stochastique elle-même.

Quelques généralités sur les champs markoviens ainsi que les algorithmes stochastiques classiques de simulation : l'échantillonneur de Gibbs et la dynamique de Métropolis ; ou d'optimisation : l'algorithme déterministe ICM (*Iterated Conditional Mode*) et le recuit simulé sont présentés dans l'annexe B.

Dans ce chapitre, nous présentons trois méthodes de restauration et de détection de contours. Le paragraphe 2 présente le cadre général et les notations utilisés pour les trois modèles. Les deux premiers modèles font appel à une optimisation par recuit simulé avec échantillonneur de Gibbs. Le troisième modèle fait appel à deux algorithmes déterministes de relaxation qui sont présentés au paragraphe 3 : le GNC (*Graduated Non Convexity*) et le recuit par champs moyens (MFA ou *Mean Field Annealing*).

Le cadre dans lequel nous nous plaçons est celui de la détection de contours dans des images numérisées qui peuvent, que ce soit par la technique d'acquisition, ou par la technique de numérisation elle-même, se trouver entachées de bruit. Parmi les méthodes présentées dans ce chapitre, la première et la troisième traitent à la fois du problème de la restauration (lissage de l'image) et de l'extraction des contours. Pour ces deux méthodes, les contours obtenus sont essentiellement des frontières de régions et donc des contours fermés. Il est important de remarquer que ces deux méthodes fournissent simultanément, et donc pour un même coût de calcul, les deux résultats : restauration et contours.

Le coût de calcul, qui est, il est vrai, supérieur à celui de méthodes plus classiques, peut s'avérer parfaitement justifié par le fait que l'on obtient simultanément les deux types de résultats (restauration et contours) mais aussi et surtout par le fait que ces méthodes se révèlent généralement plus performantes sur des images sensiblement bruitées (voir par exemple les résultats présentés sur l'image IRM à la fin de ce chapitre et au chapitre IX).

Le second modèle présenté dans ce chapitre porte uniquement sur les contours mais il a la particularité de pouvoir être utilisé sur des textures (micro- mais aussi macro-textures).

Du fait du caractère local de l'approche markovienne, plusieurs techniques de parallélisation existent et sont utilisables sur chacun des trois modèles [ZERU91], [MEMI91], [AZEN92].

2 Restauration et détection de bords

Dans le cadre de la restauration, il s'avère intéressant de définir un champ booléen dual du processus intensité, appelé *processus de ligne*. Ce champ, introduit pour la première fois par Geman et Geman [GEMA84], représente explicitement la présence ou l'absence de discontinuités et vient ainsi rompre l'hypothèse de lissage faite par les méthodes classiques de régularisation [MARR85]. Ce processus de ligne sera noté \mathbf{B} ou (\mathbf{X}, \mathbf{Y}) suivant qu'on distingue ou non les processus horizontaux et verticaux dans les modèles considérés. Nous noterons aussi $\langle s, r \rangle$ le *site bord* se trouvant entre les sites pixels s et r (figure 7.1).

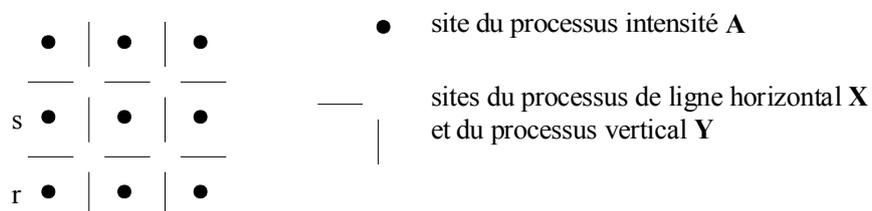


Figure 7.1 : Superposition des 2 réseaux

\mathbf{A} , λ et \mathbf{B} désignent respectivement les champs aléatoires des observations, de l'image restaurée et des bords ; \mathbf{a} , λ , \mathbf{b} représentent une réalisation de ces champs.

On définit alors une énergie sur le triplet $(\mathbf{A}, \boldsymbol{\lambda}, \mathbf{B})$ qui est souvent composée d'une somme de termes ayant des significations différentes. Un grand nombre d'énergies peuvent être utilisées, les termes qu'elles peuvent contenir sont essentiellement de quatre types :

- Interaction entre l'image restaurée $\boldsymbol{\lambda}$ et l'image observée \mathbf{a} .
- Régularisation ou lissage de l'image restaurée, qui peut prendre en compte les bords \mathbf{b} (ou processus de ligne).
- Interaction entre les bords et l'image restaurée (\mathbf{b} et $\boldsymbol{\lambda}$).
- Régularisation des bords.

Nous allons ici citer deux énergies construites par D. Geman et S. Geman à titre d'exemples et un troisième modèle utilisé par J. Zerubia dans les expériences présentées plus loin.

2.1 Premier modèle

Ce premier modèle est en fait un modèle de restauration d'images, mais il contient un processus de ligne et, de ce fait, il fournit aussi des frontières. De plus, il s'agit du premier modèle de ce type introduit dans la littérature [GEMA84].

L'énergie de ce modèle s'écrit :

$$U(\mathbf{b}, \mathbf{a}) = U_1(\mathbf{a}) + \alpha U_2(\mathbf{b}) + U_3(\mathbf{b}) \quad (7-1)$$

$$U_1(\mathbf{a}) = \sum_{s \in S} \frac{(\lambda_s - a_s)^2}{2\sigma^2} \quad (7-2)$$

$$U_2(\mathbf{b}) = \sum_{\langle s, r \rangle} (\mathbf{1}_{\lambda_s = \lambda_r} - \mathbf{1}_{\lambda_s \neq \lambda_r}) \mathbf{1}_{b_{\langle s, r \rangle} = 0} \quad (7-3)$$

où $\mathbf{1}_P = \begin{cases} 1 & \text{si la propriété } P \text{ est vraie} \\ 0 & \text{sinon.} \end{cases}$

$$U_3(\mathbf{b}) = \sum_{c \in C} V(c) \quad (7-4)$$

Les différents termes de cette somme correspondent aux types d'interactions définies dans la section précédente : U_1 est le terme de rappel aux données \mathbf{a} , U_2 est le terme de régularisation de l'image restaurée $\boldsymbol{\lambda}$ tenant compte des éléments de bords \mathbf{b} , et U_3 est un terme de régularisation des bords (C est un ensemble de cliques de la grille des bords, voir figure 7.2). $\langle s, t \rangle$ est le site bord se trouvant entre les deux sites pixels s et t , et, par conséquent, la régularisation du processus de restauration $\boldsymbol{\lambda}$ est stoppée par la présence d'un de ces bords, c'est-à-dire pour $b_{\langle s, t \rangle} = 1$, ce qui permet de reconstituer des régions de niveaux de gris constants par morceaux.

Avant de détailler l'expression de $V(c)$, notons tout d'abord que ce modèle utilise deux paramètres : σ et α . σ est l'écart-type que l'on accepte à l'intérieur d'une région homogène, et α définit l'influence relative que l'on accorde au terme d'interaction avec les données par rapport au terme de régularisation. Ces deux paramètres sont donc très importants. Comme il n'est pas développé de méthode de calcul de ces paramètres dans l'article cité, il faut les sélectionner interactivement ; cette démarche nécessite une certaine habitude du modèle.

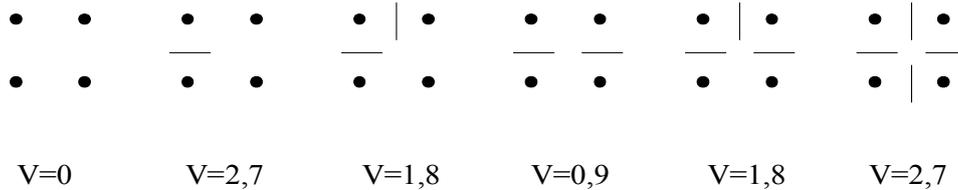


Figure 7.2 : Cliques d'ordre 4 et valeurs correspondantes de V . Les mêmes valeurs sont attribuées aux cliques déduites de celles-ci par rotation.

Les fonctions potentielles $V(c)$ sont de simples pondérations traduisant des pénalisations relatives sur les différentes cliques d'ordre 4 que l'on peut rencontrer, elles sont représentées à la figure 7.2.

L'estimation de λ et de \mathbf{b} est réalisée par un MAP, en utilisant un recuit simulé (voir [GEMA84] pour les exemples).

2.2 Deuxième modèle

Ce deuxième modèle est en fait un modèle de détection de bords uniquement, il n'y a donc pas de processus de restauration λ , et le modèle porte uniquement sur \mathbf{A} et \mathbf{B} . Par contre, la grille des sites bords est plus complexe que précédemment : les frontières se situent sur une grille rectangulaire, et les éléments de bord, qui sont des informations induites directement des données et qui jouent le rôle de rappel aux données, sont situés sur la grille duale définie précédemment (ces grilles sont représentées dans la figure 7.3). Les éléments de bords sont obtenus par seuillage et ne sont pas modifiés ensuite.

L'énergie de ce modèle s'écrit :

$$U(\mathbf{b}, \mathbf{a}) = \theta_1 U_1(\mathbf{b}) + \theta_2 U_2(\mathbf{a}, \mathbf{b}) + \theta_3 U_3(\mathbf{b}) \quad (7-5)$$

$$U_1(\mathbf{b}) = \sum_{s \in \mathcal{S}_b} (b_s - \eta_s)^2 \quad (7-6)$$

où \mathcal{S}_b est l'ensemble des sites du processus frontière

$$U_2(\mathbf{a}, \mathbf{b}) = \sum_{\langle s, r \rangle} b_s b_r \Psi(\Delta_{\langle s, r \rangle}) \quad (7-7)$$

$$U_3(\mathbf{b}) = \sum_{c \in \mathcal{C}} \Phi(c) \quad (7-8)$$

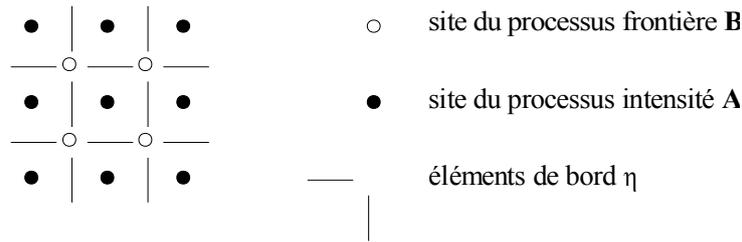


Figure 7.3 : Différentes grilles de sites.

Les différents termes de cette somme peuvent être interprétés comme précédemment en termes de rappel aux données et de régularisation. η_s est égal à 1 s'il y a une chaîne de 4 éléments de bord contenant le site s , et à 0 sinon, le terme U_1 est donc bien un terme de rappel des frontières vers des points de gradient élevé. $\Delta_{\langle s,r \rangle}$ est la différence entre les niveaux de gris situés de part et d'autre des deux sites frontières s et r , et $\Psi(u) = \frac{u^4}{K + u^4}$. Le terme U_2 joue donc sensiblement le même rôle que le précédent mais sans utiliser de seuillage.

Les fonctions potentielles $\Phi(c)$ sont de nouveau des pondérations de différentes cliques d'ordre 4 ou 5 que nous ne détaillerons pas ici.

Les paramètres θ_i sont cette fois calculés grâce à une sélection d'une liste de cas particuliers pour lesquels on peut choisir le comportement de la loi *a posteriori*.

L'estimation de \mathbf{b} est là encore réalisée par un MAP, donc par un recuit simulé (voir [GEMA90] pour les exemples).

2.3 Troisième modèle

Dans le modèle utilisé, on considère deux processus de ligne, \mathbf{X} et \mathbf{Y} (horizontal et vertical, cf. figure 7.1). L'énergie peut alors s'exprimer par :

$$U = D + S + P \quad (7-9)$$

Dans les expressions suivantes les indices i, j désignent le site $s = (i, j) \in \mathcal{S}$

Le terme $D = \sum_{i,j} (\lambda_{i,j} - a_{i,j})^2$ assure une bonne adéquation entre la solution trouvée et les données,

$$S = \sum_{i,j} \gamma^2 \left((\lambda_{i,j} - \lambda_{i-1,j})^2 (1 - x_{i,j}) + (\lambda_{i,j} - \lambda_{i,j+1})^2 (1 - y_{i,j}) \right) \quad (7-10)$$

qui représente la contrainte de lissage classique ($2 \times \text{gradient}^2$) en tenant compte du processus de ligne,

$$P = \sum_{i,j} \alpha (x_{i,j} + y_{i,j}) \quad (7-11)$$

où $\alpha (\alpha > 0)$ est le coût à payer pour l'introduction d'une ligne.

Le modèle décrit par l'équation ci-dessus est celui d'une membrane à contraintes de continuité lâches [GEMA84], [BLAK87], [GEIG90].

3 Deux algorithmes déterministes

3.1 GNC (*Graduated Non Convexity*)

Pour minimiser l'énergie décrite par le troisième modèle qui est une fonction non-convexe, Blake et Zisserman [BLAK87] ont proposé un algorithme déterministe de relaxation, appelé GNC (*Graduated Non Convexity*). L'idée de cet algorithme est la suivante : au lieu de minimiser l'énergie à la fois par rapport au processus de restauration λ et aux processus de ligne \mathbf{X} et \mathbf{Y} , on minimise l'énergie par rapport aux processus de ligne dans un premier temps. On obtient alors une énergie ne dépendant plus que de l'image restaurée que l'on va chercher à minimiser dans un deuxième temps. On construit d'abord une approximation convexe de l'énergie. Puis l'on construit une série d'énergies de plus en plus proches de celle que l'on souhaite minimiser. Blake et Zisserman ont montré en annexe de leur ouvrage comment calculer une approximation convexe pour ce type d'énergie.

Les techniques classiques de minimisation consistent soit en l'application d'une méthode déterministe, de type descente suivant la plus grande pente (auquel cas le système reste bloqué dans le premier minimum local rencontré), soit en l'utilisation d'une technique stochastique de recuit simulé (qui assure la convergence asymptotique vers le minimum global mais qui demande *a priori* un coût de calcul très élevé).

La méthode du GNC assure de trouver le minimum global dans des cas particuliers et donne de bonnes solutions dans les autres cas (*cf.* chapitre 7 de [BLAK87] pour plus de détails). Nous en rappelons brièvement le principe ci-dessous.

L'énergie décrite par le troisième modèle peut se mettre sous la forme :

$$U = D + \sum_{i,j} h_{\alpha,\gamma}((\lambda_{i,j} - \lambda_{i-1,j}), x_{i,j}) + \sum_{i,j} h_{\alpha,\gamma}((\lambda_{i,j} - \lambda_{i,j+1}), y_{i,j}) \quad (7-12)$$

avec

$$h_{\alpha,\gamma}(u, x) = \gamma^2 u^2 (1-x) + \alpha x \quad (7-13)$$

La minimisation de U doit être faite par rapport à $\lambda_{i,j}$, $x_{i,j}$, $y_{i,j}$. Cependant, le terme D ne contient pas de processus de ligne. La minimisation par rapport à $x_{i,j}$ et $y_{i,j}$ peut être effectuée avant celle par rapport à $\lambda_{i,j}$:

$$\min_{\lambda_{i,j}} U = \min_{\lambda_{i,j}} \left\{ D + \sum_{i,j} g_{\alpha,\gamma}(\lambda_{i,j} - \lambda_{i-1,j}) + \sum_{i,j} g_{\alpha,\gamma}(\lambda_{i,j} - \lambda_{i,j+1}) \right\} \quad (7-14)$$

avec

$$g_{\alpha,\gamma}(u) = \min_{x \in [0,1]} h_{\alpha,\gamma}(u, x) \text{ ou } \min_{y \in [0,1]} h_{\alpha,\gamma}(u, y) \quad (7-15)$$

L'idée de base du GNC est la suivante : la première étape est la construction d'une approximation convexe U^* de l'énergie U ; la minimisation de U^* donne donc un minimum global. Puis une séquence d'énergies $U^{(p)}$ est construite telle que $U^0 = U$ et $U^1 = U^*$. Le GNC optimise toute la séquence d'énergies $U^{(p)}$ (par exemple, $p = 1, 1/2, 1/4, 1/8...$ pour une image) en utilisant comme conditions initiales :

- les données, pour la minimisation de U^* ,
- la solution trouvée lors de la minimisation de la fonction énergie précédente dans les autres cas.

Blake et Zisserman ont montré dans [BLAK87] que la fonction $g_{\alpha,\gamma}^{(p)}$ qui assure un hessien positif est la suivante :

$$g_{\alpha,\gamma}(u) = \begin{cases} \gamma^2 u^2 & \text{si } |u| < \rho \\ \alpha - \frac{\chi(|u| - \beta)^2}{2} & \text{si } \rho \leq |u| < \beta \\ \alpha & \text{si } |u| \geq \beta \end{cases}$$

$$\text{avec : } \chi = \frac{1}{4p}, \beta^2 = \alpha \left(\frac{2}{\chi} + \frac{1}{\gamma^2} \right), \rho = \frac{\alpha}{\gamma^2 \beta} \quad (7-16)$$

et $u = (\lambda_{i,j} - \lambda_{i-1,j})$ ou $(\lambda_{i,j} - \lambda_{i,j+1})$ suivant la direction considérée.

Une fois la minimisation par rapport au champ de l'image restaurée $\lambda_{i,j}$ effectuée sur toute la série d'énergies $U^{(p)}$, les processus de ligne (donc les contours) sont obtenus de la façon suivante :

$$x_{i,j} = \begin{cases} 1 & \text{si } |\lambda_{i,j} - \lambda_{i-1,j}| > \beta \\ 0 & \text{si } |\lambda_{i,j} - \lambda_{i-1,j}| < \rho \\ \text{ambigu} & \text{sinon} \end{cases} \quad (7-17)$$

Pour obtenir $y_{i,j}$, il suffit d'effectuer le même test en remplaçant $\lambda_{i-1,j}$ par $\lambda_{i,j} + 1$.

Une extension de cette méthode au cas d'un modèle anisotrope de champs de Markov (caractérisé par θ_x , θ_y et une variance v) a été réalisée par Simchony *et al* [SIMC89b].

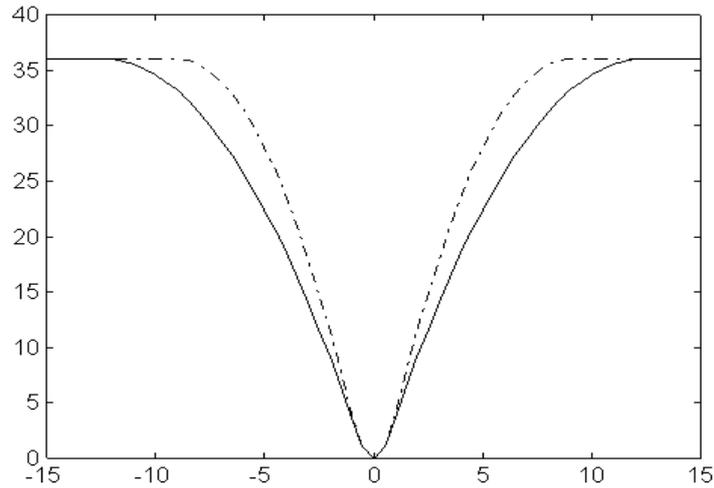


Figure 7.4 : Fonction $g_{\alpha, \gamma}^{(p)}(u)$; $\alpha=36$, $\gamma=2$
 trait discontinu : $p=1/4$; trait continu : $p=1/2$;

Choisir γ et ξ_0 ;

/* γ est un facteur de lissage, l'effet du lissage est d'autant plus sensible que γ est grand */

$$\alpha = \frac{\xi_0^2 \gamma}{2}$$

$$\omega = 2 \left(1 - \frac{1}{\gamma \sqrt{2}} \right)$$

$$p \in \left\{ 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots \right\}$$

POUR chaque p, POUR n=1, 2, 3,...FAIRE

POUR i=1,...,M-1

POUR j=1,...,N-1 FAIRE

$$\lambda_{i,j}^{(n+1)} = \lambda_{i,j}^{(n)} - \frac{\omega}{2 + 8\gamma^2} \left(\begin{array}{l} 2(\lambda_{i,j}^{(n)} - a_{i,j}) + g_{\alpha,\gamma}^{(p)}(\lambda_{i,j}^{(n)} - \lambda_{i-1,j}^{(n+1)}) + \\ g_{\alpha,\gamma}^{(p)}(\lambda_{i,j}^{(n)} - \lambda_{i,j-1}^{(n+1)}) + g_{\alpha,\gamma}^{(p)}(\lambda_{i,j}^{(n)} - \lambda_{i+1,j}^{(n)}) + \\ g_{\alpha,\gamma}^{(p)}(\lambda_{i,j}^{(n)} - \lambda_{i,j+1}^{(n)}) \end{array} \right)$$

/*Traitements spécifiques pour les bords */

Pour i=0,j=0 FAIRE :

$$\lambda_{0,0}^{(n+1)} = \lambda_{0,0}^{(n)} - \frac{\omega}{2 + 4\gamma^2} \left(\begin{array}{l} 2(\lambda_{0,0}^{(n)} - a_{0,0}) + g_{\alpha,\gamma}^{(p)}(\lambda_{0,0}^{(n)} - \lambda_{1,0}^{(n)}) + \\ g_{\alpha,\gamma}^{(p)}(\lambda_{0,0}^{(n)} - \lambda_{0,1}^{(n+1)}) \end{array} \right)$$

{FAIRE un traitement identique au précédent pour les 3 autres angles}

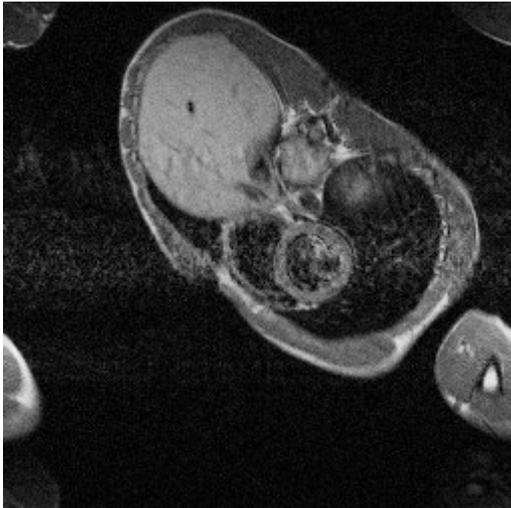
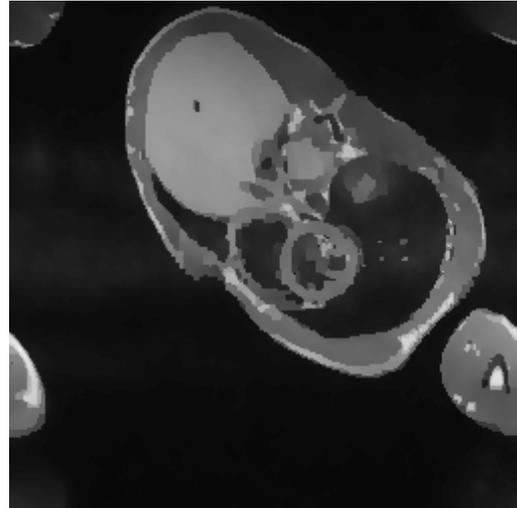
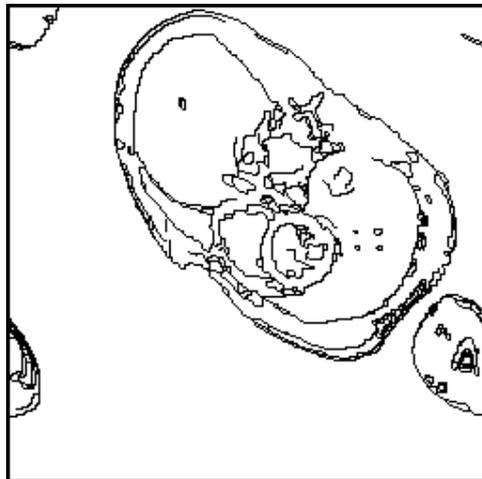
/*Traitement pour l'angle supérieur gauche*/

Pour i=0, j=1,...,N-1 FAIRE :

$$\lambda_{0,j}^{(n+1)} = \lambda_{0,j}^{(n)} - \frac{\omega}{2 + 6\gamma^2} \left(\begin{array}{l} 2(\lambda_{0,j}^{(n)} - a_{0,j}) + g_{\alpha,\gamma}^{(p)}(\lambda_{0,j}^{(n)} - \lambda_{i,j-1}^{(n+1)}) + \\ g_{\alpha,\gamma}^{(p)}(\lambda_{0,j}^{(n)} - \lambda_{1,j}^{(n)}) + g_{\alpha,\gamma}^{(n)}(\lambda_{0,j}^{(n)} - \lambda_{0,j+1}^{(n)}) \end{array} \right)$$

{FAIRE un traitement identique au précédent pour les 3 autres côtés}

Algorithme 7.1 : GNC d'après [BLAK87]

*Image originale IRM**Image restaurée**Contours extraits**Figure 7.5 : Restauration et segmentation par l'algorithme GNC*

3.2 Recuit par champs moyens (MFA)

Une autre méthode déterministe de relaxation basée sur un recuit par champs moyens (MFA : *Mean Field Annealing*) a été introduite en vision par Geiger et Giroso [GEIG90] afin de minimiser l'énergie du troisième modèle.

La technique d'approximation par champs moyens est souvent utilisée en physique statistique. En effet, l'obtention de façon analytique de la valeur moyenne d'un champ comme fonction explicite des données et des paramètres n'est en général pas possible. La physique statistique utilise alors comme outil

l'approximation par champs moyens afin d'obtenir une solution approchée donnée de façon implicite par un système d'équations non linéaires.

Tous les calculs reposent sur la fonction de partition $Z_{(T)}$ qui contient toute l'information du système physique considéré. Le calcul de la fonction de partition est équivalent à l'évaluation d'une intégrale multidimensionnelle qui ne peut pas être obtenue explicitement du fait de l'interaction entre les variables aléatoires $\lambda_s, s \in \mathcal{S}$. L'approximation par champs moyens consiste à substituer l'interaction entre les variables aléatoires λ_s de différents sites s par l'interaction de la variable aléatoire λ_s avec les valeurs moyennes des variables aléatoires des sites voisins (voisinage défini suivant l'ordre du champ de Markov considéré). Alors la fonction de partition peut se factoriser en un produit de fonctions de partition relatives à un seul site et être calculée de façon implicite par un système d'équations non-linéaires obtenues grâce à la dérivée du logarithme de la fonction de partition approximée.

Cette technique a également été employée récemment pour résoudre des problèmes combinatoires comme par exemple le partitionnement de graphe [BILB88] [HERA89], ou le stockage de formes types (*pattern storage*) [CAMP89] en utilisant des réseaux de neurones. Dans tous les cas, cette technique s'est avérée plus rapide que le recuit simulé (*cf.* [BILB88] par exemple).

Par définition, le champ moyen $\bar{\lambda}_{i,j}$ est donné par :

$$\bar{\lambda}_{i,j} = \frac{1}{Z_{(T)}} \sum_{\text{toutes les configurations } \lambda_y} \exp\left(-\frac{U(\lambda, \mathbf{x}, \mathbf{y})}{T}\right) \quad (7-18)$$

$U(\lambda, \mathbf{x}, \mathbf{y})$ étant donné par le troisième modèle. Les champs moyens $\bar{x}_{i,j}$ et $\bar{y}_{i,j}$ sont définis de façon analogue.

On peut montrer [GEIG91] que :

$$\bar{\lambda}_{i,j} = a_{i,j} + \frac{T}{2} \frac{\partial \ln Z_{(T)}}{\partial a_{i,j}} \quad (7-19)$$

$$\bar{x}_{i,j} = 1 - T \frac{\partial \ln Z_{(T)}}{\partial G_{i,j}^x} \quad (7-20)$$

où

$$G_{i,j}^x = \alpha - \gamma^2 (\lambda_{i,j} - \lambda_{i-1,j})^2 \quad (7-21)$$

Une équation similaire est définie pour $\bar{y}_{i,j}$.

Si nous voulons effectivement obtenir ces champs moyens, nous devons faire des approximations afin de calculer la fonction de partition $Z_{(T)}$ de façon explicite.

Considérons tout d'abord le calcul de $\bar{\lambda}_{i,j}$. La fonction de partition $Z_{(T)}$ est définie par :

$$\sum_{\text{toutes les configurations } \mathbf{x}, \mathbf{y}} \exp\left(-\frac{U(\mathbf{x}, \mathbf{y})}{T}\right) \quad (7-22)$$

La première étape consiste à éliminer les processus de ligne dans la fonction énergie U de telle sorte que :

$$U(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{y}) = D(\boldsymbol{\lambda}) + U^{\text{eff}}(\boldsymbol{\lambda}) \quad (7-23)$$

Ceci peut être calculé en considérant les processus de ligne \mathbf{X} et \mathbf{Y} comme étant plongés dans un champ extérieur (cf. [GEIG90] et [GEIG91] pour plus de détails).

La seconde étape est d'approximer $Z_{(T)}$.

$$Z_{(T)} = \sum_{\text{toutes les configurations}} \exp\left(-\frac{D() + U^{\text{eff}}()}{T}\right) \quad (7-24)$$

$$Z_{(T)} \approx \max \left\{ \exp - \frac{\sum_{i,j} (\lambda_{i,j} - a_{i,j})^2 + U_{i,j}^{\text{eff}}()}{T} \right\} \quad (7-25)$$

$$Z_{(T)} \approx \exp - \frac{\sum_{i,j} (\bar{\lambda}_{i,j} - a_{i,j})^2 + U_{i,j}^{\text{eff}}(-)}{T} \quad (7-26)$$

En pratique, cela revient à dire qu'au lieu de raisonner sur l'influence stochastique des champs des pixels voisins sur le champ du pixel (i,j) , il suffit de raisonner sur l'influence exercée par les valeurs moyennes des champs des pixels voisins. Cela revient à « figer » l'environnement du pixel (i,j) dans son état moyen.

Pour obtenir $\bar{\lambda}_{i,j}$, il suffit donc de minimiser l'énergie en faisant :

$$\frac{\partial (D() + U^{\text{eff}}())}{\partial \lambda_{i,j}} = 0 \quad (7-27)$$

ce qui donne :

$$\begin{aligned} \bar{\lambda}_{i,j} = & \bar{a}_{i,j} - \gamma^2 (\bar{\lambda}_{i,j} - \bar{\lambda}_{i,j+1}) (1 - \bar{y}_{i,j}) + \gamma^2 (\bar{\lambda}_{i,j-1} - \bar{\lambda}_{i,j}) (1 - \bar{y}_{i,j-1}) \\ & - \gamma^2 (\bar{\lambda}_{i,j} - \bar{\lambda}_{i-1,j}) (1 - \bar{x}_{i,j}) + \gamma^2 (\bar{\lambda}_{i+1,j} - \bar{\lambda}_{i,j}) (1 - \bar{x}_{i+1,j}) \end{aligned} \quad (7-28)$$

En partant des équations définies ci-dessus et en utilisant l'approximation par champs moyens, il vient, après quelques calculs :

$$\bar{x}_{i,j} = \sigma_{\theta} \left(\gamma^2 (\bar{\lambda}_{i,j} - \bar{\lambda}_{i-1,j})^2 - \alpha \right) \quad (7-29)$$

$$\bar{y}_{i,j} = \sigma_{\theta} \left(\gamma^2 (\bar{\lambda}_{i,j} - \bar{\lambda}_{i,j+1})^2 - \alpha \right) \quad (7-30)$$

$$\text{avec } = \frac{1}{T} \text{ et } \sigma_{\theta}(x) = \frac{1}{1 + \exp(-\theta x)}$$

Il faut remarquer que, contrairement à la méthode du GNC, les processus de lignes ne sont plus des champs booléens mais des champs continus à valeur dans $[0,1]$. Ce n'est que lorsque la température T tend vers 0 (*i.e.* $\theta \rightarrow \infty$) que la sigmoïde devient un échelon rendant ainsi les processus de ligne booléens.

Une extension de cette méthode au cas d'un modèle anisotrope de champs de Markov a été réalisée par Zerubia et Chellappa [ZERU90], [ZERU93].

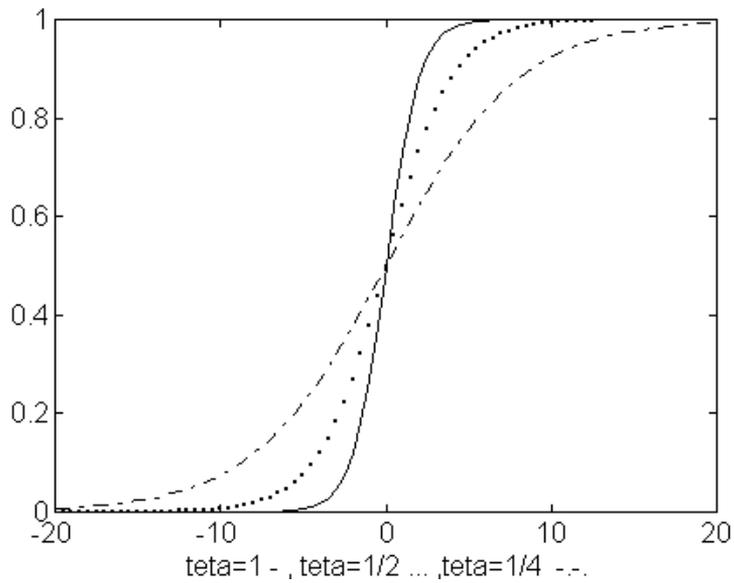


Figure 7.6 : Tracés des sigmoïdes σ_{θ} pour trois valeurs de θ

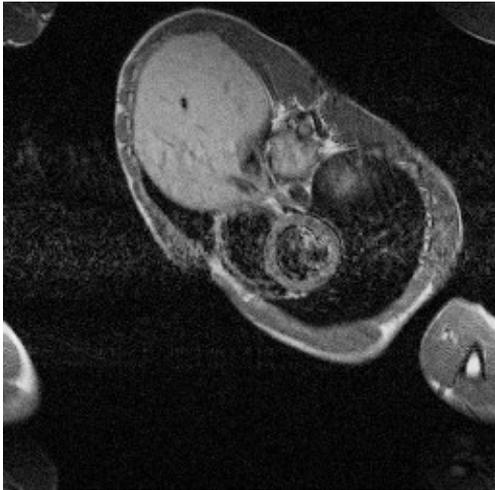


Image originale IRM

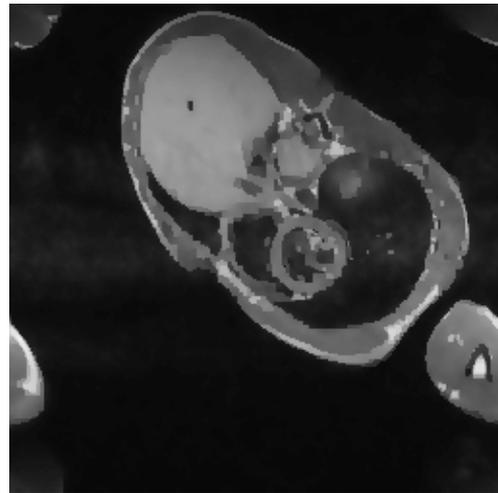


Image restaurée



Contours extraits

Figure 7.7 : Restauration et segmentation par l'algorithme MFA

CHAPITRE VIII

Approche frontière : méthodes variationnelles

Ainsi que cela a été largement décrit au chapitre V, les modèles classiques de détection de contour procèdent essentiellement par application d'un opérateur local de différentiation précédé d'un opérateur de filtrage. Cette opération de filtrage peut soit être associée à un opérateur dont la réponse en fréquence est figée (opérateur de moyenne par exemple, illustré avec les opérateurs de Sobel ou de Prewitt) soit être associée à une opération de filtrage adaptable à l'aide d'un paramètre d'échelle comme c'est le cas avec les approches de Canny et dérivées. De tels opérateurs sont qualifiés d'opérateurs locaux. Ils ne prennent pas en compte l'information globale sur le contour, information pouvant être issue d'un modèle *a priori* de contour. De plus, ils nécessitent une phase de post-traitement afin d'assurer la fermeture de contours.

1 Détection de contours : approches globales

On présente dans ce chapitre un certain nombre de modèles globaux ainsi classés du fait qu'ils prennent en compte dès le départ une notion de structure de contours, à savoir un arc ou une courbe. Parmi ces modèles, on peut distinguer plusieurs classes :

- *Les modèles perceptifs* sont issus de la théorie Gestaltiste invoquant la notion de groupement perceptif. Ces modèles, décrits au chapitre VI, permettent de percevoir des formes organisées dès l'instant où le champ de perception est suffisamment important mettant ainsi en valeur des notions de groupement à différents niveaux de complexité (alignement, coin, parallélisme, positionnement...) [GROS85b], [KASS87], [LIND89]. Ces processus psychovisuels ont fait l'objet de transcription en terme de processus numériques et les critères utilisés sont de type proximité, similarité, co-circularité ou alignement [HARR90], [MOHA89]. Il importe de mentionner des travaux récents utilisant directement la transformation de Gabor mettant en valeur ces phénomènes sans passer par une phase d'analyse [HEIT92].
- *Les modèles structurels* sont fondés sur la prise en compte d'informations *a priori* sur la notion de structure attachée au contour. Cette prise en compte associe le niveau de perception qualifié de bas-niveau au niveau de décision souvent qualifié de haut-niveau. Selon la nature de l'information *a priori* sur le contour, on peut identifier différentes catégories de modèles d'objets : les modèles rigides, paramétriques, élastiques et stochastiques.

- *Les modèles rigides* supposent que l'on dispose d'un modèle précis du contour de l'objet recherché. On utilise alors des méthodes de corrélation ou de mise en correspondance. Leur inconvénient majeur concerne la sensibilité au bruit. Leur domaine essentiel d'application se situe en vision industrielle pour la reconnaissance d'objets manufacturés où les conditions d'éclairage autorisent une visualisation robuste.
- *Les modèles paramétriques* exploitent le fait que les contours des objets sont associés à des courbes décrites par des fonctions simples (droites, arcs de cercles, arcs d'ellipses,...). Parmi les approches classiques on peut citer la transformée de Hough [LEAV93], [MAIT85] procédant par identification des courbes dans l'espace des paramètres. Dans cette même catégorie, on peut citer les méthodes fondées sur l'utilisation de gabarit (*template*) [YUIL92] ou les méthodes évoluant par approximation dans le cas de formes simples comme des cylindres, des superquadriques, des parallélépipèdes [SOLI90].
- *Les modèles élastiques* permettent de se ramener à des formes identifiées par leurs caractéristiques de régularité. Ils exploitent un modèle physique de tension d'une courbe évoluant sous contraintes, issues de l'information image [KASS88]. Parmi eux, on présentera le modèle de contours actifs (ou *snakes*).
- *Les modèles stochastiques* sont liés à une fonction d'énergie exprimable dans un cadre d'estimation bayésienne [TERZ92], [YUIL92]. Dès lors que cette fonction d'énergie peut être associée à une évolution locale du processus, on peut lui attacher un caractère markovien et la minimisation de cette fonction d'énergie s'associe au processus de convergence aboutissant à la solution de stabilité. Les modèles stochastiques généralisent les précédentes approches. Ils ont montré leur intérêt en analyse de séquences d'images [COOT92a], [COOT92b] ainsi que dans la modélisation de formes génériques permettant d'introduire une information *a priori* sur les formes à détecter [GREN91], [RIPL90].

Par la suite, ce chapitre sera consacré à la description du modèle de contours actifs. Vis-à-vis des modèles paramétriques, il permet l'étude d'une variété plus importante de formes. Il réalise un compromis entre la contrainte de régularité de formes et la possibilité de s'autoriser des variations importantes. Par ailleurs, à l'image des modèles stochastiques, il gère un procédé de convergence qui travaille localement mais dont l'implantation est moins coûteuse. La présentation qui est faite par la suite a essentiellement pour but de décrire cette approche dans le contexte de la segmentation d'objets perceptibles par tout ou partie de leur contour.

Le modèle de contours actifs a été introduit par Kass, Witkin et Terzopoulos [KASS88] et son principal domaine d'intérêt concerne la segmentation d'images [MORE94] avec des extensions en mise en correspondance en stéréovision [BASC93], en suivi temporel de contours [BLAK93], [CIPO92], [UEDA92] ainsi qu'en représentation de formes par leur squelette [LEYM93].

2 Modèle de contours actifs (ou snakes)

Le modèle général de contours actifs se présente sous la forme d'une courbe (fermée ou non) dont l'initialisation est située à proximité du contour recherché et dont l'évolution s'effectue selon un processus itératif de déformation contrôlé par un test de convergence (*cf.* figure 8.1). Dans le cas le plus élémentaire, l'initialisation s'effectue par interaction avec l'utilisateur ainsi que cela sera détaillé ultérieurement. La convergence du contour actif vers la frontière recherchée est très généralement vue comme une condition de stabilité mais on peut également la percevoir sous un aspect de rupture de modèle par détection de changement d'état d'évolution du contour au travers des différentes itérations.

Le critère selon lequel le processus de déformation fait évoluer le contour a été initialement associé à la recherche d'une minimisation d'une fonctionnelle directement interprétable en terme de contraintes physiques : raideur, élasticité, attache aux données.

On distingue deux grandes catégories de modèles de contour actif : les modèles continus et les modèles discrets. Les modèles continus s'appuient sur une représentation analytique du contour. Par exemple, durant la phase d'initialisation où l'utilisateur indique quelques points de repère, une approximation de type *spline* peut être réalisée afin de générer une courbe analytique (*cf.* figure 8.2). Cette courbe sera traitée selon une approche variationnelle permettant ainsi de la faire évoluer sous contraintes.



Figure 8.1 : Illustration de l'évolution des sommets d'un contour actif ; le contour actif est initialisé selon une courbe à l'intérieur du visage de la femme et se propage jusqu'à converger vers la limite de la main et du chapeau. La trajectoire de chaque sommet est représentée par un trait blanc

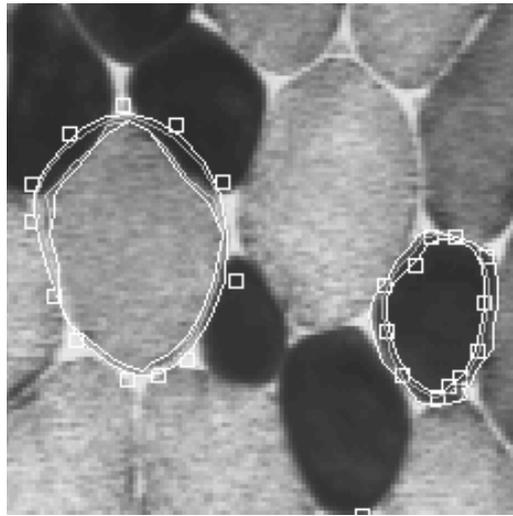


Figure 8.2 : Exemple d'initialisation du contour actif par positionnement de nœuds (indiqués par des carrés) et génération d'une spline d'approximation. Le contour actif se propage ensuite jusqu'à son état final selon un schéma de contraction (objet de gauche) ou un schéma d'expansion (objet de droite).

Le modèle discret quant à lui, associe la courbe initiale à une ligne polygonale dont les sommets sont des points discrets et l'évolution de cette ligne polygonale s'effectue dans le support discret de l'image. Il importe néanmoins de préciser que l'implémentation numérique des modèles continus induit naturellement un processus de discrétisation faisant intervenir un échantillonnage régulier du contour alors que le modèle discret que l'on détaillera par la suite n'impose nullement cette contrainte.

2.1 Présentation du modèle du contour actif

Se fondant sur une représentation paramétrée, le contour actif est assimilé à une courbe C représentée selon les notations suivantes :

$$C = \{v(s,t) = (x(s,t), y(s,t)) ; s \in [a, b] \text{ et } t \in [0, T]\}$$

où a et b désignent les extrémités (fixes, mobiles, confondues ou non) du contour, s désigne l'abscisse curviligne le long du contour, t désigne la variable temporelle et $v(s,t)$ désigne le point courant (*cf.* figure 8.3).

Une énergie est mesurée sur chaque courbe C durant son évolution en fonction de la variable temporelle t . Cette énergie intègre les caractéristiques intrinsèques de la courbe, les caractéristiques de l'image au voisinage de la courbe et enfin l'interaction entre la courbe et l'image. L'objectif visé est de rechercher la position de la courbe qui minimise cette énergie traduisant ainsi une adéquation entre la forme de la courbe et la fonction image au voisinage de cette même courbe.

Cette énergie, notée $E(C)$, s'exprime par :

$$E(C) = E_{\text{interne}}(C) + E_{\text{externe}}(C) + E_{\text{image}}(C) \quad (8-1)$$



Figure 8.3 : Contour actif ouvert à extrémités fixes et contour actif ouvert à extrémités mobiles. Pour chacun on présente la courbe d'initialisation, une phase intermédiaire et le résultat de convergence

Chacune des composantes est précisée par la suite (pour simplifier l'écriture, on sera amené à utiliser la notation $v(s)$ au lieu de $v(s,t)$) :

- L'énergie interne joue le rôle de terme régularisant au sens des problèmes mal posés [POGG85], [DEMO89]. Afin de régulariser la solution, on introduit cette énergie de manière à assurer au contour une certaine continuité (continuité étendue aux dérivées) réduisant au mieux la présence d'oscillations et permettant ainsi d'atténuer les effets du bruit.

Un opérateur de type Tikhonov est introduit selon l'expression :

$$E_{\text{interne}}(C) = \sum_{r=0}^p \int_a^b \alpha_r(s) \left| \frac{\partial^r v(s)}{\partial s^r} \right|^2 ds \quad (8-2)$$

où p désigne l'ordre du stabilisateur et où les coefficients $\alpha_r(s)$ sont des coefficients de pondération qui s'interprètent en terme de caractéristiques physiques de la courbe (continuité, raideur, élasticité...).

Le choix de p est lié à la régularité imposée sur la courbe, régularité de type \mathcal{C}^{2p-3} . Dans le cas particulier où les coefficients $\alpha_r(s)$ sont constants vis à vis de la variable s , la courbe est de classe \mathcal{C}^{2p-2} . L'énergie interne choisie par Kass *et al*, se limite au cas de $p=2$ et devient alors :

$$E_{\text{interne}}(C) = \int_a^b \alpha(s) \left| \frac{\partial v(s)}{\partial s} \right|^2 ds + \int_a^b \beta(s) \left| \frac{\partial^2 v(s)}{\partial s^2} \right|^2 ds \quad (8-3)$$

Le choix de cet ordre de régularité est cependant insuffisant pour extraire des caractéristiques telles que les courbures qui nécessitent en général des

stabilisateurs d'ordre supérieur ou égal à 3, ce qui conduit à la résolution d'une équation aux dérivées partielles d'ordre supérieur ou égal à 6 [COHE93b].

Afin d'éviter de passer par la résolution d'équations du sixième ordre, on préfère aborder une décomposition de la solution dans des bases convenables, comme les B-splines ou toute autre base.

La quantité $\left| \frac{\partial v(s)}{\partial s} \right|^2$ agit sur la longueur de la courbe et est liée à sa rigidité alors que la quantité $\left| \frac{\partial^2 v(s)}{\partial s^2} \right|^2$ agit sur la courbure et est liée à l'élasticité. La courbe doit être suffisamment lisse et rigide sans introduire de boucles.

Les coefficients $\alpha(s)$ et $\beta(s)$ permettent de pondérer différemment les termes liés respectivement à la raideur et à l'élasticité. Le fait d'affecter la valeur 0 au coefficient $\beta(s)$ introduit une discontinuité de première espèce (d'ordre 1). C'est précisément le cas pour les coins, les points anguleux. Le fait d'affecter la valeur 0 aux coefficients $\alpha(s)$ et $\beta(s)$ introduit une discontinuité d'ordre 0. En général, par simplification, on prend $\alpha(s)$ et $\beta(s)$ constants, mais il est intéressant de les rendre localement adaptatifs.

Si l'on veut trouver une analogie du modèle avec des phénomènes physiques, la minimisation du premier terme engendre dans l'équation d'évolution un laplacien analogue à celui d'un faible déplacement de type membrane. La minimisation du second terme engendre dans l'équation d'évolution un bi-laplacien similaire à l'évolution d'une plaque mince pour de petits déplacements.

- *L'énergie externe* sert à introduire, par l'intermédiaire de l'utilisateur, des informations de haut niveau. Ceci se traduit par la présence de forces de répulsions en certains points de la courbe ou par la présence de forces d'élasticité simulant l'allongement d'un ressort entre deux points de la courbe. Les informations de haut niveau peuvent être associées à la présence de frontières préalablement détectées (contraintes) ou de points de contrôle (points de passage obligés).
- *L'énergie image* $E_{\text{image}}(C)$ fait intervenir les caractéristiques image que l'on cherche à mettre en valeur. Dans le cas précis où l'on cherche à mettre en valeur les zones de fort contraste, on peut choisir une énergie image donnée par la relation :

$$E_{\text{image}}(C) = - \int_a^b \left| \nabla^* I(v(s)) \right|^2 ds \quad (8-4)$$

où $\nabla^* I(v(s))$ représente le gradient de l'image I au voisinage de la courbe $v(s)$.

Il existe différentes manières de définir une énergie associée à une image et une formulation générale est donnée par :

$$E_{\text{image}}(C) = - \int_a^b \Psi \left(\left| \nabla^* I(v(s)) \right| \right) ds \quad (8-5)$$

où ψ est une fonction croissante continue dont le choix est lié aux conditions attendues sur la convergence et la stabilité des résultats.

Un exemple classique est celui où les points contours sont définis comme des maximums de gradient et on a la formulation :

$$E_{\text{image}}(C) = - \int_a^b |\nabla(g_\sigma * I(v(s)))|^2 ds \quad (8-6)$$

où ∇ désigne le gradient et g_σ la gaussienne centrée d'écart type σ . Un minimum d'énergie sera atteint si la courbe passe par les points de gradient maximal de l'image lissée par un filtre gaussien.

2.2 Résolution par une approche de type modèle stationnaire

Dans ce paragraphe, on illustre comment la détection de contour est résolue sous forme d'un problème d'optimisation. Supposant que le modèle est stationnaire, le contour représenté par $v(s,t)$ est supposé statique et afin d'alléger l'écriture, $v(s,t)$ sera noté $v(s)$.

Le problème consiste à trouver une courbe C qui réalise le minimum de la fonctionnelle :

$$E(C) = \int_a^b (\alpha(s) \left| \frac{\partial v(s)}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 v(s)}{\partial s^2} \right|^2 - |\nabla I(v(s))|^2) ds \quad (8-7)$$

Pour cela on utilise le théorème suivant, résultat de base en calcul des variations :

Théorème [ELSG62], [GELF63] :

Soit s une variable à valeurs dans $[a, b]$ et soit x une fonction dépendant de la variable s .

Soit $F(s, x, x', x'', \dots, x^{(n)})$ une fonction de classe \mathcal{C}^2 par rapport à tous ses arguments et soit $J(x)$ la fonctionnelle définie par :

$$J(x) = \int_a^b F(s, x, x', x'', \dots, x^{(n)}) ds \quad (8-8)$$

Alors, trouver x minimisant $J(x)$ sous la condition que les extrémités de x et de ses dérivées jusqu'à l'ordre $n-1$ soient fixées, est équivalent à résoudre l'équation d'Euler donnée par :

$$\sum_{m=0}^n (-1)^m \frac{\partial^m F_{x^{(m)}}}{\partial s^m}(s, x, x', x'', \dots, x^{(n)}) = 0 \quad (8-9)$$

où $F_{x^{(m)}}$ désigne la dérivée partielle de F par rapport à m .

Dans notre cas, la variable s joue le rôle de l'abscisse curviligne et la variable x celui de l'une des deux variables $x(s)$ et $y(s)$ qui composent $v(s)$.

Par simplification d'écriture, on notera :

$$\begin{aligned}x' &= d(x(s))/ds & y' &= d(y(s))/ds \\x'' &= d^2(x(s))/ds^2 & y'' &= d^2(y(s))/ds^2 \\v' &= \partial(v(s))/\partial s & v'' &= \partial^2(v(s))/\partial s^2\end{aligned}$$

L'énergie à minimiser est de la forme :

$$E(C) = \int_a^b (\alpha(s)(x'^2 + y'^2) + \beta(s)(x''^2 + y''^2) - |\nabla I(v(s))|^2) ds \quad (8-10)$$

En appliquant le théorème précédent, la minimisation de $E(C)$ se traduit donc par la résolution de l'équation :

$$-(\alpha v')' + (\beta v'')'' = -P(v) \quad (8-11)$$

où $P(v) = \frac{1}{2} |\nabla I(v)|^2$ désigne un potentiel.

Lorsque α et β sont des constantes, l'équation (8-11) devient :

$$-\alpha v'' + \beta v^{(4)} = -P(v) \quad (8-12)$$

En l'absence d'un champ extérieur (cas d'une image uniforme), cette équation s'exprime sous la forme homogène suivante :

$$-\alpha v'' + \beta v^{(4)} = 0 \quad (8-13)$$

Dans le cas où α et β ne sont pas tous deux nuls, une solution particulière de cette équation homogène est donnée au niveau de la composante $x(s)$ par :

- si $\beta = 0$ alors $x(s) = c_1 s + c_2$
- si $\alpha = 0$ alors $x(s) = c_1 s^3 + c_2 s^2 + c_3 s + c_4$
- si $\alpha \neq 0$ et $\beta \neq 0$ alors $x(s) = c_1 e^{\lambda s} + c_2 e^{-\lambda s} + c_3 s + c_4$ avec $\lambda = \sqrt{\frac{\alpha}{\beta}}$.

Les conditions aux limites sont ensuite intégrées pour avoir des courbes fermées ou des courbes ouvertes à extrémités fixes [BERG91].

Un exemple de conditions aux limites permettant d'avoir une courbe à extrémités libres est donné par :

$$v''(a) = v^{(3)}(a) = v''(b) = v^{(3)}(b) = 0 \quad (8-14)$$

2.3 Aspects numériques

La recherche de la courbe réalisant le minimum d'énergie (équation (8-10)) nécessite la résolution de l'équation aux dérivées partielles (8-12) et ceci passe par une phase de discrétisation du problème à l'aide des différences finies.

On considère la courbe discrétisée $\{v_i = (x_i, y_i), i = 0, \dots, N-1\}$ dont les points sont équidistants d'une distance h .

Au point v_i , les dérivées première et seconde par rapport à x de la composante x du vecteur v sont respectivement approchées par $\frac{x_i - x_{i-1}}{h}$ et $\frac{x_{i+1} - 2x_i + x_{i-1}}{h}$ (de même pour y).

Supposant $\alpha_i = \alpha$ et $\beta_i = \beta$ pour toute valeur de i , l'équation $-(\alpha x)' + (\beta x)'' = f$ devient :

$$\beta x_{i-2} + (-4\beta - h^2\alpha) x_{i-1} + (6\beta + 2h^2\alpha) x_i + (-4\beta - h^2\alpha) x_{i+1} + \beta x_{i+2} = h^4 f_i \quad (8-15)$$

On obtient un système linéaire de la forme générale $\mathbf{AX} = \mathbf{F}$ où

$$\mathbf{X} = (x_i)_{i=0, \dots, N-1}$$

$$\mathbf{F} = (h^4 f_i)_{i=0, \dots, N-1}$$

et \mathbf{A} est une matrice définie ci-dessous.

Dans le cas des *snakes* fermés, compte tenu du caractère périodique des courbes, la matrice \mathbf{A} est circulante et s'exprime sous la forme (on prend le pas d'échantillonnage h égal à 1) :

$$\begin{array}{ccccccc} 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & \dots & \beta & -\alpha-4\beta \\ -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & & 0 & \beta \\ \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & & & 0 \\ 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & & \dots & \dots \\ \dots & & & & \dots & \beta & 0 \\ 0 & & & & \dots & -\alpha-4\beta & \beta \\ \beta & 0 & & \dots & & 2\alpha+6\beta & -\alpha-4\beta \\ -\alpha-4\beta & \beta & 0 & \dots & \dots & -\alpha-4\beta & 2\alpha+6\beta \end{array}$$

Dans le cas des *snakes* ouverts à extrémités libres, sous les conditions aux limites précédemment évoquées en (8.14), la matrice \mathbf{A} proposée par Berger [BERG91] s'écrit :

$$\begin{array}{ccccccc} \beta & -2\beta & \beta & 0 & 0 & 0 & 0 \\ -\alpha-2\beta & 2\alpha+5\beta & -\alpha-4\beta & \beta & 0 & 0 & 0 \\ \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & 0 & 0 \\ 0 & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \beta & -\alpha-4\beta & 2\alpha+6\beta & -\alpha-4\beta & \beta \\ 0 & \dots & 0 & \beta & -\alpha-4\beta & 2\alpha+5\beta & -\alpha-2\beta \\ 0 & 0 & \dots & 0 & \beta & -2\beta & \beta \end{array}$$

Pour le cas des *snakes* fermés, Berger a montré que les valeurs propres de la matrice \mathbf{A} s'expriment par :

$$\lambda_k = \frac{4}{h^2} \sin^2 \frac{k\pi}{N} (4\beta \sin^2 \frac{k\pi}{N} + h^2\alpha) \quad \forall k \in \{0, \dots, N-1\} \quad (8-16)$$

Donc $\lambda_0=0$ est valeur propre, ce qui rend la matrice \mathbf{A} non inversible.

Il est également facile de voir que la matrice \mathbf{A} associée au cas de *snakes* à extrémités libres est également singulière. En effet, 0 est valeur propre associée au vecteur propre $(1, \dots, 1)$.

Ceci rejoint la démarche de Kass et de ses collaborateurs [KASS 88] qui avaient perçu la singularité en proposant de résoudre le problème selon le schéma itératif :

$$\mathbf{A} \mathbf{v}^{t+1} = \mathbf{F} + \theta (\mathbf{v}^t - \mathbf{v}^{t+1}) \quad \text{où } \theta \text{ est un réel strictement positif.}$$

Le système devient alors :

$$(\mathbf{A} + \theta \mathbf{I}) \mathbf{v}^{t+1} = \mathbf{F} + \theta \mathbf{v}^t \quad (8-17)$$

où \mathbf{I} est la matrice unité de dimension $N \times N$.

Pour les *snakes* fermés, la matrice $(\mathbf{A} + \theta \mathbf{I})$ étant réelle et symétrique est inversible, ses valeurs propres sont positives et θ est la plus petite valeur propre.

Le choix de la valeur θ est lié au conditionnement de la matrice à inverser $(\mathbf{A} + \theta \mathbf{I})$.

En effet on rappelle que le conditionnement d'une matrice \mathbf{B} inversible est égal au rapport entre la plus grande et la plus petite des valeurs singulières de \mathbf{B} (racines carrées positives des valeurs propres de $\mathbf{B}^* \mathbf{B}$ où \mathbf{B}^* est la matrice hermitienne associée à \mathbf{B}).

Dans notre cas, le conditionnement de la matrice $(\mathbf{A} + \theta \mathbf{I})$ est égal à $(\rho + \theta)/\theta$ où ρ désigne la plus grande valeur propre de la matrice \mathbf{A} , donnée par :

$$\rho = \frac{4}{h^2} \sin^2 \frac{k\pi}{N} (4\beta \sin^2 \frac{k\pi}{N} + h^2\alpha) \quad \text{avec } k = \text{partie entière de } N/2 \quad (8-18)$$

Sachant qu'une matrice est bien conditionnée si son conditionnement est proche de 1 par valeurs supérieures, il faut choisir θ très grand et ρ très petit. On reviendra ultérieurement sur ce choix en fonction du pas temporel Δt suite à la présentation du modèle évolutif.

2.4 Modèle évolutif et formulation variationnelle

Du fait que l'énergie à minimiser (équation 8-7) n'est pas convexe, on ne peut être certain d'accéder à un minimum global et c'est pour cette raison qu'il importe de se situer près de l'objet à segmenter afin de s'assurer que le minimum local que l'on percevra correspondra au contour recherché.

Cette considération a amené certains auteurs à transposer l'équation précédente stationnaire (Eq 8-11) en une équation évolutive selon une variable temps que l'on assimilera aux itérations [COHE93]. Représentant v sous la forme d'une fonction de deux variables s et t , cette équation prend alors la forme du système suivant (en reprenant les notations de l'équation 8-11) :

$$\left\{ \begin{array}{l} \frac{\partial v}{\partial t} - \frac{\partial(\alpha v')}{\partial s} + \frac{\partial^2(\beta v'')}{\partial s^2} = f(v) \\ \text{conditions initiales} \\ \text{conditions aux limites} \end{array} \right. \quad (8-19)$$

Avant d'aborder la résolution de cette équation, on revient un instant au modèle stationnaire. Sa résolution a pu être abordée en introduisant un procédé itératif donné par :

$$\mathbf{A} v^{t+1} = \mathbf{F} + \theta (v^t - v^{t+1}) \quad (8-20)$$

Ceci s'écrit encore :

$$\theta (v^{t+1} - v^t) + \mathbf{A} v^{t+1} = \mathbf{F} \quad (8-21)$$

Si on assimile θ à la valeur $1/\Delta t$ où t est une variable temporelle, on a :

$$(v^{t+1} - v^t)/\Delta t + \mathbf{A} v^{t+1} = \mathbf{F} \quad (8-22)$$

On retrouve la condition déjà évoquée sur θ (cf. § 2.3) qui s'assimile au choix d'un Δt faible. Ceci amène directement à la formulation variationnelle évoquée dans ce paragraphe.

La fonction f n'étant pas linéaire, la résolution de l'équation (8-15) s'effectue de manière itérative en considérant la quantité f non plus comme une fonction de v mais comme une fonction de s et de t [COHE91]. On obtient la formulation :

$$\left\{ \begin{array}{l} \frac{\partial v}{\partial t} - \frac{\partial(\alpha v')}{\partial s} + \frac{\partial^2(\beta v'')}{\partial s^2} = f(s, t) \\ \text{conditions initiales} \\ \text{conditions aux limites} \end{array} \right. \quad (8-23)$$

On va maintenant détailler les différents types de *snakes* : *snakes* à extrémités fixes, *snakes* à extrémités libres et *snakes* fermés.

Dans le cas des *snakes* à extrémités fixes, les dérivées aux extrémités sont supposées constantes au premier ordre. Par souci de simplification, on prend la valeur 0 comme valeur des dérivées aux extrémités.

Dans le cas des *snakes* à extrémités libres, aucune contrainte n'est imposée sur la position des extrémités mais on choisira des conditions aux limites annulant les dérivées seconde et troisième pour contraindre le déplacement des extrémités.

Quant aux *snakes* fermés, ils sont représentés par des courbes fermées, qui induisent des conditions de périodicité.

Compte tenu de la séparabilité de l'équation à résoudre, v désignera indifféremment la composante x ou la composante y de v .

En notant $\langle ., . \rangle$ le produit scalaire dans $L^2(]0,1[)$ et en introduisant la variable u on a :

$$\frac{d}{dt} \langle v, u \rangle + \int_0^1 (\alpha v' u' + \beta v'' u'') ds - [\beta v'' u']_0^1 - [\alpha v' u]_0^1 + [(\beta v'')' u]_0^1 = \langle f, u \rangle \quad (8-24)$$

On note $a(v, u)$ la forme bilinéaire définie par :

$$a(v, u) = \int_0^1 (\alpha v' u' + \beta v'' u'') ds \quad (8-25)$$

On se ramène à la recherche d'un espace V satisfaisant aux conditions initiales et dans lequel, on doit résoudre le problème variationnel suivant :

$$\text{déterminer } v : \begin{cases} v: [0, T] \rightarrow V \\ t \mapsto v(t) \end{cases} \quad \text{tel que} \quad (8-26)$$

$$\forall u \in V, \frac{d}{dt} \langle v(t), u \rangle + a(v(t), u) = \langle f(t), u \rangle$$

où $[0, T]$ désigne l'intervalle de temps où l'on cherche la solution.

La donnée initiale est prise dans $L^2(]0,1[)$ tandis que la fonction f est dans $L^2([0, T]) ; L^2(]0,1[)$, espace des fonctions de carré intégrable sur $[0, T]$ à valeurs dans $L^2(]0,1[)$. f est en fait une fonctionnelle qui dépend du paramètre temps t et de la courbe $v(s, t)$; $v(s, t)$ étant elle-même élément de $L^2(]0,1[)$.

Le choix de V dépend du type de *snake* choisi et plus précisément des conditions aux limites imposées. On note $H^n(]0,1[)$ l'espace (de Sobolev) des (classes de) fonctions de $L^n(]0,1[)$ dont les dérivées (au sens des distributions) jusqu'à l'ordre n appartiennent à $L^n(]0,1[)$.

Dans le cas des *snakes* à extrémités fixes, on prend pour V l'espace $H_0^2(]0,1[)$, des fonctions de $H^2(]0,1[)$ s'annulant sur les bords (0 et 1) ainsi que leurs dérivées premières.

Dans le cas des *snakes* fermés on prend pour V l'espace $H_p^2(]0,1[)$, espace des fonctions de $L^2(]0,1[)$ périodiques sur $]0,1[$ ainsi que leurs dérivées jusqu'à l'ordre 2.

Dans le cas des *snakes* à extrémités libres, on a le théorème suivant [ELOM94] :

Théorème :

Soit $V = \{u \in \square H^4(]0,1[), u'' \in \square H_0^2(]0,1[) \}$, pour les *snakes* à extrémités libres, si α s'annule au bord et est minorée par une constante positive, alors il existe une solution unique v du problème (8-26) vérifiant :

$$v \in L^2([0, T], V) \cap C^0([0, T], L^2(]0,1[))$$

2.5 Approche numérique et évolution de la courbe

Partant de la courbe initialisée au voisinage de la frontière cherchée, on laisse évoluer la courbe au cours des itérations jusqu'à coïncidence avec le contour de l'objet. Ceci rejoint le schéma proposé par Kass en prenant $\theta = 1/\Delta t$ où Δt est associé au pas d'itération.

La résolution numérique du problème posé en 8-26 peut s'effectuer soit par différences finies, soit par éléments finis [COHE90].

La résolution par différences finies de (8-24) peut être envisagée selon deux schémas.

Un premier schéma implicite aboutit à une expression de la forme :

$$h^4 \frac{v_i^{t+\Delta t} - v_i^t}{\Delta t} + \beta v_{i-2}^{t+\Delta t} - (4\beta + h^2 \alpha) v_{i-1}^{t+\Delta t} + (5\beta + 2h^2 \alpha) v_i^{t+\Delta t} - (4\beta + h^2 \alpha) v_{i+1}^{t+\Delta t} + \beta v_{i+2}^{t+\Delta t} = h^4 f_i^t \quad (8-27)$$

Un second schéma explicite aboutit à une expression de la forme :

$$h^4 \frac{v_i^{t+\Delta t} - v_i^t}{\Delta t} + \beta v_{i-2}^t - (4\beta + h^2 \alpha) v_{i-1}^t + (5\beta + 2h^2 \alpha) v_i^t - (4\beta + h^2 \alpha) v_{i+1}^t + \beta v_{i+2}^t = h^4 f_i^t \quad (8-28)$$

Dans ces deux schémas, on a supposé que les coefficients α et β étaient constants le long de la courbe. On suppose également que le pas de discrétisation h est constant pour toutes les itérations. Compte tenu de l'évolution des points de la courbe, ceux-ci ne restent pas à distance constante et il faudrait en toute rigueur procéder à une nouvelle discrétisation de la courbe après chaque itération, ce qui se traduit par un changement de la taille de la matrice du système linéaire à résoudre.

Dans le cas du schéma implicite, ceci conduit à une inversion de matrice pour calculer $v^{t+\Delta t}$ à partir de v^t alors que le schéma explicite présente l'avantage de ne

pas nécessiter une telle phase d'inversion car il fournit directement les nouveaux points $v_i^{t+\Delta t}$ en fonction des anciens points v_i^t permettant ainsi la mise à jour de la discrétisation de la courbe afin de préserver l'équidistance. Une telle phase de mise à jour a pour avantage de prendre en compte les phénomènes d'accumulation des points de la courbe dans les zones à fort gradient et ainsi de redistribuer les points le long de la courbe globale.

La figure 8.4 illustre les résultats de convergence d'un processus de contour actif de type fermé sur différentes fibres de l'image MUSCLE. A chaque objet, l'initialisation a été effectuée manuellement avec 8 points et les paramètres de rigidité et d'élasticité sont identiques pour tous les objets.

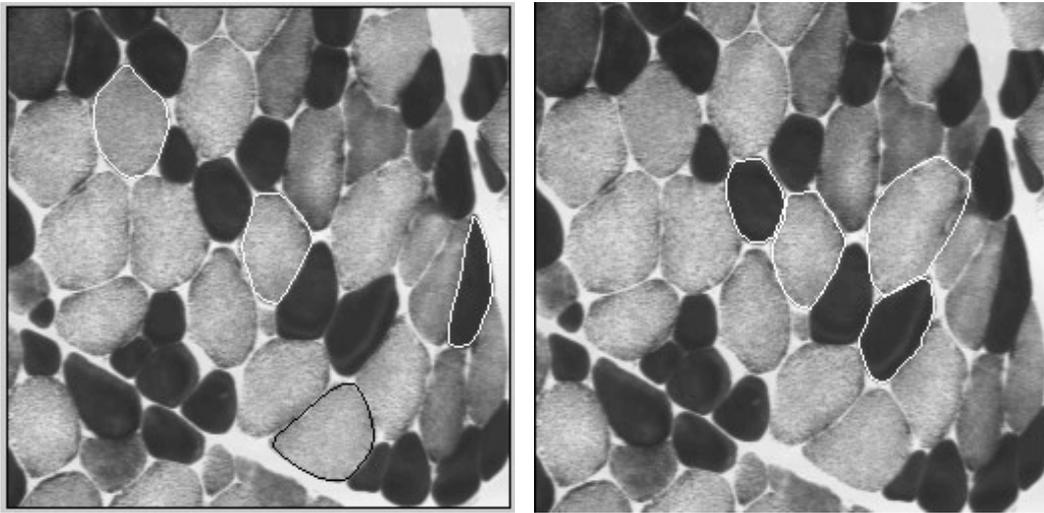


Figure 8.4 : Résultat de segmentation par contour actif fermé

3 Modèle discret de la bulle

Les modèles précédemment évoqués se fondent sur une formulation continue du problème, suivie d'une phase de discrétisation, afin d'accéder à une résolution numérique. Compte tenu des problèmes rencontrés, notamment sur la nécessité de maintenir une discrétisation régulière imposée par la phase de discrétisation, on peut envisager une approche purement discrète où le contour est identifié par une suite de points discrets représentant les sommets du contour associé à une ligne polygonale [ELOM94].

Le principe d'évolution consiste à construire une suite de contours déformés selon un critère de mesure d'énergie locale. Cette suite est déterminée de manière à ce que le contour épouse au mieux l'information image. A chaque étape, la mesure de la fonction d'énergie prend en compte la force de rétention de l'image sur le contour et la déformation locale du contour en terme de courbure. Un contrôle de l'évolution de cette fonction d'énergie informe sur la présence d'un minimum correspondant à l'étape optimale d'adéquation entre le contour et l'image à segmenter. Un tel modèle s'associe à l'évolution d'une bulle, évolution qui se

matérialise par une croissance de la bulle à laquelle on associe une fonction d'énergie. Le test d'arrêt correspond à une rupture du modèle et se matérialise par un éclatement de la bulle dont le contour peut alors de nouveau se propager librement au delà de la frontière recherchée (cf. figure 8.8).

Sur le plan de la représentation du modèle, le contour est associé à un polygone donné sous forme d'une liste chaînée de sommets P_i . Chaque sommet P_i est identifié par ses coordonnées (x_i, y_i) .

Au cours des itérations, chaque sommet P_i a la possibilité d'évoluer vers une nouvelle position située à une distance fixe de l'ancienne position et selon une direction orthogonale à la direction donnée par le segment $P_{i-1} P_{i+1}$ (cf. figure 8.5).

Par ailleurs, à chaque sommet on associe la mesure de distance au sommet précédent, la mesure de distance au sommet suivant et une mesure de courbure locale donnée par :

$$\text{courb}(P_i) = \text{Mesure_angle}(P_{i+1}P_i, P_i P_{i-1}) / (|P_{i+1} - P_i| - |P_i - P_{i-1}|) \quad (8-29)$$

Sur le plan de l'énergie, la fonction d'énergie se décompose en une somme d'une fonction d'énergie interne notée E_{int} associée à la courbure et d'une fonction d'énergie externe notée E_{ext} . Pour chaque sommet, la fonction d'énergie externe est associée à la mesure de gradient entre la valeur image au sommet considéré et la moyenne des valeurs image des points situés dans une fenêtre d'interaction construite à partir des sommets P_{i-1}, P_i, P_{i+1} après déplacement (cf. figure 8-5).

La mesure de gradient est évaluée de différentes façons permettant d'adapter le modèle au type de frontières présentes dans l'image. Ainsi, on peut avoir des frontières de type double échelon ou des frontières de type échelon simple selon une amplitude plus ou moins prononcée. La valeur globale d'énergie en chaque point intègre les deux mesures précédentes pondérées par le paramètre α qui précise l'importance donnée à la courbure vis-à-vis de l'information de gradient image, ceci selon la formule :

$$E(C) = E_{\text{int}}(C) + E_{\text{ext}} \quad (8-30)$$

La phase d'initialisation s'effectue par un positionnement manuel d'un polygone régulier à l'intérieur de la région à segmenter. Ensuite le passage d'un itéré au suivant consiste à rechercher le sommet d'énergie minimale et à le faire évoluer vers sa nouvelle position (cf. figure 8.5). Le nouveau positionnement de ce sommet peut donner naissance à un nouveau sommet du contour, si la distance inter-sommets est supérieure à un seuil préfixé. Ce seuil et cette possibilité d'insérer un nouveau sommet est à rapprocher de la notion de raideur dans les modèles de snakes impliquant le paramètre .

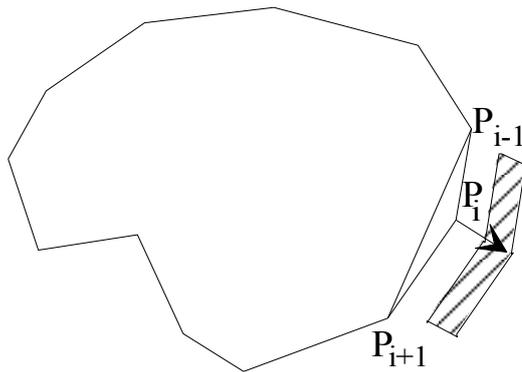


Figure 8.5 : Représentation polygonale du modèle de bulle et illustration de l'évolution du sommet courant P_i (indiquée par la flèche) ; en gris, la fenêtre d'interaction

Par ailleurs, la distance entre l'ancienne position du sommet et la nouvelle position est majorée par une valeur seuil traduisant le concept d'élasticité associé au paramètre β dans le cas des *snakes*. Ce paramètre s'associe à une notion de vitesse de propagation.

A chaque étape on réévalue la valeur de la fonction d'énergie. Cette fonction d'énergie évolue selon trois configurations possibles : libre croissance de la bulle, présence locale d'une rétention de l'image donnant lieu à une extension contrainte de la bulle, éclatement de la bulle. Ces différentes configurations sont illustrées en figure 8.6 permettant de comprendre l'allure de la fonction d'énergie. L'extremum global de cette fonction d'énergie correspond à la position où le modèle présente une rupture interprétée comme éclatement. Le contour retenu est associé à la ligne polygonale à cette étape.

On présente en figure 8.7 un exemple illustrant différentes étapes ainsi que la représentation de la courbe d'énergie en fonction des itérations.

Au niveau des paramètres disponibles, on intervient sur le paramètre de vitesse de propagation, le paramètre de raideur et le paramètre d'élasticité. Adapter ces paramètres relève d'une expertise en ce sens que l'utilisateur doit posséder des informations *a priori* sur le type d'objets qu'il veut segmenter. Les informations concernent la régularité du contour et la visibilité des frontières (contraste) ainsi que leur nature (épaisseur).

En figure 8.8, on présente un résultat illustrant la méthode avec un paramètre d'élasticité très faible permettant d'avoir une bulle très déformable. L'initialisation se fait à l'aide d'un cercle représenté par une ligne polygonale régulière. Les traits blancs montrent l'évolution des sommets P_i .

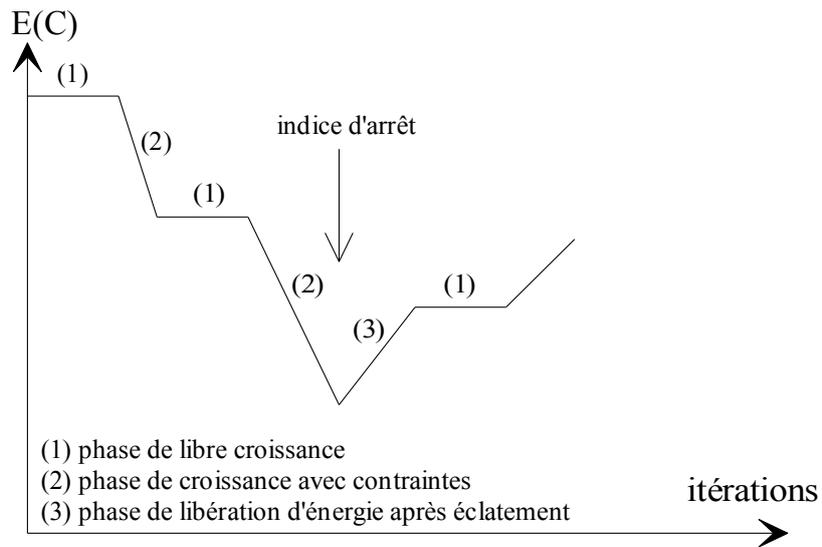
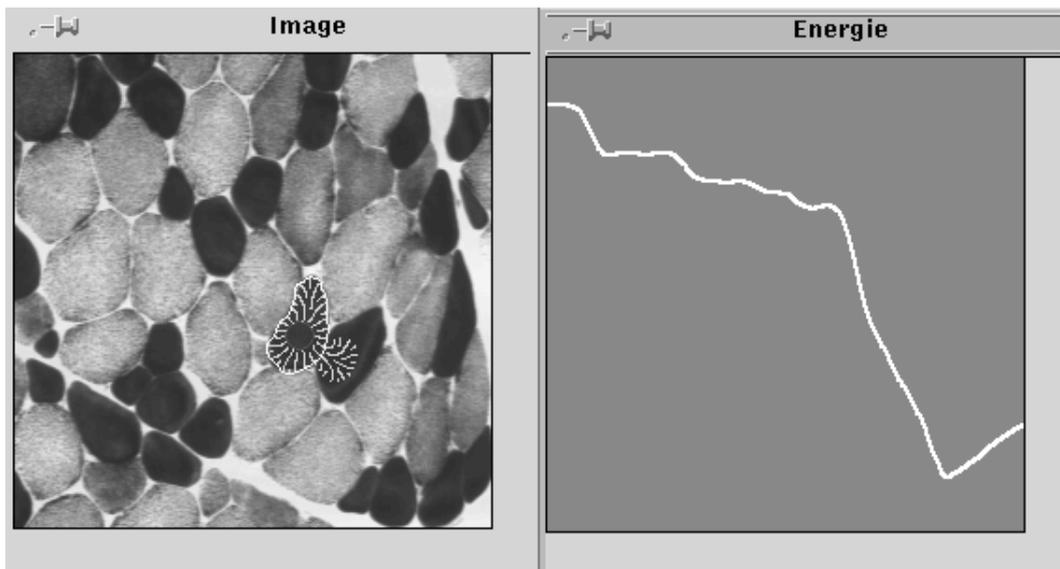


Figure 8.6 : Illustration de l'évolution de la fonction d'énergie au cours des itérations.



a)

b)

a) image initiale, évolution de la bulle et résultat du contour détecté
 b) évolution de la fonction d'énergie au cours des itérations. L'indice associé au minimum correspond à l'itération retenue qui fournit alors le contour détecté et représenté au a)

Figure 8.7 : Exemples d'évolution de la bulle et de représentation de la fonction d'énergie associée.

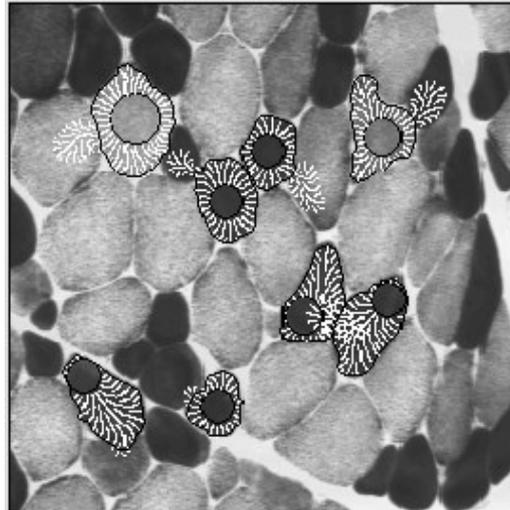


Figure 8.8 : Différents exemples d'application de l'algorithme bulle ;
les contours sont représentés en noir

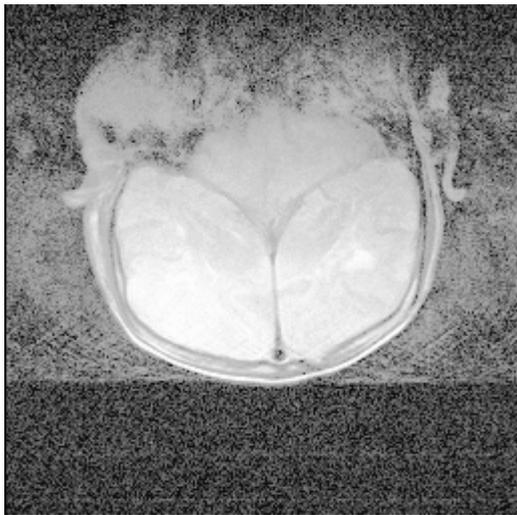
La figure 8.9 montre les potentialités de la méthode dans le cas de frontières faiblement contrastées. L'exemple est emprunté à l'imagerie cérébrale.

4 Conclusion

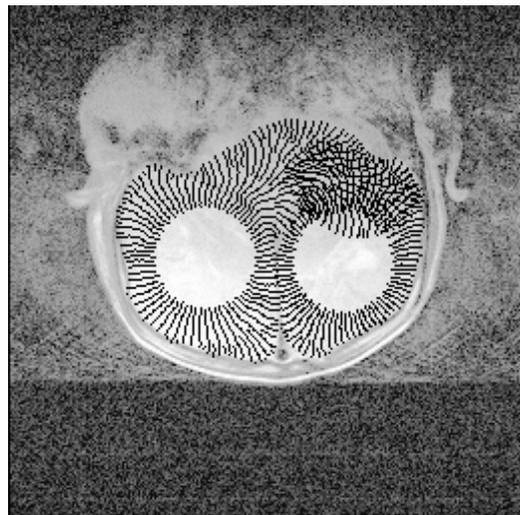
Ce chapitre fournit un aperçu des possibilités des méthodes variationnelles par contours déformables. Ces méthodes s'adaptent très bien à la segmentation des structures dans une image. Un des inconvénients majeurs réside dans la phase d'initialisation manuelle malgré l'importance des travaux effectués pour pallier ce handicap.

Par ailleurs, il apparaît une réelle sensibilité au choix des paramètres de rigidité et d'élasticité. Cependant l'intégration d'une telle approche avec une analyse multi-résolution permet de s'affranchir de cette faiblesse de robustesse [CHAS95], [BOSS94].

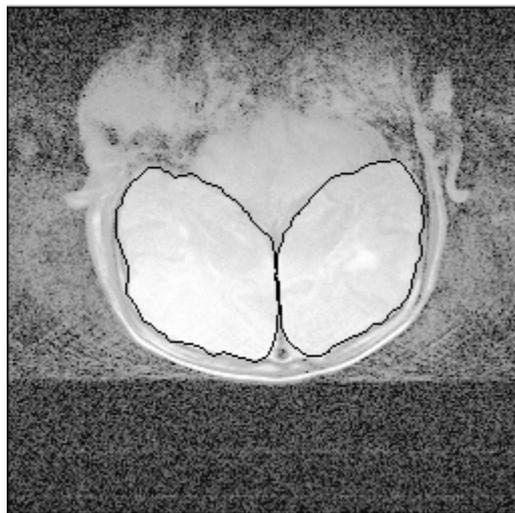
Deux approches d'esprits différents ont été proposées : une méthode analytique et une méthode discrète. L'avantage de la méthode analytique réside dans son support théorique. La solution apparaît alors comme un état de convergence. L'avantage de la méthode discrète réside dans sa simplicité d'implantation. La solution se caractérise comme un changement d'état lié à un minimum d'énergie.



a) image initiale



b) évolution des sommets à partir des 2 cercles du centre



c) résultat de la segmentation

Figure 8.9 : Application de l'algorithme bulle en imagerie cérébrale.

Comparaison des extracteurs de contour

Parmi les méthodes d'extraction de contour présentées dans les chapitres V et VII, un certain nombre a été implanté et testé, sur des images de synthèse et sur des images réelles, comportant les caractéristiques habituellement rencontrées en analyse d'images : images bruitées, texturées, *etc.* Les méthodes testées sont celles qui nous semblaient les plus intéressantes au vue des résultats obtenus. Il s'agit de quatre méthodes de l'approche dérivative, deux méthodes utilisant la modélisation markovienne et une méthode employant la morphologie mathématique. Les algorithmes correspondants ont tous été implantés par l'ETIS^v (ENSEA), sauf ceux exploitant les champs de Markov, qui ont été implantés à l'INRIA^{vi} (Sophia Antipolis). Certains de ces algorithmes ont été intégrés à la banque de logiciels du GDR 134 (TDSI^{vii}) du CNRS, SIMPA^{viii}.

Les images de synthèse comportent une seule transition, dont la position est connue ; il est donc possible de définir des critères quantitatifs pour comparer les résultats (*cf.* § 2.3). Par contre, cette démarche est impossible pour des images réelles (*cf.* § 3.3), la comparaison est donc subjective et dépend des objectifs que s'est fixé l'expérimentateur qui a réglé les paramètres des opérateurs. Ces images sont extraites de la banque d'images du GDR 134. Quelques indications concernant les temps de calcul sont également fournies.

1 Description des méthodes d'extraction de contour testées

Toutes les méthodes mises en œuvre intègrent un prétraitement de réduction du bruit. La méthode morphologique testée est une suite de traitements morphologiques contenant également un filtrage du bruit.

^vEquipe Traitement des Images et du Signal

^{vi}Institut National de Recherche en Informatique et Automatique

^{vii}Traitement du Signal et Images

^{viii}Signal and Image software PAckages

Les méthodes suivantes ont été testées :

- Deriche (chapitre V § 1.6.3)
- Shen-Castan (chapitre V § 1.6.5)
- MDIF (chapitre V § 1.4.4)
- NAGDIF (chapitre V § 1.4.4)
- Recuit par champs moyens ou MFA (chapitre VII § 3.2)
- *Graduated Non Convexity* ou GNC (chapitre VII § 3.1)
- Méthode morphologique basée sur un calcul de gradient morphologique (chapitre V § 3.1)

Pour chacune des méthodes de Deriche et Shen-Castan, trois opérateurs ont été comparés, ils suivent la démarche générale présentée dans le chapitre V (§ 1.2) et utilisent le seuillage par hystérésis décrit au chapitre V (§ 4.2) :

Deriche1 et Shen1 :

- calcul de l'image A_1 de la norme du vecteur gradient en chaque point
- élaboration de l'image A_2 qui correspond à l'image A_1 dans laquelle n'ont été conservés que les maximums locaux suivant la direction du gradient (affectés de la norme du gradient), les autres pixels étant affectés de la valeur zéro
- seuillage par hystérésis de l'image A_2 .

Deriche2 et Shen 2 :

- calcul de l'image A_1 de la norme du vecteur gradient en chaque point
- calcul de l'image A_2 du laplacien en chaque point
- élaboration de l'image A_3 telle que tout pixel qui est un passage par zéro dans A_2 est affecté de la norme du gradient, les autres pixels étant affectés de la valeur zéro
- seuillage par hystérésis de l'image A_3 .

Deriche3 et Shen3 :

- calcul de l'image A_1 de la norme du vecteur gradient en chaque point
- calcul de l'image A_2 de la dérivée seconde dans la direction du vecteur gradient en chaque point
- élaboration de l'image A_3 telle que tout pixel qui est un passage par zéro dans A_2 est affecté de la norme du gradient, les autres pixels étant affectés de la valeur zéro

- seuillage par hystérésis de l'image A_3 .

Pour les autres méthodes :

MDIF :

- calcul de la norme du gradient par l'opérateur MDIF
- seuillage

NAGDIF :

- calcul de la norme du gradient par l'opérateur NAGDIF
- seuillage

MFA : *algorithme déterministe de relaxation*

- résolution de façon itérative d'un système d'équations non-linéaires faisant intervenir les moyennes approximées des niveaux de gris et des processus de ligne.
- recuit en température

GNC : *algorithme déterministe de relaxation*

- approximation convexe U^* de l'énergie à minimiser U
- construction grâce à un paramètre de contrôle p d'une séquence d'énergies non convexes $U^{(p)}$ se rapprochant de plus en plus de l'énergie à minimiser U .
- extraction des contours par calcul des processus de ligne après convergence de l'algorithme (*i.e.* lorsque le minimum de U a été obtenu)

Morpho : *enchaînement de traitements de Morphologie Mathématique* :

- filtre alterné séquentiel
- calcul du gradient morphologique
- amincissement jusqu'à stabilisation de l'image de la norme du gradient
- seuillage

2 Comparaison sur images de synthèse

2.1 Génération des images de test

Les performances des opérateurs ont été évaluées sur des images de synthèse comportant une seule transition. Ces images ont une taille de 64 sur 64 pixels ; elles comportent les motifs suivants :

a) une transition verticale située entre les colonnes 31 et 32 (colonnes numérotées de 0 à 63).

Quatre images ont été fabriquées pour quatre valeurs de saut d'amplitude 3, 6, 12 et 20 niveaux de gris, l'intensité de la partie gauche est de 20.

b) une transition oblique (60°), passant par le point (32, 32).

Quatre images ont été fabriquées avec les mêmes valeurs de saut d'amplitude que précédemment et la même intensité pour la partie gauche.

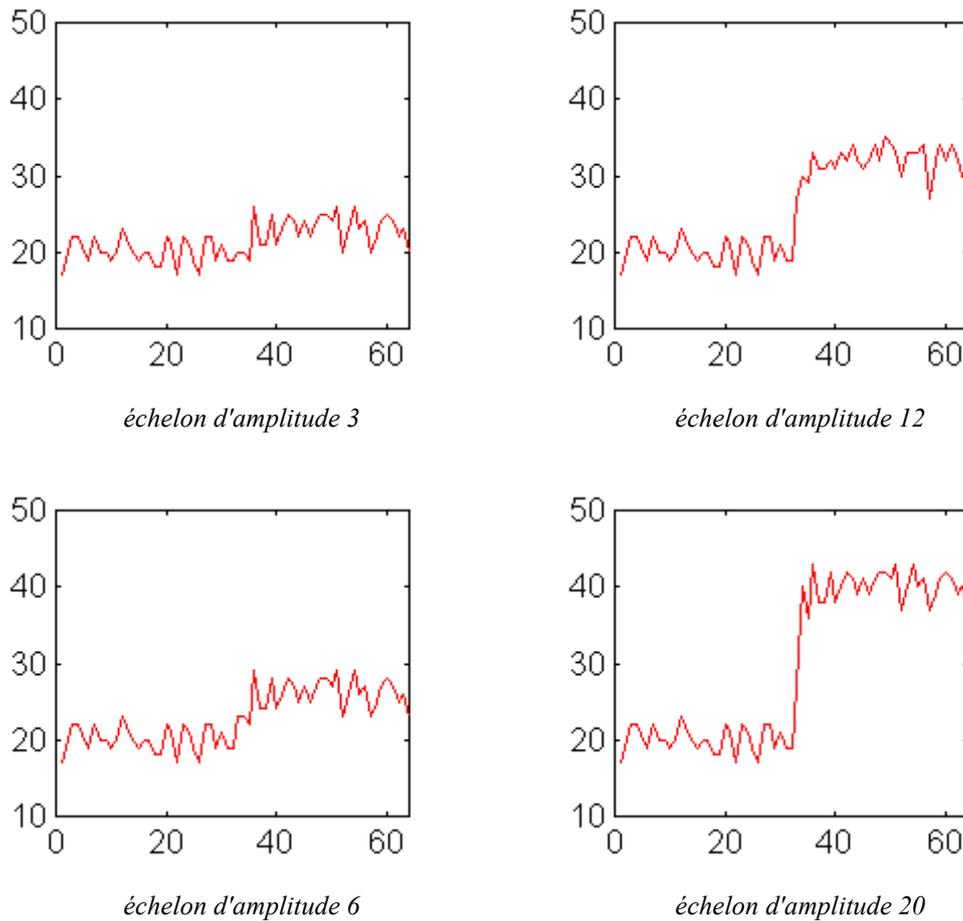


Figure 9.1 : Profil d'une ligne de chacune des images de test

Ces images ont ensuite subi des dégradations réalistes proches de celles apportées par une chaîne réelle d'acquisition d'images.

Cette chaîne comprend un système linéaire caractérisé par sa réponse impulsionnelle $h(x,y)$ qui est paire. Pour la simulation, le signal d'origine, à valeurs réelles $A(x,y)$ est approché par un signal $A_e[i,j]$ suréchantillonné à 10 fois le pas d'échantillonnage normal. La fonction de transfert de la chaîne est approchée par la réponse impulsionnelle finie $h_f(x,y)$, l'image obtenue est notée B_e :

$$B_e[i,j] = A_e * h_f[i,j].$$

Après application d'une loi de quantification uniforme, le signal est rééchantillonné avec le pas d'échantillonnage initial ce qui conduit à l'image $B[i,j]$.

Cette image numérique subit ensuite des dégradations, simulant une transmission, obtenues lors d'une compression par un codage différentiel unidirectionnel. Enfin un bruit gaussien d'écart-type égal à deux niveaux de gris est ajouté.

Les images obtenues pour les sauts d'amplitude 3 et 6 niveaux sont présentées sur les figures 9.4 à 9.7. Leur dynamique a été augmentée à l'affichage, pour une meilleure lisibilité. La figure 9.1 représente un profil (horizontal) de chacune des images de la transition verticale (le saut se situe en abscisse 32) ; on notera le niveau de difficulté des images comportant le saut d'amplitude 3 !

2.2 Définitions des critères de comparaison

Ces images sont utilisées pour une évaluation quantitative des performances des opérateurs d'extraction de contour, en employant les critères de Fram et Deutsch [FRAM75] [FRAM78] modifiés [COCQ85]. Ces critères ne s'appliquent que sur des images comportant une seule transition.

Les performances d'un opérateur sont quantifiées par les valeurs de deux paramètres P_1 et P_2 . Le premier mesure la robustesse du détecteur en présence de bruit, et le second caractérise le comportement du détecteur vis-à-vis du signal seul (c'est-à-dire du contour).

Dans le cas d'une transition non verticale, le contour théorique est la droite d'équation (cf. figure 9.2) :

$$y = -a(x-x_0) + y_0 \quad \text{avec } a = \text{tg}\theta \quad (9-1)$$

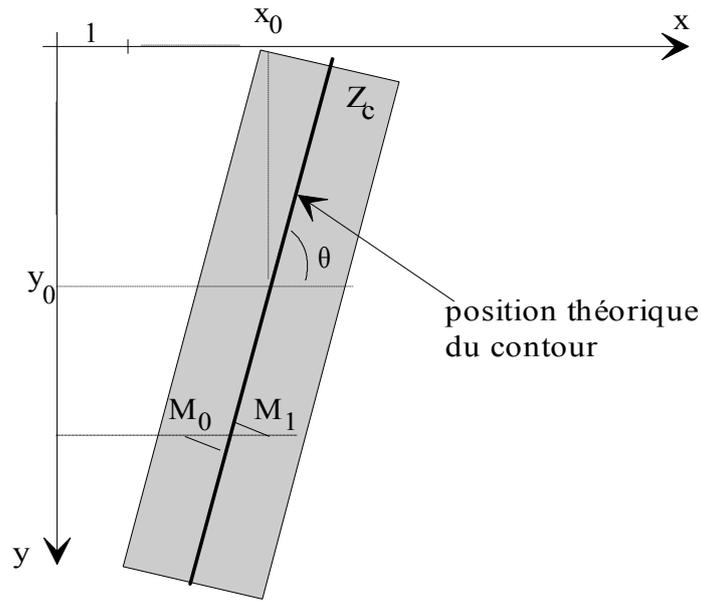


Figure 9.2 : Définition de la zone de transition Z_c (zone grisée)

La zone contour Z_c est limitée, en théorie, par les deux droites situées de part et d'autre du contour théorique, à une distance égale au pas d'échantillonnage suivant la direction orthogonale au contour.

Leurs équations sont :

$$y = -a(x-x_0) + y_0 - 1/\cos\theta \quad (9-2)$$

et

$$y = -a(x-x_0) + y_0 + 1/\cos\theta \quad (9-3)$$

avec $x_0 = 32$ et $y_0 = 32$.

Pour le contour vertical, la zone contour Z_c est limitée par les deux droites

$$x = x_0 - 1 \quad (9-4)$$

et

$$x = x_0 + 1 \quad (9-5)$$

La définition du paramètre P_1 est :

$$P_1 = \frac{\text{nombre de points du signal détectés dans } Z_c}{\text{nombre de points du signal détectés dans } Z_c + \text{nombre total de points de bruit détectés}}$$

Les points provenant du signal, ne sont détectés que dans la zone contour.

Le paramètre P_2 proposé par Fram et Deutsch est l'estimation du rapport du nombre de perpendiculaires à la ligne de contraste qui contiennent un point détecté dans Z_c provenant du signal seul, avec le nombre de perpendiculaires ne contenant pas de point détecté dû au bruit dans Z_c .

A cause des problèmes liés à la discrétisation, au lieu de faire un calcul sur les perpendiculaires au contour, le calcul est réalisé sur les lignes de l'image. Pour un contour à 60° , le fait de considérer les lignes de l'image n'induit pas plus d'erreurs que de se ramener à un contour vertical par rotation de l'image comme le faisaient Fram et Deutsch. En effet, pour un contour à 60° , si la largeur de Z_c est strictement inférieure à 2, deux points consécutifs de Z_c sur une ligne ont des projections orthogonales sur le contour « vrai » distantes de moins de 0,5 pixel (*cf.* figure 9.2 points M_0 et M_1).

Pour les images de test, la zone de contour est limitée à une longueur de 54 pixels, afin de ne pas prendre en compte les effets de bords ; c'est-à-dire que les calculs de P_1 et P_2 ne sont pas effectués sur le pourtour de l'image (sur une épaisseur de 5 pixels). La largeur de Z_c est de 2 pixels (comptés le long de Ox). Sur le contour vertical, on ne considère comme correctement positionnés que les pixels des colonnes 31 et 32. Sur le contour oblique à 60° , on considère que deux pixels par ligne appartiennent à la zone de contour (*cf.* figure 9.3).

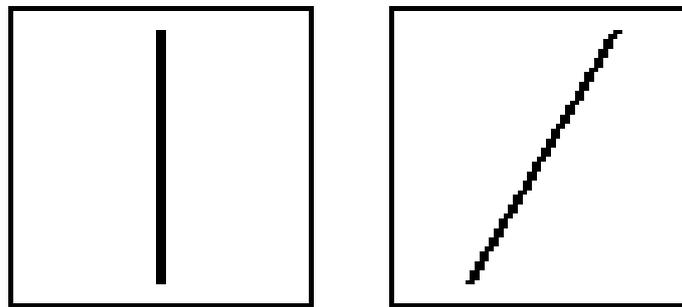


Figure 9.3 : Position de Z_c pour le contour vertical et pour le contour oblique

De plus, on suppose que la détection optimale est obtenue pour un contour mince, c'est-à-dire si on détecte un et un seul pixel par ligne dans Z_c . Les contours de 2 pixels d'épaisseur verront leur score P_1 diminué. Ceci signifie que le nombre de pixels de contour détectés est égal au nombre de perpendiculaires contenant exactement un point de contour détecté dans Z_c . Ce nombre est noté n_{sign}^c .

Les autres notations sont :

n_{br}^c : nombre de points provenant du bruit détectés dans Z_c

n^0 : nombre de points détectés en dehors de Z_c et donc dus au bruit seul

n_{br}^p : nombre de perpendiculaires contenant un point détecté dû au bruit dans Z_c

nlong : nombre de lignes prises en compte (ici 54)

$$P_1 = \frac{n_{sign}^c}{n_{sign}^c + n_{br}^c + n^0} \quad (9-6)$$

$$P_2 = \frac{n_{sign}^c}{nlong - n_{br}^p} \quad (9-7)$$

P_1 mesure donc la robustesse du détecteur en présence de bruit ; il est faible quand il y a beaucoup de fausses détections et vaut 1 quand il n'y a pas de pixels détectés dus au bruit.

P_2 caractérise bien le comportement du détecteur vis-à-vis du signal seul, il vaut 1 quand tous les points de la transition réelle sont extraits sans bruit et qu'il n'y a pas de lacune.

2.3 Paramètres de réglage des différents opérateurs

D'une manière générale, les paramètres des opérateurs ont été choisis de façon à optimiser (à l'œil) l'extraction du contour. Le but est bien entendu de détecter le maximum de pixels du contour, et le minimum de pixels dus au bruit. Mais il y a toujours un compromis à faire entre sur-détection (et donc détecter aussi du bruit) et sous-détection (et donc laisser des lacunes dans le contour). Un contour mince, d'épaisseur un pixel est en outre préférable et fournit un meilleur indice P_2 . Quand l'optimum oscille entre deux jeux de paramètres voisins, celui qui extrait la ligne de contour la plus complète est choisi, quitte à extraire un peu plus de bruit, ce qui revient à favoriser P_2 par rapport à P_1 .

	transition verticale				transition oblique			
	a=3	a=6	a=12	a=20	a=3	a=6	a=12	a=20
α	0,7	1	1	1	0,5	0,5	1	1
sh _{d1} , sb _{d1}	2 1	4 2	8 5	15 1	2 1	4 1	10 1	15 1
sh _{d2} , sb _{d2}	3 2	5 4	9 5	20 10	2 2	4 4	8 8	15 15
sh _{d3} , sb _{d3}	3 2	6 4	7 4	18 15	2 2	4 4	8 8	15 13
β	0,5	0,7	0,7	0,7	0,5	0,5	0,7	0,7
sh _{s1} , sb _{s1}	2 1	3 2	6 4	8 5	1 1	4 1	6 1	8 1
sh _{s2} , sb _{s2}	2 1	3 2	6 4	8 5	1 1	3 2	5 4	8 7
sh _{s3} , sb _{s3}	2 1	3 2	6 4	8 5	1 1	3 2	5 4	8 7
s _M	12	17	20	20	11	15	20	30
s _N	2	3	6	8	2	3	4	5
s _{MM}	2	2	2	2	2	2	3	6
γ_G	5				5	5		
s _G	2				2	3		
ΔU_G	80				80	80		
γ_F^2	6				7	10		
s _F	5				4	4		
ΔU_F	100				100	100		

Tableau 9.1 : Paramètres de réglage des différents opérateurs

Les paramètres des opérateurs (donnés dans le tableau 9.1) sont les suivants :

Deriche1, Deriche2 et Deriche3

Pour les trois méthodes, on utilise un seuillage par hystérésis avec respectivement sh_d et sb_d comme seuils haut et bas. Le paramètre α , qui correspond à l'échelle d'analyse, a la même valeur pour les trois méthodes.

Shen1, Shen2 et Shen3

Pour les trois méthodes, on utilise un seuillage par hystérésis avec respectivement sh_s et sb_s comme seuils haut et bas. Le paramètre β a la même

valeur pour les trois méthodes. Ce paramètre a été choisi de manière à optimiser P_1 et P_2 . Il ne respecte donc pas nécessairement la règle employée pour la comparaison théorique du chapitre V (cf. tableau 5.1).

MDIF

un seul seuil s_M , sur la norme du gradient

NAGDIF

un seul seuil s_N , sur la norme du gradient

Morpho

Toutes les images sont filtrées avec un filtre alterné séquentiel comprenant une fermeture suivie d'une ouverture (FAS_1). L'amincissement de l'image du gradient morphologique est itéré jusqu'à stabilisation. Le seul paramètre est le seuil s_{MM} pour la binarisation de l'image de la norme du gradient.

MFA

Pour la méthode du recuit par champs moyens (MFA), le choix de la loi de variation de température est très important. En effet, lorsque les images sont peu contrastées ou fortement bruitées, une variation trop rapide de la température donne des images lissées de mauvaise qualité ainsi qu'une détection de contours moyenne alors que le test de convergence de l'algorithme (basé sur la variation d'énergie ΔU_F) est correct. Après avoir testé différentes lois de variation de température proposées dans la littérature, nous avons choisi une loi empirique qui donne de bons résultats sur toutes les images testées :

$\theta_{\square} = 1/T$ est initialisé à 0.0002

$\theta_{\square} = \theta_{\square} * 4$ tant que θ_1 .

Les paramètres à fixer sont le facteur de lissage γ_F et le coût α à payer pour créer une discontinuité (ce qui revient à choisir le seuil $s_F = \sqrt{2\alpha/\gamma_F}$).

GNC

Pour la méthode du GNC, les deux paramètres à choisir sont le facteur de lissage γ_G , qui est également un facteur d'échelle (cf. [BLAK87]) et le seuil $s_G = \xi_0 = \sqrt{2\alpha/\gamma_G}$ qui fixe la limite de détection de contours. Le test d'arrêt de l'algorithme est basé sur un seuillage de la variation d'énergie ΔU_G calculée après chaque itération.

2.4 Analyse des résultats

Certains des logiciels utilisés figurent dans la banque de logiciels SIMPA du GDR 134 ; il s'agit des opérateurs de Deriche (dérivée première, seconde et laplacien), de l'extraction des maximums de la norme du gradient et des passages

par zéro de la dérivée seconde ou du laplacien dans la direction du gradient, de l'opérateur MDIF et du seuillage par hystérésis. Les opérateurs NAGDIF et Shen-Castan des tests sont différents, dans leur implantation, de ceux de la banque SIMPA.

Temps d'exécution

Les algorithmes testés ont été implantés sans souci d'optimisation. Les temps de calcul sont donc donnés à titre indicatif. Les opérateurs de l'approche markovienne (GNC et MFA) ont été exécutés sur une « Connection Machine » CM2, à l'INRIA (Sophia Antipolis), les autres ont été exécutés sur une station SUN de type SPARC5, au laboratoire ETIS (ENSEA).

	Deriche1	Shen1	MDIF	NAGDIF	Morpho	MFA	GNC
machine	SPARC5	SPARC5	SPARC5	SPARC5	SPARC5	CM2	CM2
temps	2,5s	2,1s	0,8s	1,4s	70,7s	18,2s	11,2s

Tableau 9.2 : Temps d'exécution des opérateurs testés sur une image 256 × 256

Il est à noter que certains de ces opérateurs ont été implantés en ASIC pour des applications en « temps réel », c'est le cas des algorithmes NAGDIF [DEMI93a] et Deriche [DEMI93b].

Les tableaux 9.3 et 9.4 présentent les valeurs des deux paramètres P_1 et P_2 , calculés sur chacune des images, comportant la transition verticale ou oblique et pour les différentes valeurs de saut d'amplitude.

	a=3		a=6		a=12		a=20	
	P ₁	P ₂						
Deriche 1	1,00	0,93	1,00	1,00	1,00	1,00	1,00	1,00
Deriche 2	0,47	0,83	0,42	0,91	0,60	1,00	0,56	1,00
Deriche 3	0,77	0,98	0,71	1,00	0,86	0,98	0,89	1,00
Shen-Castan 1	0,96	0,98	1,00	1,00	1,00	1,00	1,00	1,00
Shen-Castan 2	0,19	0,51	0,30	0,39	0,52	0,70	0,19	0,22
Shen-Castan 3	0,39	0,76	0,29	0,28	0,63	0,74	0,04	0,04
MDIF	0,33	0,79	0,96	1,00	1,00	1,00	1,00	1,00
NAGDIF	0,29	0,53	0,72	0,96	0,71	1,00	0,86	1,00
Morpho	0,81	0,24	0,69	0,88	1,00	1,00	1,00	1,00
GNC	0,32	0,96						
MFA	0,62	0,63						

Tableau 9.3 : Transition verticale, d'amplitude a (cf. figures 9.4 et 9.5)

	a=3		a=6		a=12		a=20	
	P ₁	P ₂						
Deriche 1	0,67	0,56	0,90	0,80	0,98	0,98	1,00	1,00
Deriche 2	0,22	0,42	0,34	0,63	0,42	0,97	0,54	1,00
Deriche 3	0,20	0,40	0,17	0,39	0,30	0,44	0,31	0,45
Shen-Castan 1	0,25	0,57	0,46	0,84	0,56	0,95	0,49	0,97
Shen-Castan 2	0,13	0,47	0,18	0,37	0,43	0,55	0,39	0,46
Shen-Castan 3	0,13	0,41	0,29	0,48	0,41	0,52	0,37	0,43
MDIF	0,13	0,47	0,41	0,73	0,50	0,95	0,51	0,98
NAGDIF	0,16	0,30	0,36	0,67	0,39	1,00	0,34	1,00
Morpho	0,35	0,13	0,34	0,69	0,39	0,95	0,41	0,97
GNC	0,05	0,28	0,08	0,12				
MFA	0,31	0,15	0,36	0,59				

Tableau 9.4 : Transition oblique, d'amplitude a (cf. figures 9.6 et 9.7)

*Cas des signaux de faible amplitude noyés dans le bruit
(amplitude 3 ; cf. figures 9.4 et 9.6)*

Pour ces images très difficiles, le meilleur opérateur est Deriche1, puis Shen1, tant pour la transition verticale que pour la transition oblique. On observe sans surprise que les opérateurs utilisant la dérivée seconde (Deriche2, Deriche3, Shen2, Shen3) sont plus sensibles au bruit (paramètre P_1). En particulier, ce phénomène apparaît pour les transitions obliques où les contours obtenus sont plus tourmentés. On peut remarquer que les opérateurs MDIF et NAGDIF obtiennent des scores honorables malgré leur simplicité.

Pour les valeurs de réglage proposées, l'opérateur MFA donne moins de fausses alarmes que l'opérateur GNC, mais il laisse beaucoup de lacunes dans les contours.

*Cas des signaux d'amplitude importante vis-à-vis du bruit
(amplitude 6 ; cf. figures 9.5 et 9.7)*

Pour une transition verticale, tous les opérateurs donnent de bons résultats, sauf les opérateurs « dérivée seconde » de Shen-Castan. Les mauvais scores obtenus, lorsque l'amplitude de la transition augmente, sont dus à un décalage du contour extrait d'un pixel vers la droite. De façon classique, on observe que les opérateurs utilisant la dérivée seconde donnent des contours moins « lisses » que ceux fournis par les opérateurs employant le gradient.

Pour une transition oblique, les résultats sont globalement moins bons que pour une transition verticale. Ceci est dû en partie aux dégradations apportées par la quantification spatiale. Les contours tourmentés obtenus ont en effet tendance à « déborder » de la zone contour.

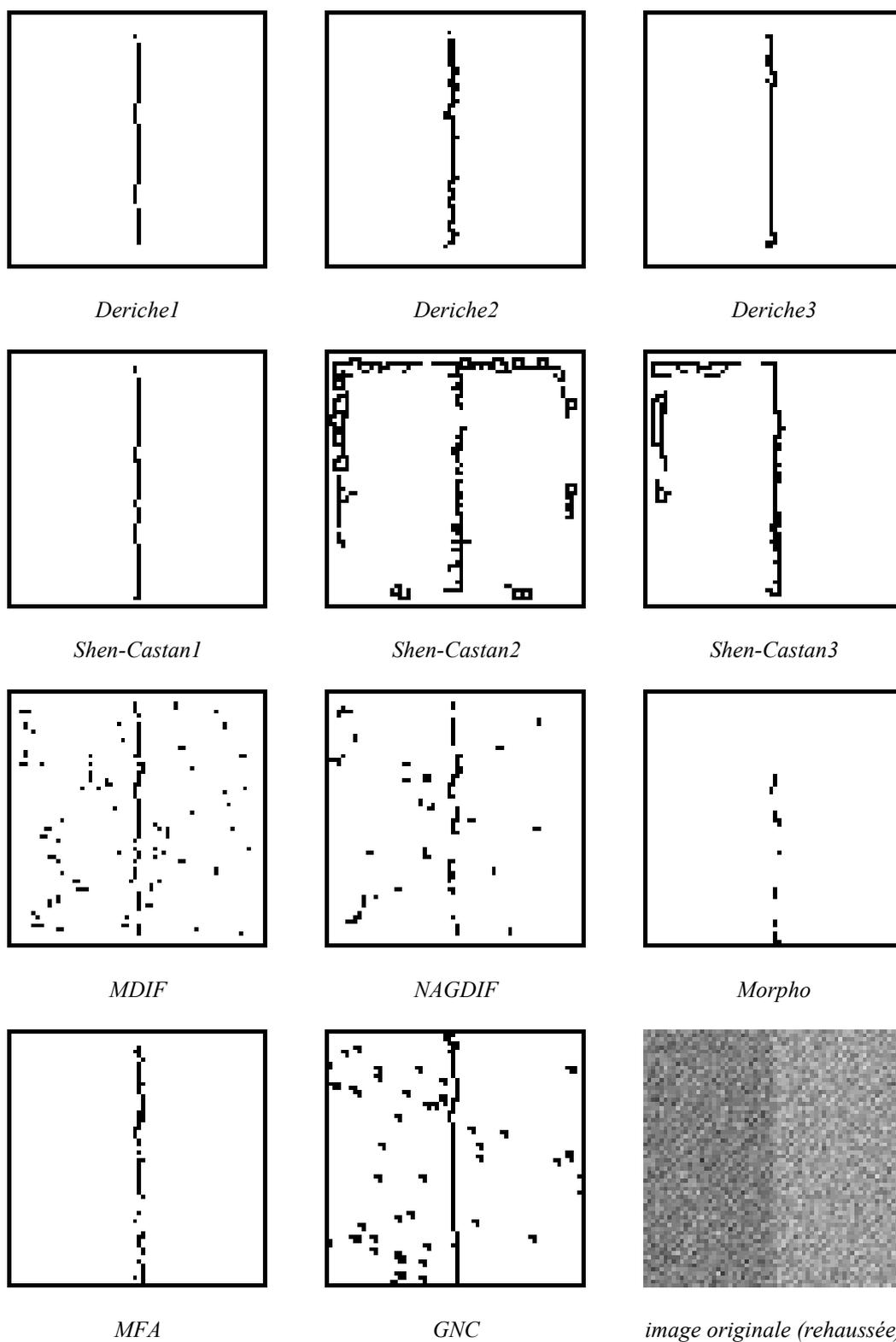


Figure 9.4 : Détecteurs de contours sur image de saut d'amplitude 3

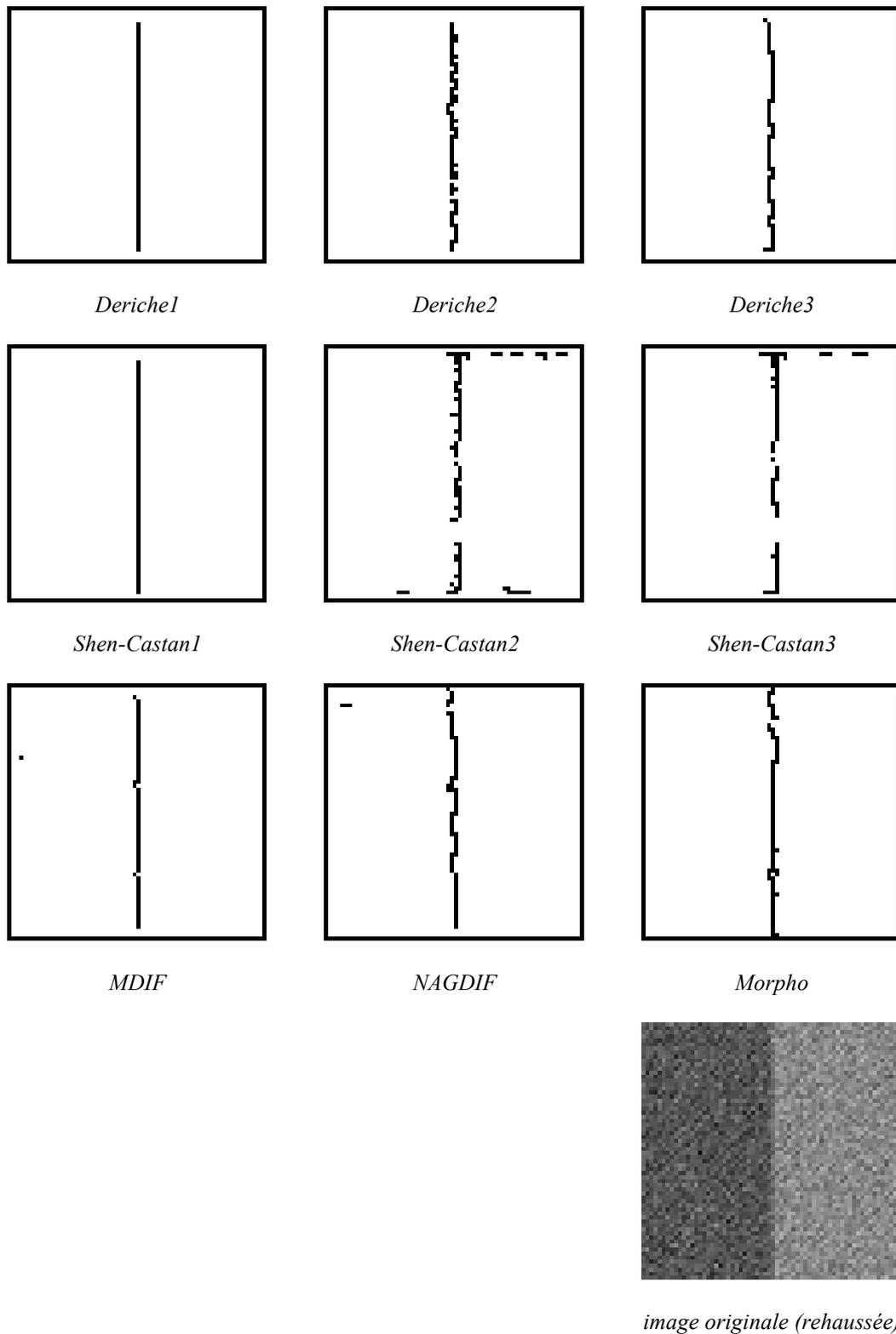


Figure 9.5 : Détecteurs de contours sur image de saut d'amplitude 6

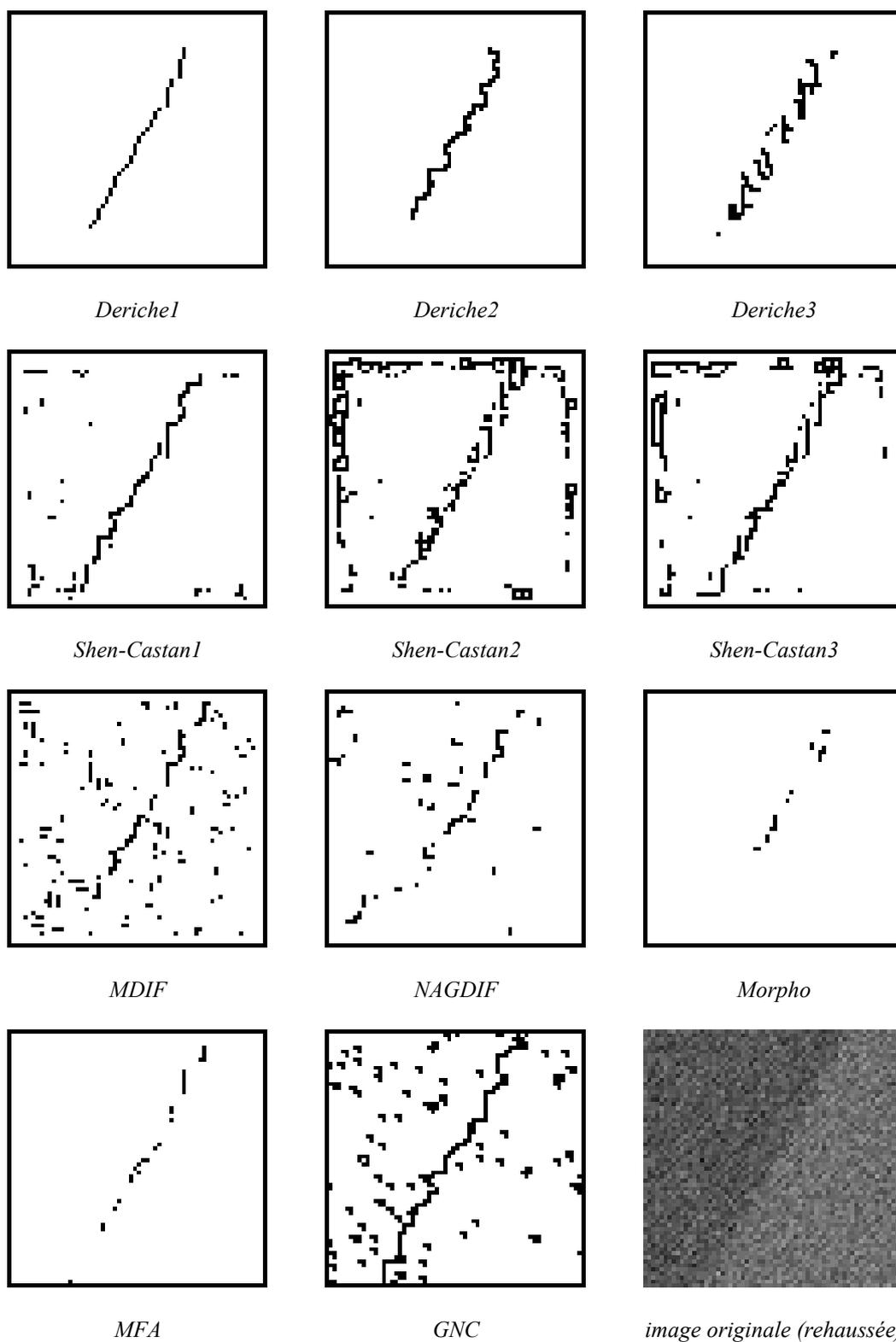


Figure 9.6 : Détecteurs de contours sur image de saut d'amplitude 3

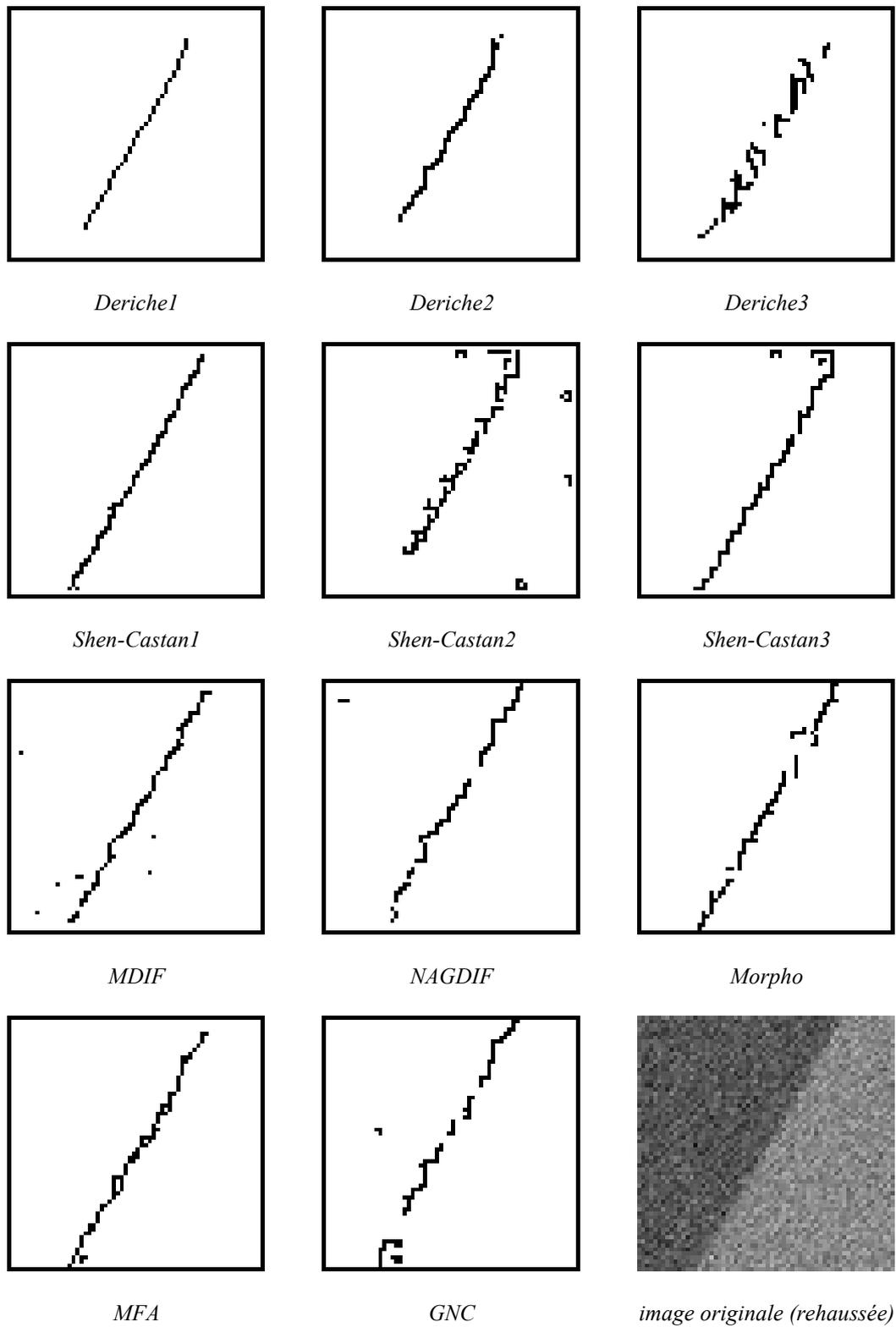


Figure 9.7 : Détecteurs de contours sur image de saut d'amplitude 6

3 Résultats sur les images réelles

3.1 Description des images

Les opérateurs d'extraction de contour ont été testés sur 6 images réelles appartenant à la banque d'images du GDR 134 TDSI du CNRS. Elles ont toutes pour dimension 256 x 256 et sont codées sur 8 bits.

Elles ont été sélectionnées, d'une part pour leur appartenance à des domaines d'application différents (trois images biomédicales, une image satellitaire, une image d'intérieur et une image de scène industrielle) et d'autre part parce que la qualité de la segmentation obtenue peut être jugée par un observateur non averti.

Voici leur description :

ANGIOGRAPHIE : image d'une angiographie crânienne qui comporte plusieurs vaisseaux sanguins de contrastes très variés. L'image a été obtenue par numérisation d'un film radiographique contenant des rayures que l'on retrouve sur l'image. On remarque également du bruit impulsionnel lié à la numérisation. Les contours des vaisseaux sont souvent mal délimités, parce qu'ils correspondent à des transitions lentes.

Les niveaux de gris sont compris entre 20 et 255.

AQUITAINE : image satellitaire SPOT (scène S1H287122911729-WRS 38-262 du 29 décembre 1987) d'une zone parcellaire de la région Aquitaine.

La scène comporte à la fois des grandes zones homogènes (grandes parcelles, bois), des contours nets (cours d'eau, routes, lisières des bois) et des zones plus disparates (habitat dispersé, bosquets, petites parcelles), ou texturées (lignage dans les champs). Les contours ou les régions à extraire peuvent être très variables selon le domaine d'application. Pour les contours, on pourra se focaliser sur les cours d'eau, les routes, les bordures des champs et des bois. Pour les régions, on peut s'intéresser aux parcelles.

La dynamique de l'image est très faible, puisque les niveaux de gris vont de 10 à 40. L'image affichée en figure 9.9 est rehaussée pour une meilleure lisibilité (*cf.* figure 4.1), mais les traitements ont été effectués sur l'image brute (ainsi que ceux du chapitre XIII).

BAGUES : image industrielle comprenant 8 bagues de roulement. Les contours des objets sont nets pour certains, plus difficiles à discerner pour d'autres, à cause de la présence d'ombres et de reflets. L'image est bien contrastée (la dynamique s'étend de 0 à 255), mais il y a beaucoup de reflets sur les parties métalliques et le fond (sombre) est très bruité.

COULOIR : image d'intérieur, utilisée en robotique mobile. Elle inclut des contours nets (dalles du sol, éclairage plafonnier, limites de couloirs latéraux) et des grandes régions homogènes mais dont l'éclairage varie lentement à cause de l'éloignement progressif des sources de lumière et de la caméra (tout

particulièrement le mur du fond et les murs latéraux). La dynamique s'étend de 0 à 255.

IRM (image par résonance magnétique) : image scanner d'une coupe du thorax au niveau du cœur obtenue par résonance magnétique.

Selon le domaine d'application, on pourra chercher à extraire un organe, un muscle ou un os. L'image comporte un objet principal (coupe du thorax) et une partie du bras gauche sur la droite de l'image. La personne fait face au coin inférieur gauche de l'image. On peut distinguer une vertèbre, en haut de l'image et quelques côtes. Les parties bien visibles de la coupe du thorax sont : une masse claire occupant toute la partie supérieure gauche et qui correspond au foie, une importante masse foncée, qui occupe la partie droite et qui correspond au poumon. Le cœur est situé entre le foie et le poumon, à l'avant de la personne, on peut distinguer les deux ventricules (deux parties sombres).

Bien que les niveaux de gris s'étendent de 0 à 255, l'image est très sombre, peu contrastée et contient du bruit lié à la numérisation, notamment sur une bande horizontale au milieu de l'image. L'image de la figure 9.12 est affichée après un rehaussement de la dynamique pour assurer une meilleure lisibilité.

MUSCLE : image de fibres musculaires, avec des fibres foncées, homogènes, des fibres claires et texturées et des interstices entre ces fibres qui sont clairs et homogènes (voir l'histogramme de cette image Figure 2.2). Cette image est la plus facile à juger, puisque tout observateur est capable d'en donner une segmentation idéale. Celle-ci serait toutefois guidée par des critères de forme, de taille, que n'intègrent pas les opérateurs de segmentation testés.

La dynamique de l'image est bonne (44, 249). Les contours sont nets entre les fibres de couleur différente et entre les fibres sombres et le fond. Par contre, ils sont peu marqués entre les fibres claires et le fond ainsi qu'entre les fibres de même couleur.

3.2 Paramètres de réglage des différents opérateurs

D'une manière générale, les paramètres des opérateurs ont été choisis de façon à optimiser (à l'œil) l'extraction du contour. Le but est bien entendu de détecter le maximum de pixels du contour, et le minimum de pixels dus au bruit. Mais il y a toujours un compromis à faire entre sur-détection (et donc détecter aussi du bruit) et sous-détection (et donc laisser des lacunes dans le contour). Un contour mince, d'épaisseur un pixel est en outre préférable et fournit un meilleur indice P_2 . Quand l'optimum oscillait entre deux jeux de paramètres voisins, celui qui extrayait la ligne de contour la plus complète a été choisi, quitte à extraire un peu plus de bruit, ce qui revient à favoriser le critère P_2 par rapport à P_1 .

Cinq méthodes d'extraction de contour ont été comparées sur ces 6 images :

- Deriche, dérivée première
- Shen-Castan, dérivée première

- MDIF
- NAGDIF
- MFA

Les paramètres de réglage sont les suivants :

	Deriche			Shen			MDIF	NAG DIF	MFA		
	α	sh_d	sb_d	β	sh_s	sb_s	s_M	s_N	γ^2_F	α	ΔU_F
ANGIOGRAPHIE	0.5	5	3	0.35	7	3	100	20	6	160	100
AQUITAINE	1.0	2	2	0.7	1	1	10	2	2	2	100
BAGUES	1.0	10	5	0.7	5	3	100	20	6	180	100
COULOIR	1.0	9	3	0.7	5	3	50	10	6	180	100
IRM	1.0	10	5	0.7	5	2	60	12	6	80	100
MUSCLE	1.0	25	5	0.7	15	7	130	30	6	300	100

Tableau 9.5 : Paramètres de réglage des opérateurs

3.3 Analyse des résultats

Les méthodes GNC et MFA donnant sensiblement les mêmes résultats sur ces images réelles, seuls les résultats sur MFA sont présentés. La méthode morphologique présentée au paragraphe précédent est inadaptée sur ces images réelles, ce qui n'exclut pas que d'autres enchaînements de traitements morphologiques ne donnent pas d'excellents résultats.

Pour chaque opérateur, les paramètres ont été réglés de façon empirique, après plusieurs essais, avec toutefois le souci d'extraire sensiblement les mêmes contours. Pour chaque image, sont donnés dans les paragraphes suivants les critères qui ont guidé les choix des paramètres des opérateurs, ainsi que des commentaires sur les résultats obtenus.

ANGIOGRAPHIE : Le but ici est d'extraire les contours des vaisseaux. L'image comporte plusieurs types de bruit : rayures du film radiographique, bruit plutôt impulsif dans le fond et de distribution plus concentrée à l'intérieur des vaisseaux.

Comme les transitions sont lentes (parfois plus de 5 pixels), les opérateurs MDIF et NAGDIF fournissent des contours hachés, doublés, non fermés, même pour les plus marqués d'entre eux, si bien que les contours extraits sont parfois difficiles à distinguer du bruit. Ceci est dû au fait que ces deux opérateurs travaillent sur des voisinages de taille 5x5. De même l'opérateur markovien MFA dédouble les transitions lentes et n'extraie pas les vaisseaux faiblement marqués.

Les rayures ne perturbent pas trop la détection de l'information, mais conduisent à une certaine sous-détection. Par contre, MFA et, dans une moindre mesure, MDIF et NAGDIF sont sensibles au bruit impulsionnel.

Les meilleurs résultats sont obtenus par Shen-Castan et Deriche, ce dernier donnant plus de contours dans le bas de l'image, certains sont dus au bruit et aux rayures mais d'autres sont des contours de vaisseaux à peine visibles sur l'image.

AQUITAINE : Dans cette image, il y a différents types d'objets que l'on peut souhaiter extraire selon le domaine d'application. L'attention s'est plutôt portée sur les cours d'eau, la route au milieu de l'image et le petit bois du bas de l'image.

Les résultats de toutes les méthodes sont assez proches. Si l'on se focalise sur le cours d'eau et sur les routes, ils sont extraits par tous les opérateurs, avec toutefois un léger avantage pour la méthode MDIF, qui extrait correctement les deux bords de ces structures fines.

BAGUES : Le but ici est de trouver au mieux les contours extérieurs et intérieurs des bagues et de s'affranchir des divers reflets et du bruit de fond d'image.

Les bagues sont extraites par tous les opérateurs, toutefois NAGDIF et MFA sont plus gênés par les variations d'éclairage sur les bagues du haut, qui engendrent des transitions lentes, considérées comme des contours par ces deux opérateurs. Deriche et Shen-Castan sont plus sensibles aux reflets sur les parties métalliques ainsi qu'au bruit dans la partie sombre de l'image. Les pourtours des bagues sont mieux fermés avec MDIF et Shen-Castan, qui ne créent pas de liaisons parasites entre les contours externes et internes des bagues comme MFA et NAGDIF.

COULOIR : Sur cette image, l'attention s'est portée sur l'extraction des lignes de séparation des dalles du sol et les contours verticaux, limites de couloirs latéraux, avec le souci de ne pas obtenir de faux contours liés aux variations d'éclairage des murs.

Les lignes fines (pourtours des dalles du sol, lignes au fond du plafond) sont bien extraites par MDIF, Shen-Castan et Deriche. Les variations d'éclairage (dégradés) du mur du fond perturbent le moins MDIF, elles engendrent des régions différentes pour MFA, délimitées par des contours francs. Elles créent de petits contours, assimilables à du bruit pour les autres opérateurs.

IRM : Le but sur cette image a été d'obtenir les contours du thorax complet, du foie et du poumon, les plus fermés possible, en ayant un minimum de bruit à l'intérieur du foie et du poumon.

Tous les opérateurs extraient correctement ces trois pourtours. Shen est celui qui ferme le mieux ces trois contours, mais avec la contrepartie d'avoir beaucoup de petits contours à l'intérieur des organes. L'opérateur MFA fournit l'image la plus « propre » ; il est moins sensible au bruit dans le fond ainsi que dans les différents organes. Le cœur est extrait de manière inégale, mais seuls les

spécialistes des images médicales peuvent juger de la qualité de la segmentation de cette image.

MUSCLE : Il s'agit ici d'extraire les fibres noires et les fibres claires, en ayant un minimum de détection à l'intérieur des fibres claires, qui sont très texturées. La difficulté réside donc d'une part dans la présence de cette texture, et d'autre part dans des frontières extrêmement tenues, voire inexistantes, entre fibres de même nature.

Les contours des amas de fibres noires et des amas de fibres claires sont dans l'ensemble bien extraits. Cependant, bien que ne présentant pas d'angle, ces contours ne sont pas toujours bien fermés, sauf par MFA. On remarque d'ailleurs bien ici que, pour MFA, les contours les plus marqués sont épais, alors que les autres sont minces. La texture présente dans les fibres claires a le plus perturbé NAGDIF et MDIF. Les frontières peu marquées entre fibres noires accolées ne sont extraites par aucun opérateur.

3.4 Conclusion

A partir des résultats obtenus pour ces images, on peut avancer prudemment quelques observations :

Les transitions franches, dans un milieu peu bruité, sont bien extraites par tous les traitements, les opérateurs MDIF et NAGDIF ayant l'avantage de la rapidité d'exécution.

Dans le cas de transitions lentes, correspondant soit à des contours « flous », soit à des variations d'éclairement, on a un mauvais comportement de MDIF, NAGDIF et MFA, qui vont, dans le premier cas, créer des contours hachés ou doublés, et, dans le deuxième cas, créer des artefacts (faux contours). Par contre, comme on peut régler la taille des filtres de Deriche et de Shen, par le choix des paramètres α et β , ces deux opérateurs sont mieux adaptés à ce type de transition, avec un léger avantage pour Deriche.

Les contours de très faible amplitude, même en milieu bruité, sont particulièrement bien extraits par Deriche et, dans une moindre mesure, par Shen-Castan.

Les contours extraits par MFA sont plutôt longs, fermés ; les transitions de forte amplitude y sont indiquées par des contours épais et dans le cas de transitions lentes entre objets, ils sont doublés.

Les lignes fines, bien marquées, correspondant à des contours proches sont particulièrement bien extraites par MDIF.

Les contours de zones texturées sont mal extraits par NAGDIF et MDIF, mais peuvent l'être par MFA, Deriche ou Shen-Castan, pourvu que les intensités moyennes soient différentes de part et d'autre des contours.

On note enfin une sensibilité au bruit impulsionnel de MFA et MDIF. Les autres types de bruit rencontrés sur ces images ne nuisent pas à MFA, alors que NAGDIF y est plus sensible.

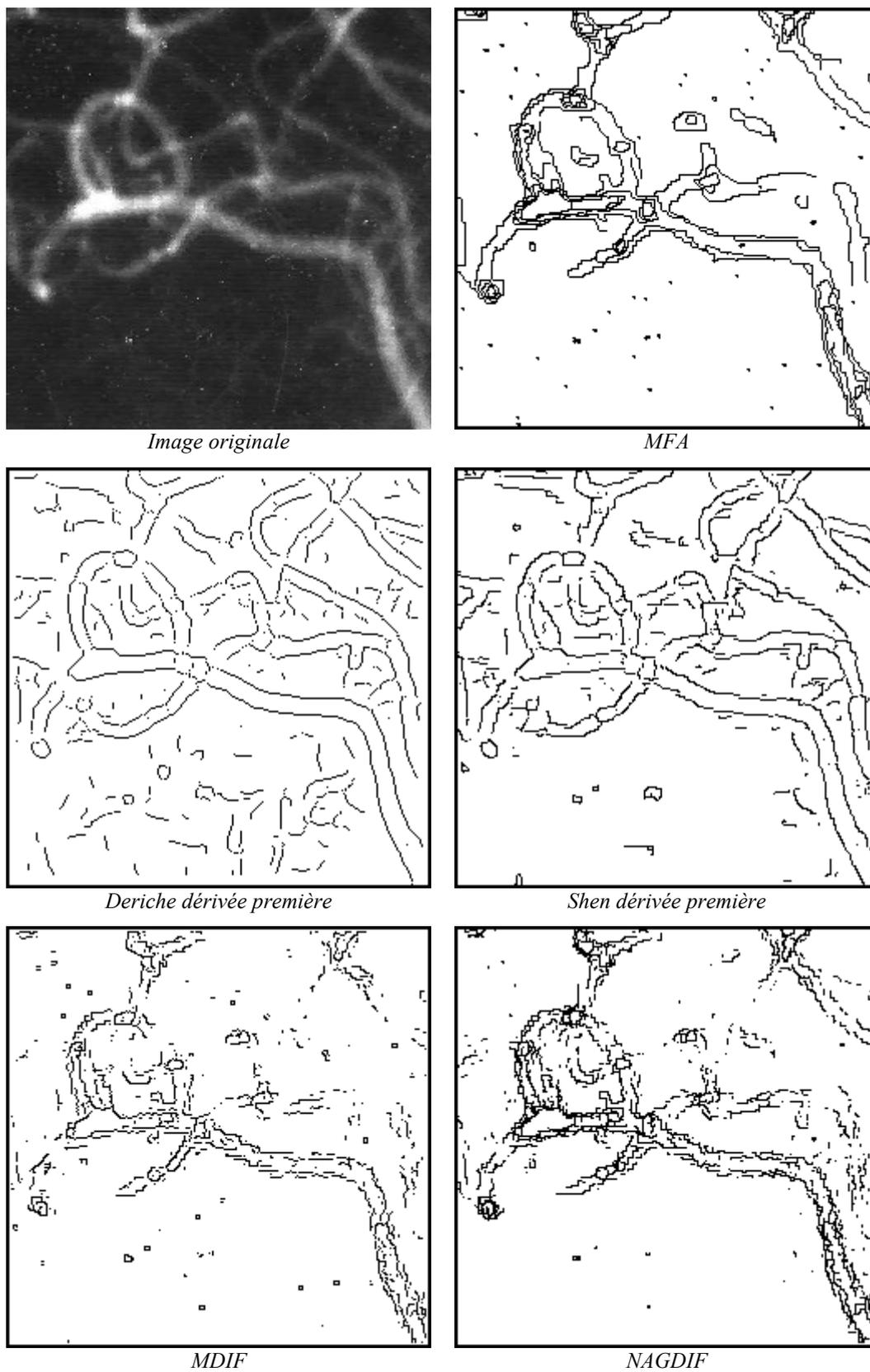


Figure 9.8 : Image ANGIOGRAPHIE



Image originale (rehaussée)



MFA



Deriche dérivée première



Shen dérivée première



MDIF



NAGDIF

Figure 9.9 : Image AQUITAINE

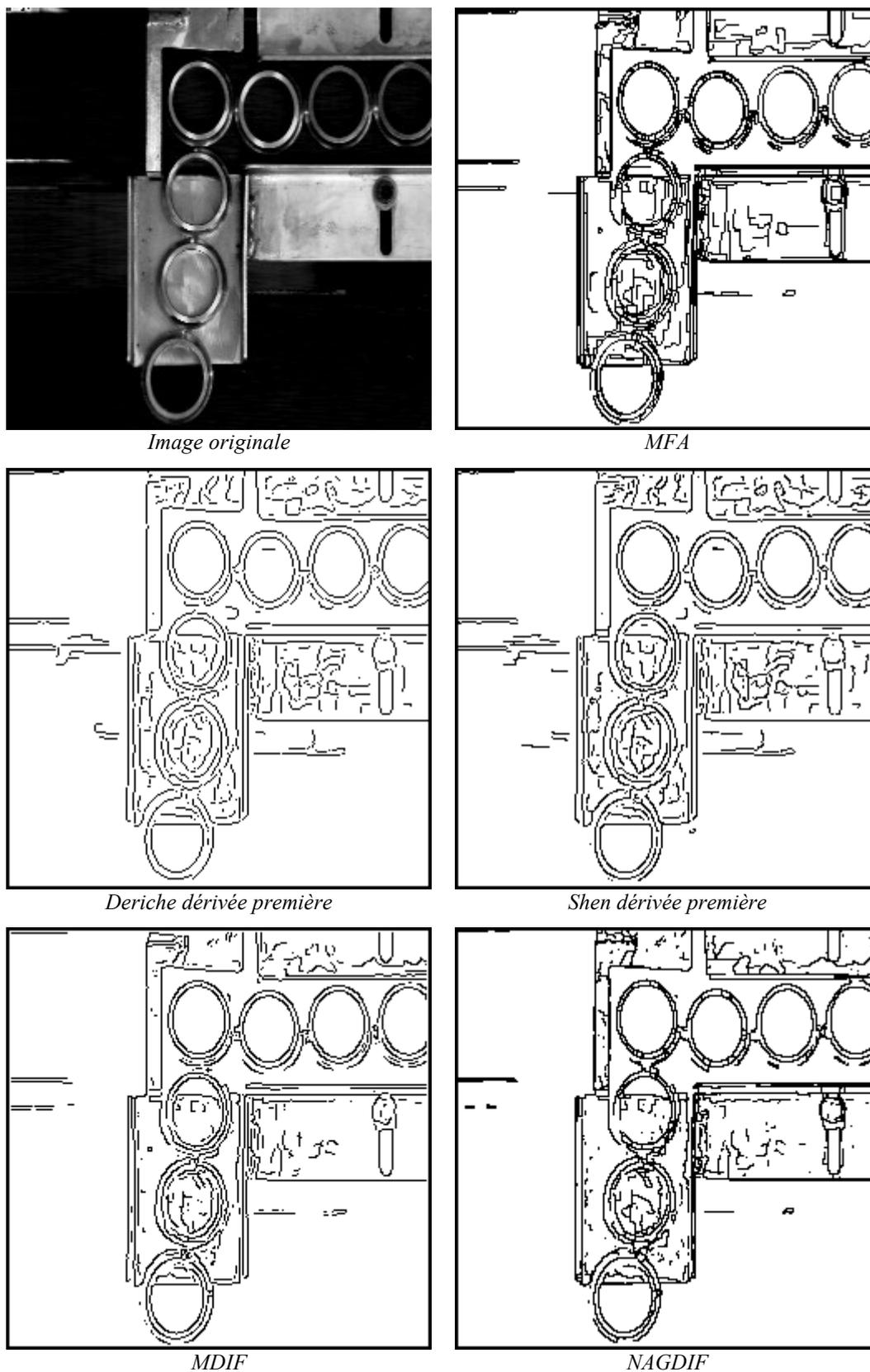
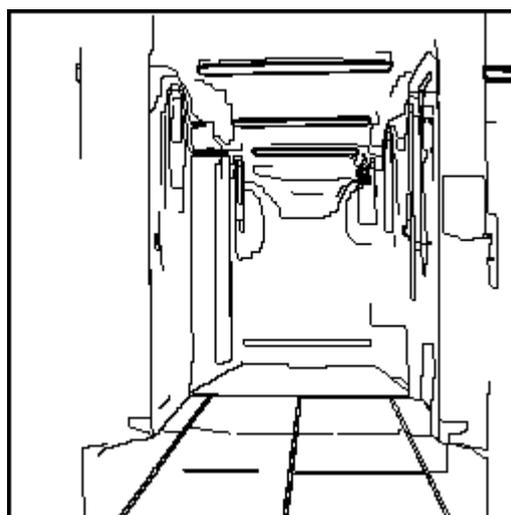


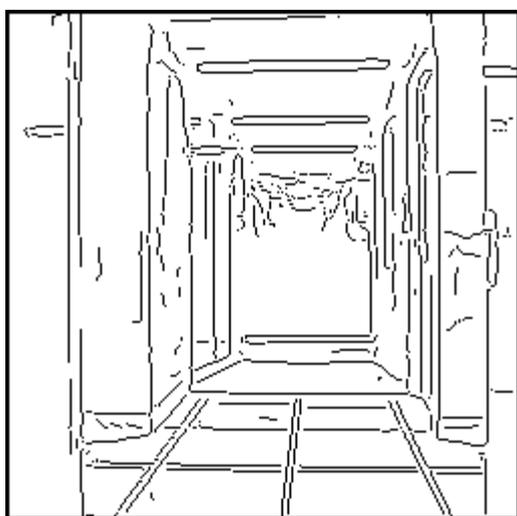
Figure 9.10 : Image BAGUES



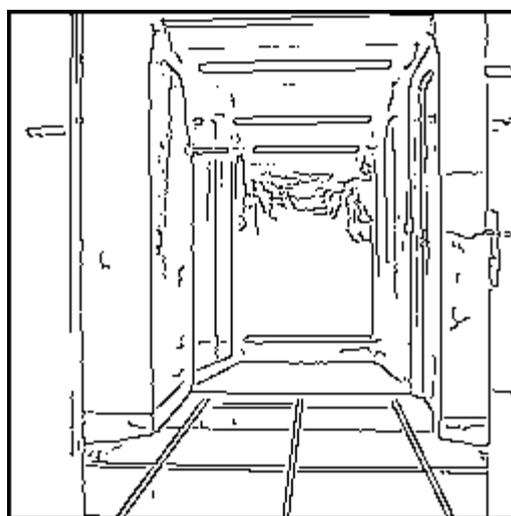
Image originale



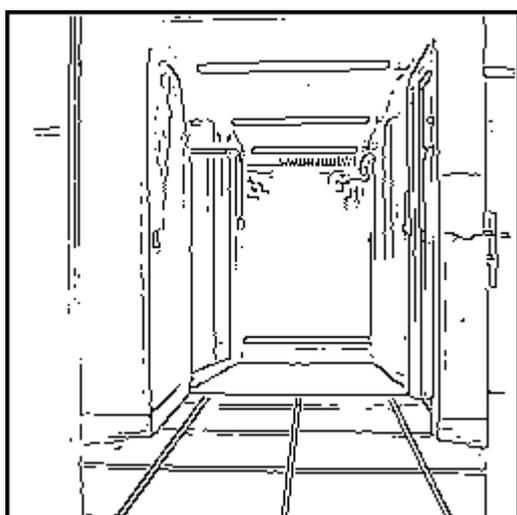
MFA



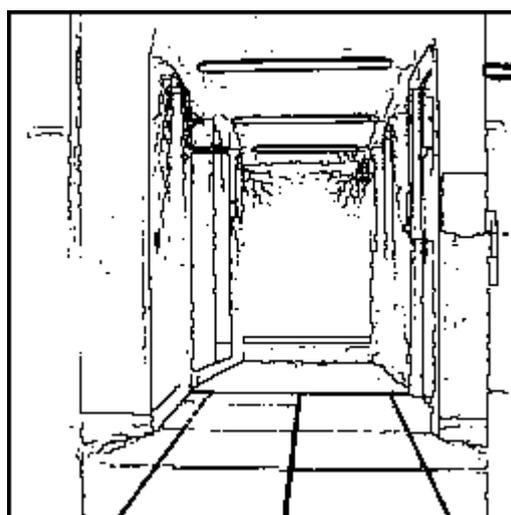
Deriche dérivée première



Shen dérivée première



MDIF



NAGDIF

Figure 9.11 : Image COULOIR

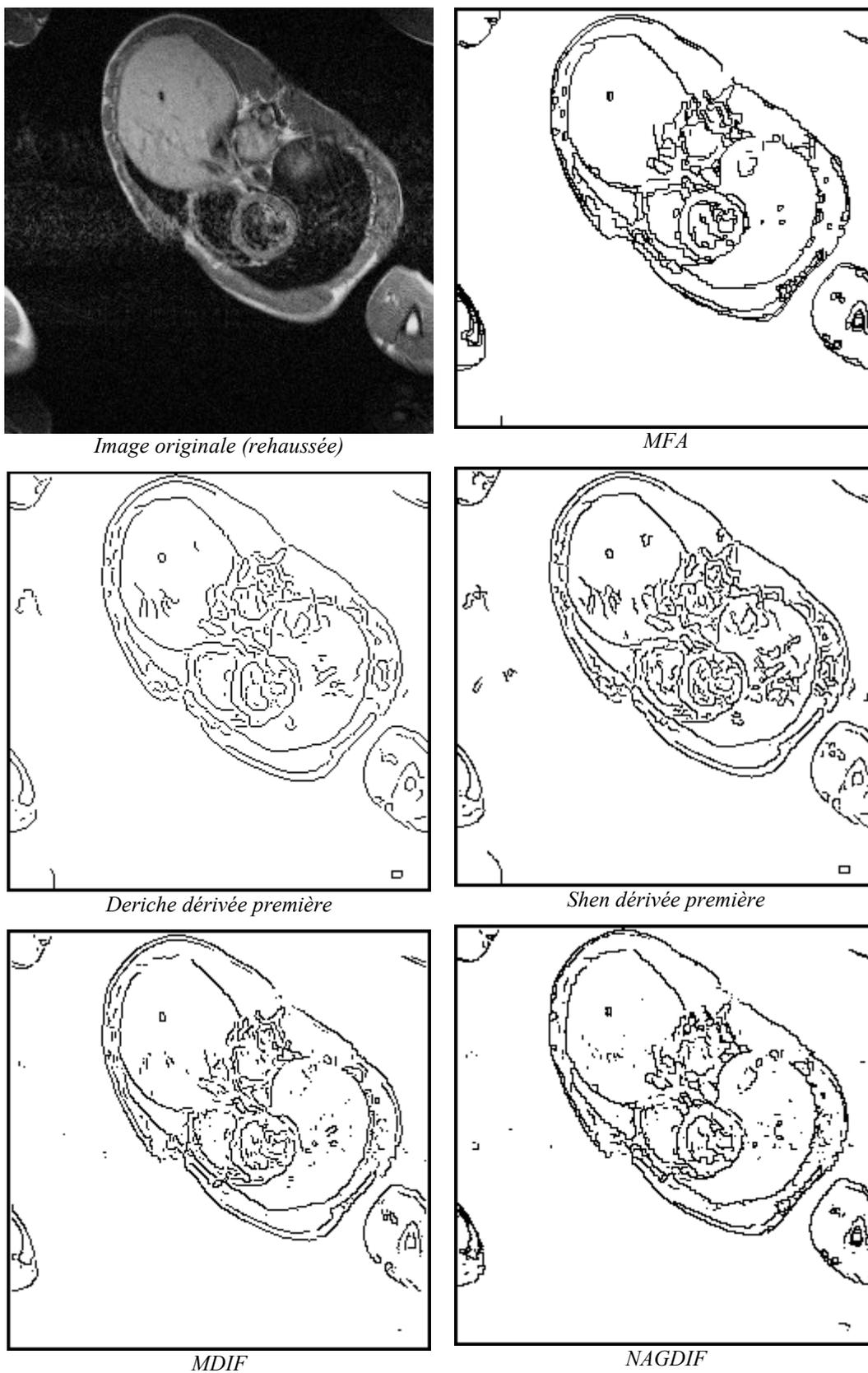


Figure 9.12 : Image IRM

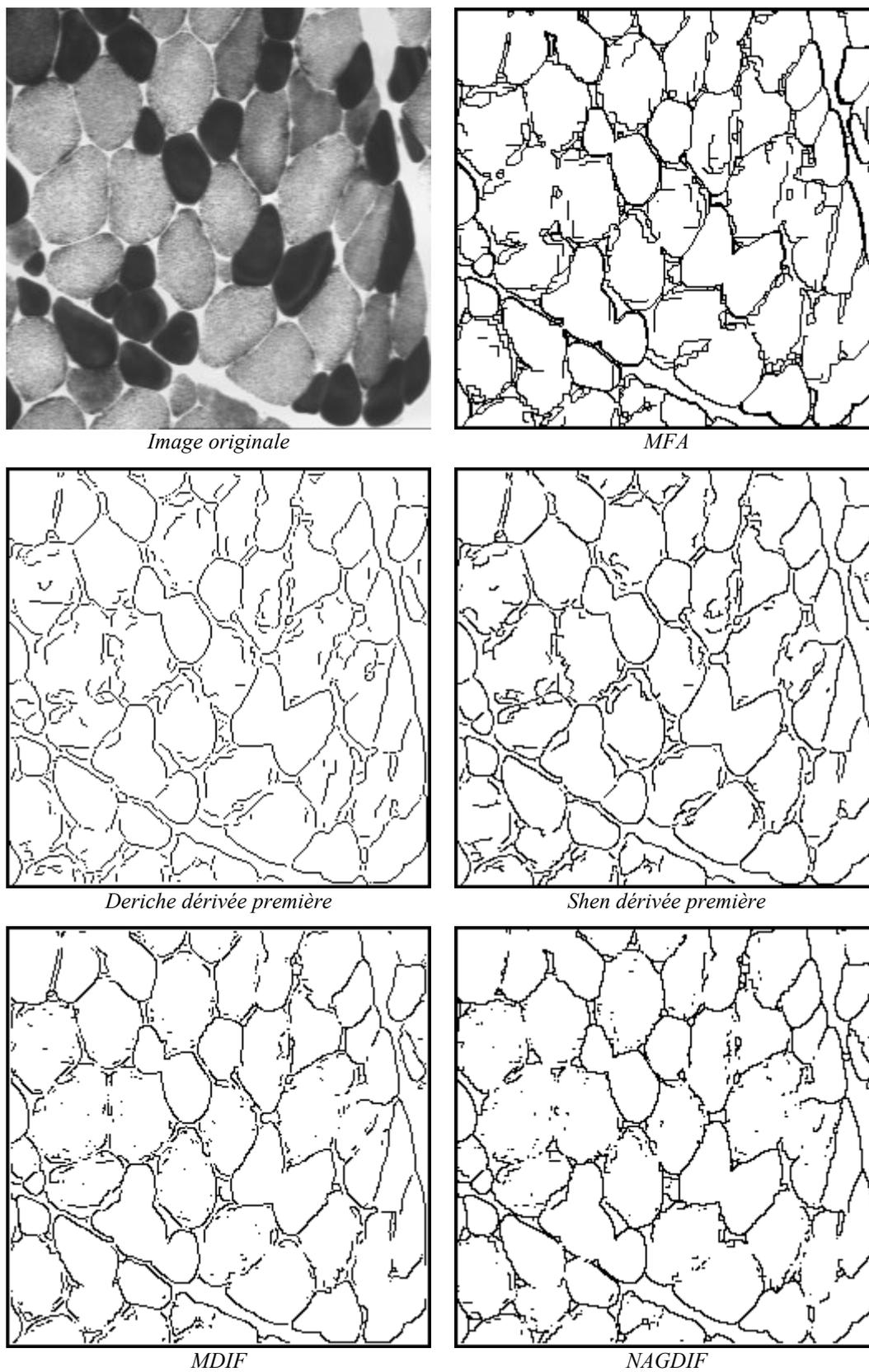


Figure 9.13 : Image MUSCLE

CHAPITRE X

Méthodes de classification

Ce chapitre présente quelques méthodes consistant à classifier des *individus* en exploitant leur similarité, par opposition à l'approche frontière, qui recherche les dissimilarités.

Les individus peuvent être :

- les pixels eux-mêmes ; les attributs sont alors ponctuels (comme le niveau de gris) ou calculés sur un voisinage du pixel (comme la moyenne des intensités ou un indice de texture calculé sur une fenêtre centrée sur le pixel).
- une sous-image de taille fixe ; les attributs sont alors calculés pour cette sous-image. Par exemple, on peut découper l'image en fenêtres ne se chevauchant pas et classifier chaque fenêtre.
- une région obtenue par une méthode de segmentation et que l'on cherche à classifier ; on se situe ici en aval de la segmentation, dans une phase d'interprétation de l'image ou de reconnaissance des formes.

Dans le cadre de la *segmentation* des images, les individus seront les pixels ou des sous-images, que l'on cherche à classifier, en fonction d'un ou de plusieurs attributs. Ceux-ci ont été présentés au chapitre II. Par exemple, le niveau de gris est un attribut ponctuel, mais les moments ou indices de texture du second ordre calculés sur un voisinage du site peuvent être affectés au site et constituer aussi des attributs ponctuels.

Les classes sont construites à partir des attributs des pixels. Chaque pixel est affecté à une classe unique, mais il est classifié indépendamment de ses voisins. Les critères de proximité employés sont en effet, des distances entre vecteurs d'attributs et ne tiennent pas compte, en principe, de la proximité entre pixels dans l'image. *La classification est une opération préalable à la segmentation* qui est ensuite obtenue par extraction des composantes connexes des pixels appartenant à une même classe. Une région sera constituée de pixels d'une même classe mais il peut y avoir dans l'image plusieurs régions correspondant à une même classe. L'extraction des composantes connexes et l'étiquetage des sites de chaque région par un numéro de région sont présentés dans le paragraphe 5 du chapitre III ; le présent chapitre est donc exclusivement consacré à l'exposé des méthodes de classification.

Les méthodes ne prenant en compte qu'un seul attribut (en général le niveau de gris) sont qualifiées de méthodes *monodimensionnelles*. La plupart d'entre elles

extraient automatiquement des seuils puis affectent les pixels à une classe par comparaison de leur niveau de gris à ces seuils. Elles sont présentées dans le premier paragraphe.

Les méthodes exploitant plusieurs attributs sont qualifiées de *multidimensionnelles*. Elles utilisent des algorithmes d'analyse de données. Elles ont pour rôle d'extraire parmi les nombreux attributs, ceux qui peuvent être discriminants, de déterminer les limites entre les classes puis de classifier les pixels en fonction de ces attributs. Ces différents aspects sont détaillés dans le paragraphe 2.

1 Méthodes monodimensionnelles

Ces méthodes reposent sur l'exploitation de l'histogramme, qu'on supposera normalisé (*cf.* chapitre II § 3) et qui caractérise la distribution des niveaux de gris. Par exemple, une image représentant des objets foncés sur un fond clair présentera un histogramme avec deux modes bien distincts (*cf.* figure 10.1).

Les méthodes monodimensionnelles déterminent des seuils qui constitueront les limites des différentes classes. Ces seuils peuvent être les mêmes pour toute l'image (seuils globaux). Dans ce cas, ils sont déterminés à partir de l'histogramme de l'image complète, par exemple, en recherchant des minimums locaux ou en approximant les modes par des gaussiennes. Dans le cas d'un histogramme unimodal, la recherche des seuils ne peut se faire par le seul examen de l'histogramme global, il faut alors faire appel à des informations locales. Celles-ci peuvent provenir d'un histogramme local ou résulter de l'étude de la compatibilité avec la classification des pixels voisins.

On distingue essentiellement trois catégories de méthodes de *seuillage* [WESK78]. Si on désigne par s un site, par $A[s]$ son niveau de gris et par $p(s)$ le résultat d'un opérateur local agissant sur un voisinage $V(s)$ du site s , le seuillage peut alors être associé à un opérateur $T(s, A[s], p(s))$. On parle alors de :

- seuillage global si T ne dépend que de $A[s]$,
- seuillage local si T ne dépend que de $A[s]$ et de $p(s)$,
- seuillage dynamique si T dépend de $A[s]$, de $p(s)$ et de s .

Dans la plupart des cas, et ce pour des raisons liées aux difficultés techniques et aux domaines d'application, l'opération de seuillage est assimilée à la comparaison à un seuil définissant la limite entre la classe « fond » et la classe « objet ». On parle alors de *binarisation*. L'extension au cas de plusieurs seuils est souvent désignée sous le terme de multiseuillage ou classification.

D'une manière précise, la segmentation par méthode de classification monodimensionnelle se réalise en trois étapes :

- identification des seuils interclasses,
- affectation des points aux différentes classes,

- extraction des composantes connexes de chaque classe.

Quelques techniques de seuillage sont maintenant présentées.

1.1 Seuillage global

1.1.1 Recherche des seuils inter-modes

Dans le cadre précis du seuillage global, l'identification des seuils peut s'effectuer par analyse de l'histogramme des niveaux de gris et recherche des minimums locaux.

La figure 10.1 présente un histogramme qui illustre la séparation interclasse par la détection d'un minimum local. Les seuils de classification sont positionnés sur ces minimums locaux.

Avant d'effectuer la détection des seuils, l'histogramme peut subir des améliorations soit par lissage, soit par approximation (au sens des moindres carrés par exemple).

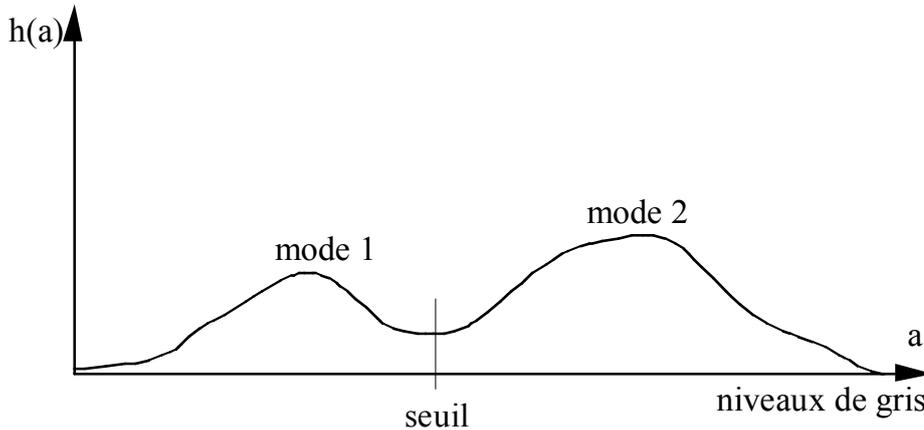


Figure 10.1 : Histogramme bimodal illustrant la présence de deux classes d'intensité dans l'image.

1.1.2 Recherche des modes

La méthode précédente, qui a l'avantage d'être très simple et rapide, est bien adaptée aux images peu bruitées. Mais le bruit introduit sur l'histogramme de nombreuses irrégularités pouvant mettre en défaut la détection de minimums locaux.

On peut alors se tourner vers d'autres méthodes qui modélisent l'histogramme par une somme pondérée de distributions gaussiennes. La méthode de Fisher [FISH58] [DIDA82] [OTSU79] consiste à localiser les seuils comme les séparateurs des distributions ; ceci est réalisé à l'aide d'un critère de minimisation de la somme des inerties associées aux différentes classes. La méthode de Bhattacharya [BHAT67] identifie chaque mode par sa moyenne et sa variance.

La méthode de Fisher nécessite de connaître *a priori* le nombre de classes à détecter, alors que la méthode de Bhattacharya fournit ce nombre de classes.

1.1.2.1 Méthode de Fisher

On recherche N_c classes de niveaux d'intensité, avec pour critère celui de la minimisation de la somme des inerties de chacune des classes.

Soit $\Omega = \{0, \dots, L-1\}$ l'ensemble des niveaux d'intensité de l'image. Soit $P = \{C_1, C_2, \dots, C_{N_c}\}$ une partition de Ω en N_c classes. Le problème consiste à optimiser la partition P en identifiant les séparateurs entre les classes (seuils).



Par la suite, k désignera l'indice courant de niveau de gris, $h(k)$ la composante de rang k du vecteur histogramme et l'entier c sera un compteur de classes.

Le critère d'optimalité consiste à minimiser la somme des inerties des N_c classes, donnée par :

$$W(P) = \sum_{n=1}^{N_c} \sum_{k \in C_n} h(k)(k - G(C_n))^2 \quad (10-1)$$

$$\text{où } G(C_n) = \frac{\sum_{k \in C_n} k h(k)}{\sum_{k \in C_n} h(k)} \quad (10-2)$$

désigne le centre de gravité (valeur moyenne) de la classe C_n .

Les notations suivantes sont utilisées :

$P_a^{(c)}$ désigne une partition de l'intervalle $[a, L-1] \subset \Omega$ en c classes C_1, C_2, \dots, C_c

$G_{a,b}$ désigne le centre de gravité du sous-ensemble $[a, b]$ et $I_{a,b}$ son inertie :

$$G_{a,b} = \frac{\sum_{k=a}^b k h(k)}{\sum_{k=a}^b h(k)} \quad I_{a,b} = \sum_{k=a}^b h(k) (k - G_{a,b})^2 \quad (10-3)$$

Remarque :

I désigne ici l'inertie associée à un intervalle
et W l'inertie associée à une partition.

L'algorithme de calcul de la partition optimale est de type dynamique évaluant une suite de partitions optimales $P_a^{(c)}$ selon le schéma décrit dans l'algorithme ci-dessous :

1ère phase : initialisation

Pour $c = 1$, on considère les ensembles $P_a^{(1)} = [a, L-1]$, pour $a = 1, \dots, L-1$.

Pour chaque niveau a , on calcule l'inertie $W(a) = I_{a, L-1}$ et le centre de gravité $G_{a, L-1}$ du sous-ensemble $P_a^{(1)}$.

2ème phase : phase intermédiaire

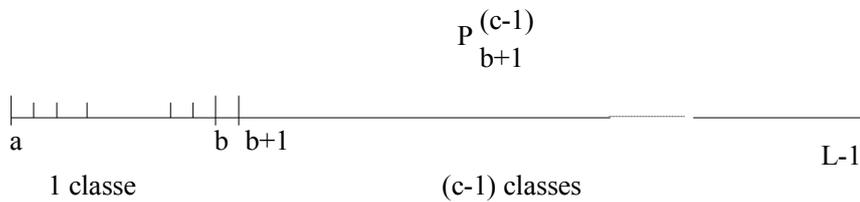
Cette phase intermédiaire comporte N_c-1 étapes examinant successivement des partitions en 2, 3, ..., N_c-1 classes.

Pour $c = 2, \dots, N_c-1$

on calcule la partition $P_a^{(c)}$ du sous-ensemble $[a, L-1]$ selon la formule :

$$P_a^{(c)} = ([a, b], P_{b+1}^{(c-1)})$$

où $P_{b+1}^{(c-1)}$ est la partition de $[b+1, L-1]$ en $c-1$ classes déterminée à l'étape précédente $c-1$.



La valeur de l'indice b , élément de $[a, L-c]$, a été déterminée de façon à minimiser l'inertie de la partition $P_a^{(c)}$ donnée par la formule :

$$W(P_a^{(c)}) = I_{ab} + W(P_{b+1}^{(c-1)})$$

$W(P_{b+1}^{(c-1)})$ correspond à la valeur d'inertie mesurée à l'étape précédente sur les $c-1$ classes de la partition de $[b+1, L-1]$ et I_{ab} correspond à l'inertie de la classe $[a, b]$ introduite lors de l'étape courante.

Phase finale : partition optimale

Pour $c = N_c$, on construit la partition $P_0^{(N_c)} = ([0, a], P_{a+1}^{(N_c-1)})$

Dans ce cas, le seuil a appartient à l'ensemble $\{1, 2, \dots, L-N_c\}$ et il est choisi de façon à minimiser $I_{0a} + W(P_{a+1}^{(N_c-1)})$.

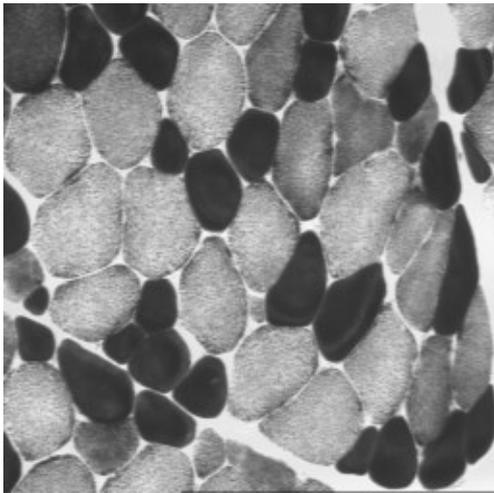
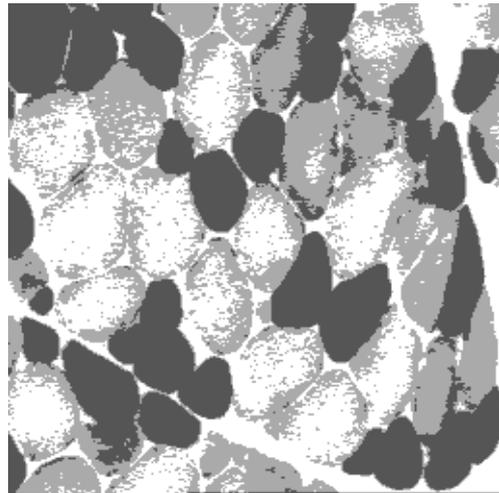
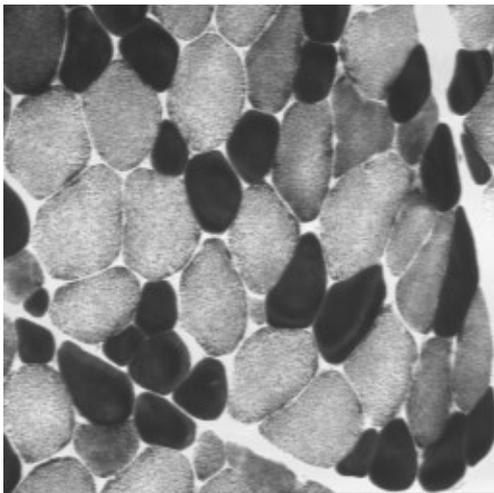


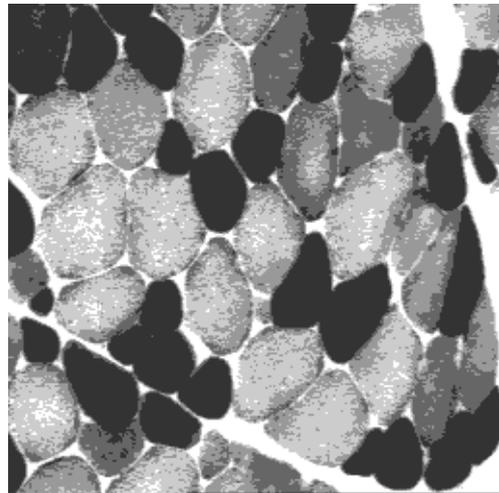
Image originale : MUSCLE



3 classes : seuils 102, 168



4 classes : seuils 93, 147, 194



5 classes : seuils 86, 132, 168, 205

Figure 10.2 : Classification des pixels par la méthode de Fisher en 3, 4 et 5 classes

Un exemple de classification en 3, 4 et 5 classes est donné par la figure 10.2 ; l'histogramme de l'image MUSCLE se trouve en figure 2.2.

Cas particulier : la binarisation

Dans le cas particulier d'une segmentation en 2 classes, la solution peut se formuler comme suit :

Trouver le seuil a entre les classes C_1 et C_2 de manière à minimiser l'inertie $W(\mathbf{P})$ donnée par :

$$W(\mathbf{P}) = \sum_{k \in C_1} h(k)(k - G(C_1))^2 + \sum_{k \in C_2} h(k)(k - G(C_2))^2 \quad (10-4)$$

où $\mathbf{P} = \{C_1, C_2\}$

En développant le membre de droite de l'équation précédente et remarquant que la somme $\sum_{k \in C_1} k^2 h(k) + \sum_{k \in C_2} k^2 h(k)$ est constante, la minimisation de $W(\mathbf{P})$ revient à maximiser la quantité :

$$J(\mathbf{P}) = \frac{\left(\sum_{k \in C_1} k h(k) \right)^2}{\sum_{k \in C_1} h(k)} + \frac{\left(\sum_{k \in C_2} k h(k) \right)^2}{\sum_{k \in C_2} h(k)} \quad (10-5)$$

Le problème de la partition se réduit en la recherche du niveau a dans $[0, L-1]$ tel que $J(\mathbf{P})$ soit maximum.

1.1.2.2 Méthode de Bhattacharya

L'idée de la méthode de Bhattacharya [BHAT67] est d'étudier le comportement du logarithme de la fonction histogramme, au travers de la fonction z définie sur $[1, L]$ par :

$$z(a) = \ln h(a) - \ln h(a - 1) \quad (10-6)$$

et de rechercher les intervalles sur lesquels la fonction $z(a)$ est décroissante. Ceci est illustré par la figure 10.3. Les seuils seront associés aux limites supérieures des intervalles de décroissance.

Pour chacun de ces intervalles maximaux, on effectue une approximation linéaire au sens des moindres carrés de la fonction $z(a)$ par la droite $z = m a + b$ selon les formules suivantes :

$$m = \frac{\sum_{a=\alpha}^{\beta} a z(a) - \frac{\alpha + \beta}{2} \sum_{a=\alpha}^{\beta} z(a)}{\sum_{a=\alpha}^{\beta} a^2 - \frac{1}{\beta - \alpha - 1} \left(\sum_{a=\alpha}^{\beta} a \right)^2} \quad b = \frac{\sum_{a=\alpha}^{\beta} z(a) - m \sum_{a=\alpha}^{\beta} a}{\beta - \alpha - 1} \quad (10-7)$$

où $[\alpha, \beta]$ est l'un de ces intervalles maximaux sur lequel la suite $z(a)$ est décroissante.

Chaque mode c de l'histogramme, associé à un intervalle $[\alpha_c, \beta_c]$ est approximé par une gaussienne dont la valeur moyenne μ_c et l'écart type σ_c sont donnés par :

$$\mu_c = \frac{b}{|m|} + \frac{\delta}{2} \text{ et } \sigma_c = \sqrt{\frac{\delta}{|m|}} \quad (10-8)$$

où δ désigne le pas de quantification des niveaux d'intensité (en général $\delta = 1$).

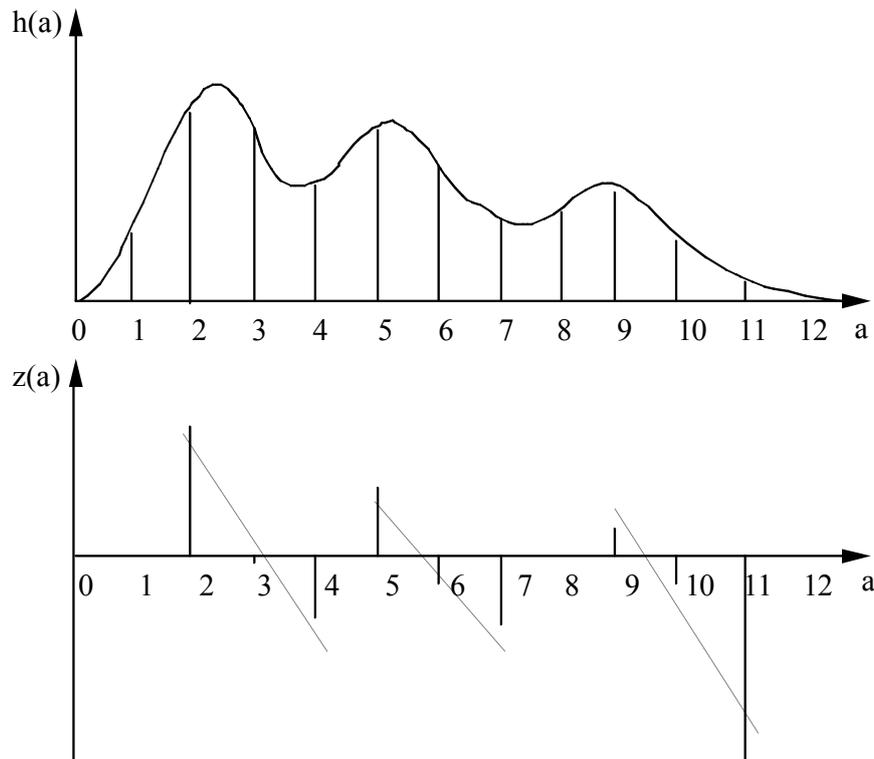


Figure 10.3 : Histogramme trimodal et représentation de la suite $z(a)$ des différences des valeurs logarithmiques ; en pointillé les droites d'approximation, les seuils déterminés sont 4, 7 et 11.

On peut également construire une approximation de l'histogramme initial par une somme de distributions gaussiennes pondérées par un poids adéquat w_c donné par la formule [BART79] :

$$w_c = h(a)\sigma_c\sqrt{2\pi} \exp\left(\frac{1}{2}\left(\frac{a - \mu_c}{\sigma_c}\right)^2\right) \quad (10-9)$$

où a est un niveau d'intensité pris dans l'intervalle $[\alpha_c, \beta_c]$, par exemple celui pour lequel $h(a)$ est maximal.

En procédant de cette manière à partir des modes extrémaux (correspondant aux intensités les plus faibles et les plus fortes), on peut corriger l'histogramme initial et améliorer ainsi l'estimation des paramètres associés aux modes centraux.

Ces méthodes peuvent être étendues par l'utilisation d'histogrammes locaux sur des sous-images de l'image initiale permettant de s'adapter d'une part aux défauts d'homogénéité de l'image globale dus à l'éclairage et d'autre part à la présence de différentes zones d'intérêt dans l'image.

Commentaires :

En complément de ces approches, on peut mentionner l'existence d'autres techniques non liées à un modèle gaussien. En particulier, sur la base d'une distribution intraclasse uniforme, Pun [PUN80] propose de maximiser un critère de somme des entropies intraclasse.

Par ailleurs, il faut noter que la méthode de Fisher peut être vue comme un cas particulier de la méthode des nuées dynamiques [DIDA82] présentée au paragraphe 2.1.1.

1.2 Méthodes locales itératives et adaptatives

Dans le but de rendre les approches précédemment évoquées plus robustes vis-à-vis d'éventuelles variations de la distribution des intensités dans l'image, des travaux ont donné lieu à des approches itératives ou récursives basées sur des techniques de seuillage local.

Quelques exemples de ces méthodes sont maintenant présentés.

1.2.1 Méthode itérative de classification bayésienne

Cette méthode, proposée par Mardia et Hainsworth [MARD88], entre dans la catégorie des techniques locales. Elle consiste à estimer itérativement des paramètres utilisés ensuite pour une classification bayésienne sous hypothèse gaussienne. Elle comprend également un traitement par filtrage linéaire adaptatif destiné à atténuer les effets du bruit. Après avoir rappelé les règles d'affectation utilisées en classification bayésienne et présenté le filtrage linéaire local, l'algorithme itératif de classification sera décrit. Cette méthode requiert la connaissance du nombre de classes. Elle est présentée ici dans le cas de classes ayant la même variance (le cas de variances différentes est traité dans [MARD88]).

1.2.1.1 Classification bayésienne

Lorsqu'il y a deux classes (binarisation) caractérisées par des distributions gaussiennes $G(\mu_i, \sigma^2)$, $i = 1, 2$, et si on affecte le même coût aux différentes erreurs de classification, l'affectation se fait par comparaison de l'intensité du pixel courant au seuil t_{12} :

$$t_{12} = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln \frac{P_2}{P_1} \quad (10-10)$$

où P_1 et P_2 désignent les probabilités de chaque classe ; si celles-ci ne sont pas connues *a priori*, elles peuvent être estimées par une technique itérative décrite ultérieurement.

Le caractère bayésien provient du fait que l'on minimise le risque, défini comme l'espérance mathématique du coût de la décision.

Pour un nombre de classes N_c , de distributions gaussiennes $G(\mu_i, \sigma^2)$, $i = 1, \dots, N_c$, de probabilités *a priori* P_i et numérotées par ordre croissant des moyennes μ_i , on détermine les seuils t_{ij} par :

$$t_{ij} = \frac{\mu_i + \mu_j}{2} + \frac{\sigma^2}{\mu_i - \mu_j} \ln \frac{P_j}{P_i} \quad (10-11)$$

L'affectation se fait alors selon le principe suivant :

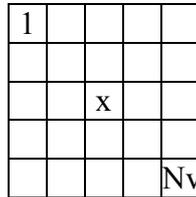
```

i := 1
POUR j variant de 2 à Nc
    SI intensite_pixel > tij ALORS i := j
FIN POUR
affecter le pixel à la classe Ci
    
```

1.2.1.2 Filtrage linéaire local

Afin d'atténuer les effets du bruit, on calcule une moyenne pondérée Y des intensités des N points situés dans un voisinage du pixel courant.

Les voisins de X sont numérotés de 1 à N_v dans la fenêtre d'analyse :



On désigne par $A_n(x)$ l'intensité du pixel numéro n et par γ_n le poids affecté à ce pixel. La moyenne pondérée Y est donc :

$$Y = \sum_{n=1}^{N_v} \gamma_n A_n(x) \quad (10-12)$$

La classe C_i est caractérisée par un processus gaussien de moyenne μ_i et de matrice de covariance \mathbf{P} donnée par l'équation 10-15 où :

$$\text{var } A_k = \frac{1}{\text{card } C_i} \sum_{x \in C_i} (A_k(x) - \mu_i)^2 \quad (10-13)$$

$$\text{et } \text{cov}(A_k, A_j) = \frac{1}{\text{card } C_i} \sum_{x \in C_i} (A_k(x) - \mu_i)(A_j(x) - \mu_i) \quad (10-14)$$

$$P = \begin{bmatrix} \text{var } A_1 & \cdots & \vdots \\ \vdots & \text{cov}(A_k, A_j) & \vdots \\ \cdots & \cdots & \text{var } A_{Nv} \end{bmatrix} \quad (10-15)$$

On notera \mathbf{R} la matrice de covariance normalisée $\mathbf{R} = \frac{1}{\sigma^2} \mathbf{P}$.

Dans ces conditions, la quantité Y suit une loi gaussienne de moyenne μ_i ($\mathbf{t}_1 \boldsymbol{\gamma}$) et de variance $\sigma^2 \mathbf{t}_1 \mathbf{R} \boldsymbol{\gamma}$ avec $\mathbf{t}_1 = (1, 1, \dots, 1)$ et $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n)$

En imposant la contrainte de normalisation $\mathbf{t}_1 \boldsymbol{\gamma} = 1$, l'affectation du pixel à l'une des classes se fait, selon la règle décrite au paragraphe 1.2.1.1, en comparant la quantité Y aux seuils t_{ij}^* donnés par :

$$t_{ij}^* = \frac{\mu_i + \mu_j}{2} + \frac{\sigma^2 \mathbf{t}_1 \mathbf{R} \boldsymbol{\gamma}}{\mu_i - \mu_j} \ln \frac{P_j}{P_i} \quad (10-16)$$

Mardia et Hainsworth proposent d'utiliser un voisinage 3×3 et de modéliser l'image comme un ensemble de processus gaussiens isotropes, c'est-à-dire, tels que les variances soient toutes égales et que la covariance des intensités de 2 pixels distants de d soit donnée par $\sigma^2 \cdot \rho(d)$, avec $\rho(0) = 1$.

Cela nécessite la connaissance des quantités $\rho(1)$, $\rho(\sqrt{2})$, $\rho(2)$, $\rho(\sqrt{5})$ et $\rho(2\sqrt{2})$ qui sont alors issues de l'estimation du semi-variogramme $V(\delta)$ donné par :

$$V(\delta) = \frac{1}{2} E\{(A(s) - A(s'))^2\} = \sigma^2(1 - \rho(\delta)) \quad (10-17)$$

où s et s' désignent deux pixels distants de δ .

On choisit naturellement le vecteur de coefficients $\boldsymbol{\gamma}$ qui minimise la variance de Y :

$$\boldsymbol{\gamma} = \frac{\mathbf{R}^{-1} \cdot \mathbf{1}}{\mathbf{t}_1 \cdot \mathbf{R}^{-1} \cdot \mathbf{1}} \text{ conduisant à } \sigma_Y^2 = \frac{\sigma^2}{\mathbf{t}_1 \cdot \mathbf{R}^{-1} \cdot \mathbf{1}} \quad (10-18)$$

ou éventuellement, pour des raisons de simplicité, une moyenne locale ($\gamma_i = 1/Nv$, $\forall i$) conduisant à la variance suivante :

$$\sigma_Y^2 = \frac{\sigma^2}{Nv^2} \mathbf{t}_1 \cdot \mathbf{R}^{-1} \cdot \mathbf{1} \quad (10-19)$$

1.2.1.3 Algorithme itératif de classification

La méthode décrite ci-dessus est « optimale » mais elle requiert la connaissance des moyennes, covariances et probabilités *a priori* de chaque classe. Ces quantités peuvent être estimées itérativement selon l'algorithme 10.1.

classification initiale en N_c classes à l'aide des quantiles¹ de l'histogramme global.

REPETER

calculer μ_i et n_i , intensité moyenne et nombre de pixels de la classe C_i

extraire les composantes connexes de chaque classe C_i

soustraire aux niveaux de gris des pixels de la classe C_i la moyenne μ_i

estimer le semi-variogramme V sur l'image résultante

déterminer le vecteur optimal γ

calculer les seuils par la relation
$$t_{ij} = \frac{\mu_i + \mu_j}{2} + \frac{\sigma^2}{\mu_i - \mu_j} \ln \frac{n_j}{n_i}$$

pour tous les pixels, calculer $Y = \gamma \square \cdot A$ et affecter le pixel à une classe selon la règle décrite au § 1.2.1.1.

JUSQU'A STABILISATION

Algorithme 10.1 : Classification bayésienne itérative

En toute rigueur, il faut recalculer un vecteur γ' de taille $N_{v'} < N_v$ pour les pixels situés au voisinage des frontières entre composantes connexes de classes différentes.

Dans leurs conclusions, les auteurs mentionnent des nombres d'itérations assez faibles (de l'ordre de 4) et préconisent l'emploi de la moyenne locale pour des raisons de simplicité. L'image finale pouvant néanmoins être bruitée, ils recommandent un post-traitement par filtrage médian. Par ailleurs, ils notent que les meilleurs résultats s'obtiennent pour 2 classes (binarisation).

1.2.1.4 Remarques

Les quantités estimées localement sont les probabilités *a priori* et les moyennes locales. Les variances et covariances peuvent être estimées. Un certain nombre d'informations *a priori* sont nécessaires : nombre de classes, distribution gaussienne et isotrope, covariance stationnaire. Il est à noter que l'hypothèse gaussienne est plus réaliste pour la quantité Y que pour les données initiales A .

1.2.2 Méthode de relaxation-gradient

Cette méthode de classification s'adapte aux images ayant un histogramme global unimodal [BHAN82] [BHAN87]. Le cas de la répartition des pixels en deux classes est d'abord présenté.

¹ Les quantiles sont les valeurs qui découpent une distribution en parties d'effectifs égaux. Par exemple, le premier décile sépare les 10% des pixels les plus sombres des 90% les plus clairs.

1.2.2.1 Binarisation par relaxation-gradient

Les deux classes cherchées sont dénommées C_1 et C_2 . Les sites de l'image sont indexés par la lettre s , $s = 1, \dots, S$.

On appelle \mathbf{P}_s le vecteur de probabilité associé au site s :

$$\mathbf{P}_s = {}^t(p_s(C_1), p_s(C_2)) \quad \text{avec } p_s(C_1) + p_s(C_2) = 1$$

où $p_s(C_i)$ est la probabilité de l'événement « le pixel s appartient à la classe C_i ».

On définit le vecteur de compatibilité

$$\mathbf{Q}_s = {}^t(q_s(C_1), q_s(C_2)) \quad \text{où } q_s(C_i) = \frac{1}{8} \sum_{t \in V_s} p_t(C_i)$$

est la moyenne des probabilités d'appartenance à la classe C_i des 8 voisins du site s .

$$\text{Le critère à maximiser est } J = \sum_{s=1}^S {}^t\mathbf{P}_s \cdot \mathbf{Q}_s \quad (10-20)$$

Les dérivées partielles de J par rapport à C_1 et C_2 sont respectivement $2 \cdot q_s(C_1)$ et $2 \cdot q_s(C_2)$. Après calcul des projections du gradient et simplifications, on aboutit aux équations itératives suivantes :

à l'étape n :

$$\mathbf{P}_s^{n+1}(C_1) = \begin{cases} (1 - \alpha_1)p_s^n(C_1) + \alpha_1 & \text{si } q_s(C_1) > 0,5 \\ (1 - \alpha_2)p_s^n(C_1) & \text{sinon} \end{cases} \quad (10-21)$$

où l'exposant n désigne l'indice d'itération.

Les paramètres α_1 et α_2 contrôlent le degré de lissage à chaque itération, tandis que leur rapport contrôle la tendance à classer en C_1 ou en C_2 .

Si la compatibilité est forte, on renforce la probabilité d'appartenance à la classe C_1 .

Pour une image dont l'intensité évolue entre 0 et $L-1$, les probabilités initiales sont données par :

$$p_s^0(C_1) = \eta \cdot A(s) \frac{A(s) - \mu(A)}{L-1} + 0,5 \quad (10-22)$$

où $\mu(A)$ est la moyenne des intensités de l'image et η est un facteur correctif dépendant de l'intensité $A(s)$ du site courant, valant 1 si $A(s)$ est supérieure à $\mu(A)$, et compris entre 0,5 et 1 sinon.

Avec cette méthode, le seuil de binarisation ne dépend pas seulement de l'intensité du site courant, mais également du résultat de la binarisation de ses

voisins. Les auteurs ne mentionnent pas de stratégie de test de la convergence, mais montrent que peu d'itérations sont nécessaires.

1.2.2.2 Extension à plus de deux classes

L'algorithme présenté ci-dessus est appliqué à l'image entière. On obtient un ensemble de composantes connexes qui sont étiquetées et qui servent de masque binaire de l'image initiale pour effectuer une partition ultérieure. Le processus s'applique récursivement jusqu'à ce que les régions obtenues soient déclarées non partitionnables pour l'une des raisons suivantes :

- l'histogramme reste unimodal
- la région a une surface inférieure à un pourcentage fixé de la surface de l'image (par exemple 1%).

Cette phase est alors suivie d'une étape de fusion de régions et éventuellement d'un filtrage morphologique pour régulariser la forme des régions.

1.2.2.3 Remarques

Cette méthode ne nécessite pas la connaissance du nombre de classes. Elle peut donc s'adapter à des variations de ce nombre dans l'image. Les connaissances *a priori* sont introduites par l'intermédiaire des paramètres α_1 et α_2 , ainsi que par le facteur correctif η qui a, selon les auteurs, une influence sensible sur les résultats.

Cette méthode est à rapprocher des méthodes de relaxation introduites par Rosenfeld [ROSE76].

1.3 Méthodes de seuillage dynamique intégrant des informations de localisation

1.3.1 Méthode de Nakagawa

Cette méthode est une extension, au cas de trois classes [NAKA79], de la méthode de binarisation de Chow et Kaneko [CHOW72] qui est d'abord présentée.

Binarisation

L'image initiale est découpée en blocs d'intersection vide. Dans chaque bloc, on calcule l'histogramme et on cherche si cet histogramme est bimodal de la manière suivante :

si l'écart-type des niveaux de gris sur le bloc est suffisant, l'histogramme local est lissé puis approximé par la somme de deux gaussiennes (*cf.* figure 10.4) de la forme :

$$f(a) = \frac{p_1}{\sigma_1} \exp\left(-\frac{(a - \mu_1)^2}{2\sigma_1^2}\right) + \frac{p_2}{\sigma_2} \exp\left(-\frac{(a - \mu_2)^2}{2\sigma_2^2}\right) \quad (10-23)$$

où μ_i (resp. σ_i) est la moyenne (resp. l'écart-type) des intensités de chaque gaussienne.

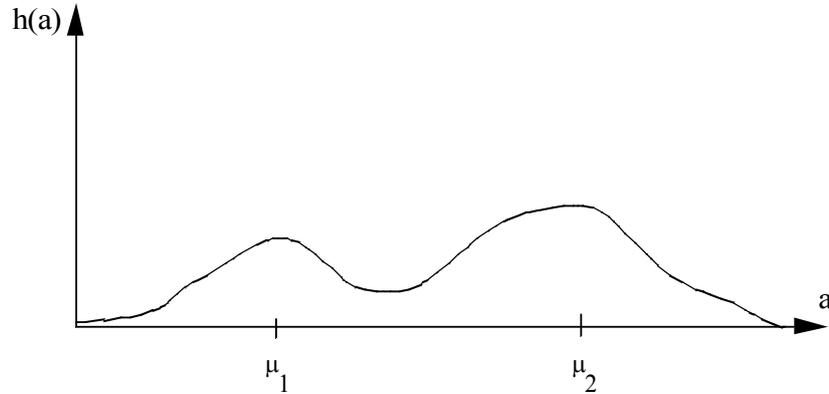


Figure 10.4 : Histogramme approximable par la somme de deux gaussiennes de moyennes μ_1 et μ_2

Cette approximation est toujours possible et fournit les 6 paramètres p_1 , p_2 , μ_1 , μ_2 , σ_1 et σ_2 .

Le bloc sera dit bimodal si les deux gaussiennes sont suffisamment éloignées, c'est-à-dire si :

$$\mu_2 - \mu_1 > 4$$

$$0.1 < \sigma_1/\sigma_2 < 10$$

$$\delta_{12} < 0.8$$

$$\text{où } \delta_{12} = \frac{\min(f(a), a \in [\mu_1, \mu_2])}{\min(f(\mu_1), f(\mu_2))} \quad (10-24)$$

Si la bimodalité est décidée, le seuil t est choisi, entre μ_1 et μ_2 , comme celui qui minimise la probabilité de mauvaise classification pour la distribution f . Le seuil t vérifie l'équation :

$$\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} \right) t^2 + 2 \left(\frac{\mu_2}{\sigma_1^2} - \frac{\mu_1}{\sigma_2^2} \right) t - \frac{\mu_2^2}{\sigma_2^2} + \frac{\mu_1^2}{\sigma_1^2} + 2 \ln \left(\frac{p_2 \cdot \sigma_1}{p_1 \cdot \sigma_2} \right) = 0 \quad (10-25)$$

Le seuil est affecté au centre du bloc. Si le bloc n'a pas été déclaré bimodal, on lui affecte une moyenne pondérée des seuils de ses voisins. Puis, à chaque pixel de l'image initiale, on affecte un seuil par interpolation bilinéaire des seuils des centres des blocs qui l'entourent.

Extension à trois classes

Pour un bloc ayant un écart-type empirique suffisant, on effectue maintenant un ajustement par la somme de trois gaussiennes de l'histogramme local lissé (cf. figure 10.5).

$$f(a) = \sum_{i=1}^3 \frac{P_i}{\sigma_i} \exp\left(-\frac{(a - \mu_i)^2}{2\sigma_i^2}\right) \text{ avec } \mu_1 < \mu_2 < \mu_3 \quad (10-26)$$

Le test de trimodalité consiste à tester la bimodalité de chaque couple de gaussiennes (1,2) et (2,3). Les seuils bas et haut sont obtenus comme pour la binarisation, à l'aide de la relation mentionnée précédemment. Pour chaque bloc déclaré bimodal, on décide d'affecter le seuil t au seuil bas ou au seuil haut selon la proximité. Les seuils bas et haut sont ensuite interpolés et affectés à chaque pixel comme pour la binarisation.

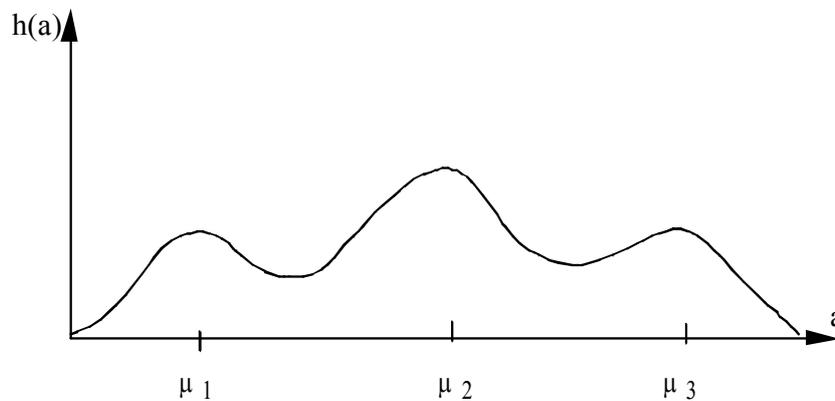


Figure 10.5 : Histogramme à 3 classes

Remarques

L'aspect dynamique du seuillage apparaît clairement par le découpage en blocs et le processus d'interpolation. Cette technique permet de s'adapter à des non stationnarités d'intensité moyenne et d'amplitude similaires à celles des fluctuations.

Le filtrage des fluctuations statistiques se fait par le choix de blocs de taille suffisante et par lissage de l'histogramme de bloc.

Les auteurs montrent que pour des images de pièces manufacturées, la classification en trois classes met souvent en évidence les zones d'ombre. La taille des blocs est 32×32 .



a) Image initiale



b) 1 bloc 256×256



c) 4 blocs 128×128



d) 16 blocs 64×64

Figure 10.6 : Classification adaptative. Méthode de Nakagawa

Exemple

La figure 10.6 présente quelques résultats de classification adaptative par la méthode de Nakagawa. Les tailles des blocs varient de 64×64 à 256×256. Dans chaque bloc, l'histogramme initial a été lissé par un filtre gaussien. Sur l'image 10.6b qui ne compte qu'un seul bloc et pour laquelle les seuils sont globaux, la classification a tendance à morceler les régions (voir par exemple le tuyau d'échappement au niveau de la cabine). Inversement, le découpage en blocs permet de compenser, malgré le nombre limité de classes, les variations globales

d'intensité. Un découpage en blocs trop petits aura tendance à créer de fausses régions.

1.3.2 Utilisation du laplacien

Le principe est d'utiliser le fait qu'au voisinage d'une frontière entre régions, le laplacien passe par deux extremums de signes opposés. Weska *et al.* [WESK74] ont utilisé cette information pour l'élaboration de l'histogramme global. En ne tenant compte que des pixels de fort laplacien, on accentue les vallées de l'histogramme, rendant ainsi la détermination des seuils plus facile. La méthode proposée par Chéhikian [CHEH89] peut être considérée comme une version locale adaptée au cas de la binarisation.

Binarisation

Cette méthode [CHEH89] repose sur un balayage ligne par ligne de l'image. Elle consiste à estimer localement le niveau de référence supérieur (blanc : $B[i, j]$) et le niveau de référence inférieur (noir : $N[i, j]$) à partir du laplacien.

On note $\Delta[i, j]$ la sortie de l'opérateur laplacien au pixel $[i, j]$ (*cf.* chapitre V § 1.1).

Pour le traitement de la ligne i , le calcul de $B[i, j]$ et de $N[i, j]$ s'effectue à l'aide des règles récursives suivantes :

$$B[i, j] = \begin{cases} A[i, j] & \text{si } \Delta[i, j] < 0 \\ B[i, j-1] & \text{sinon} \end{cases}$$

$$N[i, j] = \begin{cases} A[i, j] & \text{si } \Delta[i, j] > 0 \\ N[i, j-1] & \text{sinon} \end{cases}$$

En effet, un laplacien négatif correspond à un gradient d'intensité décroissant (*cf.* figure 5.1). L'intensité du pixel courant est dans ce cas prise comme nouveau niveau de référence haut, la référence bas étant inchangée. Un laplacien positif correspond au contraire à un gradient d'intensité croissant.

Le seuil local pour la binarisation est alors donné par la relation :

$$t_{i,j} = \frac{N_{i,j} + B_{i,j}}{2} \quad (10-27)$$

Pour atténuer les effets du bruit d'image, la binarisation se fait en comparant une version filtrée passe-bas de \mathbf{A} (notée \mathbf{AF}) avec le seuil local t .

Ces règles de base sont sensibles au bruit d'image, surtout dans les zones stationnaires correspondant généralement au fond. Elles sont en fait modifiées en prenant en compte l'image \mathbf{AF} au lieu de l'image initiale \mathbf{A} , et en imposant un seuil minimal entre \mathbf{AF} et les niveaux de références N et B pour la remise à jour.

Remarques

Cette méthode a été développée dans un contexte de lecture optique de caractères où les contrastes, la taille des objets et le niveau du fond sont

relativement bien connus *a priori*. Elle ne semble pas pouvoir facilement s'étendre à un nombre plus élevé de classes. L'auteur préconise l'emploi de filtres gaussiens pour le lissage suivi du calcul du laplacien, ceci introduit une plus grande robustesse vis-à-vis du bruit ainsi qu'il est démontré dans le chapitre V (§ 1.8.2.3) et l'annexe D.

Cette technique fait appel à des informations locales (résultant des opérateurs de filtrage passe-bas et de laplacien). L'aspect dynamique provient de la prise en compte des niveaux de référence du pixel voisin lors de l'application des règles de remise à jour ou de leur version modifiée.

1.3.3 Méthode de Hertz : coopération contour/multiseuillage

Partant d'un découpage initial de l'image en blocs d'intersection non vide, cette méthode [HERT88] consiste à ajuster, dans chaque bloc, un ensemble de seuils de façon à faire coïncider au mieux l'ensemble des points-contours de l'image multiseuillée et celui déterminé par un détecteur de contours de type gradient. C'est donc une méthode faisant coopérer les résultats d'une détection de contour et d'une première classification. Le nombre de classes pour ce multiseuillage est fixé ou déterminé à l'aide de l'histogramme global. L'histogramme global fournit également le jeu initial de seuils (*cf.* § 1.1).

Soit **AG** l'image binaire des points contours obtenus par un opérateur gradient et **AS** l'image binaire des contours des composantes connexes obtenues par un multiseuillage.

La méthode repose sur l'analyse de l'image de différence $\mathbf{D} = \mathbf{AG} - \mathbf{AS}$ dont les pixels prennent les valeurs -1, 0 ou +1, en utilisant les règles ci-dessous :

- a/ Si un bloc de l'image **D** contient un grand nombre de pixels à 1 et -1, cela correspond à la présence d'un vrai contour mal localisé. Ceci se corrige par ajustement d'un seuil. Pour tous les pixels tels que $D[s] = +1$, déterminer le seuil le plus proche de l'intensité initiale de s . Pour chaque seuil t_i associé à un nombre suffisant de pixels tels que $D[s] = +1$, on détermine la valeur moyenne a_m de ces pixels. On recherche alors dans l'intervalle $[a_m - 10, a_m + 10]$ la valeur t_i' qui minimise le nombre de pixels différents entre l'image **AG** et l'image **AS**. Le seuil t_i' remplace alors t_i .
- b/ Si un bloc contient un grand nombre de pixels à -1 et peu de pixels à +1, un faux contour a été introduit par un seuil superflu qu'il faut éliminer de la liste.
- c/ Si un bloc contient un grand nombre de pixels à +1 et peu de pixels à -1, un vrai contour a été détruit par la fusion de deux classes. Il faut introduire un nouveau seuil. Soit a_m la moyenne des pixels du bloc tel que $D[s] = +1$, soit t_k le seuil le plus proche de a_m , remplacer t_k par a_m .

Ici, la variation spatiale du seuillage provient essentiellement du découpage en blocs. Par ailleurs, lorsque l'étape d'ajustement échoue, c'est-à-dire que le seuil n'y est pas modifié, la nouvelle valeur est fixée comme étant la moyenne des valeurs

obtenues sur les blocs voisins. Le chevauchement des blocs introduit également un effet de dépendance spatiale.

Les exemples présentés par les auteurs concernent des images obtenues en contrôle de circuits imprimés et en production manufacturière. Le nombre de classes est faible (typiquement 2 ou 3). La détection de contours est faite par un opérateur de type gradient morphologique (voir chapitre V).

Le choix de l'action à effectuer (a-c) dépend de seuils qui sont fixés par l'opérateur et qui sont liés au type d'image et aux conditions d'acquisition. Il en va de même de la taille des blocs (32×32 avec un recouvrement de largeur 8 dans les exemples présentés) qui dépend en particulier de l'échelle des non stationnarités dans l'image.

1.4 Discussion

Les méthodes adaptatives visent à faire varier, en fonction de la position du pixel courant, les valeurs de seuils de classification et éventuellement le nombre de classes. Cette variation peut être introduite explicitement par découpage en blocs et interpolation des seuils, ou bien implicitement par seuillages itératifs ou récursifs.

Les méthodes adaptatives font appel, dans leur principe, à une hypothèse d'unimodalité de la distribution des intensités dans une région homogène, cette distribution étant souvent considérée comme gaussienne. Ceci rend nécessaire le choix *a priori*, par l'opérateur, de valeurs numériques pour le test d'unimodalité.

La méthode de relaxation-gradient vise à déterminer le nombre de classes. Dans la pratique, il semble que ce nombre reste faible sur les images présentées.

Les méthodes visant à déterminer la loi de variation spatiale des seuils (Nakagawa, Chéhikian) fonctionnent correctement si la variation de gradient entre deux régions est assez douce (*i.e.* à grande échelle spatiale) vis-à-vis des dimensions des zones de transition entre régions. Le nombre de classes doit être faible. De plus, des zones contiguës de l'image doivent posséder un nombre de classes assez proche.

Notons enfin que la plupart des auteurs préconise l'emploi d'un post traitement (filtrage médian ou morphologique) pour améliorer les performances de la classification.

Il est important de mentionner que différents articles de synthèse ont été consacrés à la segmentation au travers desquels les méthodes ici présentées sont largement évoquées [FU81], [HARA85], [SAHO88].

1.5 Conclusion

Les opérations de classification visent à simplifier l'image à analyser, tout en conservant le maximum d'information sur la géométrie des objets qui y sont représentés. Elles ne donneront donc de bons résultats que si les pixels proches géographiquement (situés dans une même région), le sont également dans l'espace

des mesures (intensité) et si des pixels appartenant à des régions différentes sont éloignés dans l'espace des mesures.

Des méthodes de seuillage local et dynamique permettent d'effectuer cette classification même si les non stationnarités de l'image rendent l'histogramme global inexploitable. Ces méthodes utilisent des informations locales (voisinage) ou semi-locales (bloc ou composante connexe). Elles nécessitent d'y ajouter des opérations amont (prétraitement) ou aval (suppression de petites régions). Une plus grande sophistication de la classification se traduit par une amélioration des performances. Il ne faut cependant pas perdre de vue que ces techniques doivent, pour conserver leur intérêt, rester simples et rapides. Dans le cas contraire, il est sans doute plus performant de repartir d'un modèle de l'image faisant apparaître explicitement les dépendances entre pixels voisins.

D'une manière générale, le bon fonctionnement de ces méthodes de choix automatique de seuils repose sur l'emploi d'autres seuils fixés par l'opérateur en fonction de sa propre expérience. Ceci signifie qu'il est vraisemblablement délicat de transposer ces techniques en dehors du contexte de leur mise au point.

2 Méthodes multidimensionnelles

Les méthodes multidimensionnelles consistent à classifier des individus en fonction non plus d'un seul attribut (en principe le niveau de gris) comme dans les paragraphes précédents, mais en fonction d'un ensemble d'attributs (*cf.* chapitre II).

On utilisera dans tout ce paragraphe les notations suivantes :

E est l'ensemble des individus ($\text{card } E = n$).

C est l'ensemble des classes ($\text{card } C = N_c$).

Q est l'ensemble des attributs ($\text{card } Q = q$).

A chaque individu est associé un vecteur de q attributs. C'est dans l'espace des attributs ou *espace de représentation* (de dimension q) que la classification s'effectue.

Il existe deux types de méthodes de classification multidimensionnelle.

La *classification supervisée*, ou *classification avec apprentissage* consiste à construire, à partir d'une classification connue *a priori* d'un certain nombre d'individus, une *fonction d'identification* ou de *discrimination* pour les autres individus. Cette fonction d'identification réalise un découpage de l'espace de représentation. A chaque zone de ce découpage est affectée une classe de la classification *a priori*. Les autres individus sont ensuite classifiés en fonction de leur position dans l'espace des paramètres. Des méthodes de classification supervisée appliquées à l'image sont présentées dans le paragraphe 2.2.

La *classification non supervisée*, consiste à découper l'espace de représentation en zones homogènes selon un critère de ressemblance entre les individus, c'est-à-dire selon un critère de proximité dans l'espace des attributs. Des méthodes de ce type sont exposées dans le paragraphe 2.1.

Si on a des connaissances sur la classe d'appartenance de certains individus, on utilisera le premier type de méthodes. Si ce n'est pas le cas, il faut définir une mesure de proximité entre les attributs et utiliser la deuxième approche.

On appellera *méthode paramétrique*, une méthode qui fait appel à une hypothèse *a priori* sur l'expression analytique de la fonction de représentation (par exemple, l'analyse factorielle discriminante suppose que la fonction de répartition des classes est gaussienne ; les paramètres à estimer sont dans ce cas la moyenne et la variance).

Quel que soit l'algorithme adopté, le choix des attributs à prendre en compte est très important. En effet, si l'utilisateur a des connaissances *a priori* sur la pertinence des attributs qu'il peut calculer sur la région, il peut se limiter aux plus judicieux. En l'absence d'information fiable sur les attributs, il faut faire appel à des méthodes de sélection automatique des attributs qui sont présentées dans le paragraphe 2.3.

2.1 Classification non supervisée

Les méthodes de classification non supervisée ont pour but de trouver des *partitions* d'un ensemble d'individus en fonction de critères de proximité de leurs vecteurs d'attributs dans l'espace de représentation. Elles sont utilisées pour effectuer de la classification en aveugle et, ainsi réaliser une segmentation sans connaissances *a priori* sur l'image [GAGA88]. Il est par conséquent nécessaire de prendre quelques précautions. La mesure de proximité est calculée sur tous les attributs, il n'y a donc pas de réduction de l'espace de représentation. Comme il n'y a pas d'exemples d'apprentissage, si certains attributs utilisés sont non représentatifs des classes recherchées, le résultat pourra être médiocre, puisque le partitionnement obtenu dépendra aussi de ces attributs (*cf.* figure 10.7).

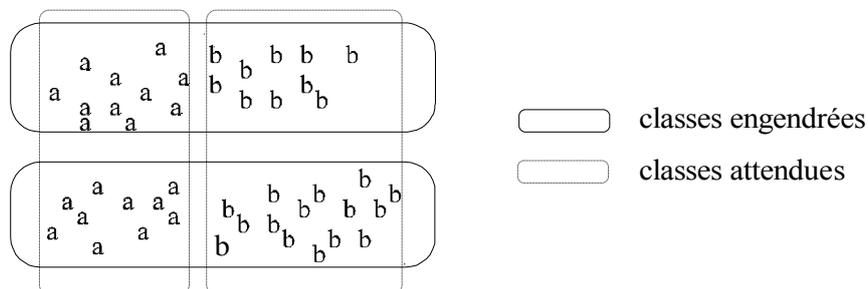


Figure 10.7 : Individus de deux types (a et b) représentés dans un espace de deux attributs. L'attribut sur l'axe vertical modifie la dispersion des individus et crée des classes factices.

2.1.1 Nuées dynamiques

La méthode des nuées dynamiques consiste à calculer la partition optimale de l'ensemble des individus en N_c sous-ensembles, chaque sous-ensemble (ou classe) étant représenté par un « noyau » (cf. figure 10.8). Proposée par Diday, cette méthode est largement développée dans [DIDA82].

On note :

Π , l'ensemble des partitions π de E en N_c classes (C_1, \dots, C_{N_c}),

$\mathbf{g} = (g_1, \dots, g_{N_c})$ un ensemble de N_c noyaux, où g_i est le noyau de la classe C_i .

$W(\pi, \mathbf{g})$ un critère qui mesure l'adéquation entre une partition π et l'ensemble des noyaux \mathbf{g} .

On cherche :

$$\left(\pi^*, \mathbf{g}^* \right) = \underset{(\pi, \mathbf{g})}{\text{Arg min}} \{W(\pi, \mathbf{g})\} \quad (10-28)$$

c'est-à-dire la partition π et la représentation \mathbf{g} associée qui optimisent leurs adéquations au sens du critère W .

Nous présentons ici les nuées dynamiques dans le cas où les noyaux ont comme mode de représentation le centre d'inertie de la classe C_i . Le critère d'adéquation est alors :

$$W(\pi, \mathbf{g}) = \sum_{i=1}^{N_c} \sum_{x \in C_i} d^2(x, g_i) \quad (10-29)$$

où d représente la distance euclidienne. W est donc la variance intraclasse. On définit alors deux fonctions :

- une fonction de représentation $h : \pi \rightarrow \mathbf{g}$ qui, connaissant une partition π , calcule la représentation unique \mathbf{g} associée (en calculant chaque centre de gravité g_i de C_i).
- une fonction d'affectation $f : \mathbf{g} \rightarrow \pi$ qui, connaissant la représentation \mathbf{g} , calcule la partition unique π associée, en calculant chaque classe C_i telle que :

$$C_i = \{x \in E, \forall j \neq i, d(x, g_i) < d(x, g_j)\} \quad (10-30)$$

et si x est équidistant de g_i et g_j , il est intégré à la classe de plus faible indice.

L'initialisation se fait en évaluant ou en tirant au hasard un ensemble \mathbf{g}^0 de noyaux (ou une partition π^0). La convergence et la stationnarité de la suite u_n définie ci-dessous sont aisément démontrables.

$$\begin{cases} g^{n+1} = f(g^n) \\ g^{n+1} = h(g^{n+1}) \\ u_{n+1} = W(g^{n+1}) \end{cases} \quad (10-31)$$

La suite u_n converge en décroissant vers la valeur optimale u^* . La partition π^* et la représentation g^* associées à u^* sont optimales pour un g^0 donné et fournissent la solution au problème.

Il est important de noter que la partition obtenue n'est pas indépendante de g^0 . Le choix des noyaux initiaux peut donc s'avérer crucial.

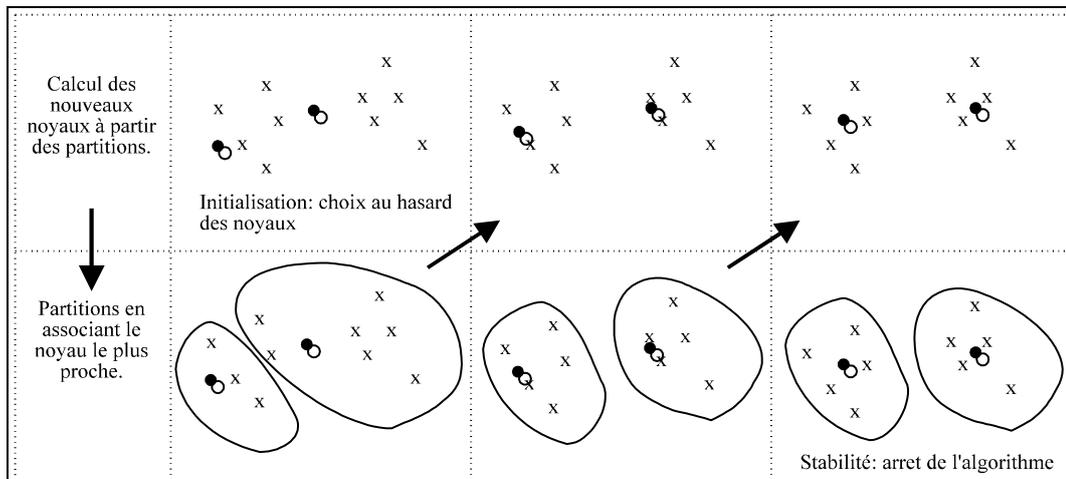


Figure 10.8 : Construction de 2 classes par l'algorithme des nuées dynamiques

En pratique, le nombre de classes N_c n'est pas connu *a priori*. L'algorithme suivant propose une solution intéressante à ce problème.

Algorithme de Coleman et Andrews

Cet algorithme [COLE79] est une méthode complète de segmentation basée sur les nuées dynamiques et comprenant la détermination du nombre de classes et le choix des attributs.

Une première phase opère une décorrélation des attributs par la transformée de Karhunen-Loeve présentée au paragraphe 2.3.1.

Puis, l'algorithme des nuées dynamiques est appliqué avec un nombre de classes k croissant. L'adéquation entre la partition en k classes et les données est calculée, à chaque étape, en utilisant la valeur de $\beta = \text{Tr}(\mathbf{B}_k) \cdot \text{Tr}(\mathbf{W}_k)$ où \mathbf{B}_k et \mathbf{W}_k sont respectivement les matrices de variance interclasse et intraclasse (cf. § 2.2.1.2) de la partition obtenue et Tr indique la trace de la matrice. Cette valeur mesure la séparabilité des partitions obtenues. β passe par un maximum quand k varie, N_c est choisi comme étant la valeur de k pour laquelle ce maximum est atteint.

La distance de Bhattacharya [BHAT67] est ensuite utilisée pour réduire le nombre d'attributs. Sa définition générale est donnée par l'équation (10-62). Dans le cas de distributions gaussiennes, la distance de Bhattacharya entre deux classes C_i et C_j , est définie, pour un attribut donné, par :

$$B(C_i, C_j) = \frac{1}{4} \ln \left(\frac{1}{4} \left(\frac{\sigma_i^2}{\sigma_j^2} + \frac{\sigma_j^2}{\sigma_i^2} + 2 \right) \right) + \frac{1}{4} \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \quad (10-32)$$

où μ_i (resp. μ_j) est la moyenne de l'attribut dans la classe C_i (resp. C_j)

σ_i (resp. σ_j) est l'écart-type de l'attribut dans la classe C_i (resp. C_j).

On applique la méthode de Fisher (cf. § 1.1.2) à chacun des attributs séparément. L'attribut n'est conservé que si la moyenne des distances entre les classes est suffisamment grande.

L'algorithme des nuées dynamiques est finalement appliqué sur les attributs sélectionnés et avec un nombre de classes optimal.

2.1.2 Classification ascendante hiérarchique

Le but de la classification ascendante hiérarchique est de construire une arborescence dont la racine est l'ensemble E , les feuilles sont les individus x et qui soit une représentation de la proximité entre les individus [CAIL76] [DIDA82].

Soit H un ensemble de parties (appelées paliers) de E , H est une *hiérarchie* (présentée sous forme d'arborescence figure 10.9) si :

- $E \in H$, et $\forall x \in E, \{x\} \in H$
- si deux paliers ne sont pas disjoints, alors l'un contient l'autre

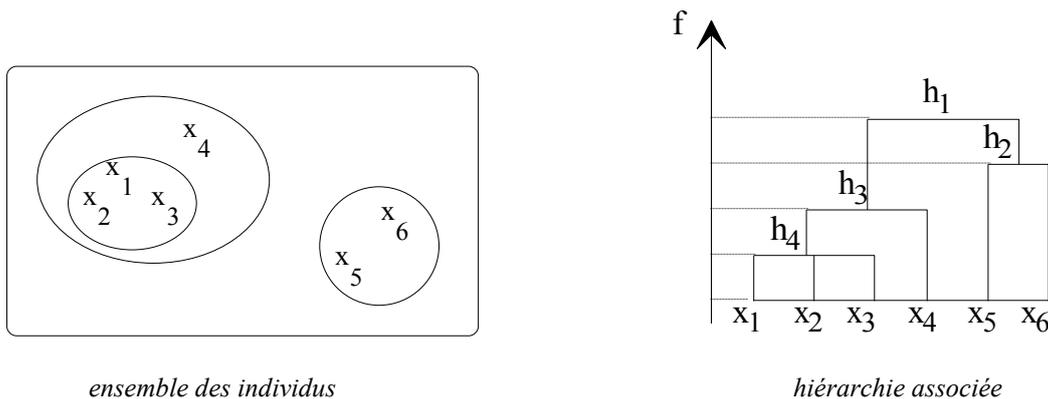


Figure 10.9 : Exemple de hiérarchie indicée $H = \{h_1=E, h_2, h_3, h_4, \{x_1\}, \dots, \{x_6\}\}$

Une *hiérarchie indicée* est un couple (H, f) où H est une hiérarchie sur un ensemble fini E et f une application de H dans \mathbb{R}^+ telle que :

$$\begin{cases} f(h)=0 \Leftrightarrow \text{card}h=1 \\ \forall h, h' \in \mathbf{H}, h \subset h' \text{ et } h \neq h' \Rightarrow f(h) < f(h') \end{cases} \quad (10-33)$$

Un résultat fondamental est qu'une hiérarchie indicée est équivalente à une ultramétrie. Une distance δ est dite *ultramétrique* si :

$$\forall x, y, z \in \mathbf{E} \quad \delta(x, y) \leq \max(\delta(x, z), \delta(y, z)) \quad (10-34)$$

ce qui signifie que tout triangle est isocèle, le troisième côté étant le plus petit.

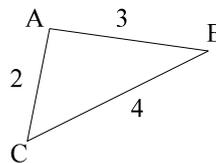
Alors, dans un triangle, on peut soit :

- ne pas agréger,
- agréger deux sommets (le plus petit côté),
- agréger les trois sommets

Avec une distance ultramétrique, on peut réaliser un partitionnement de l'ensemble de la manière suivante : x et y sont agrégés dans la même classe si $\delta(x, y) < S$. Le partitionnement peut être obtenu pour n'importe quelle valeur de S . Par contre, si la distance n'est pas ultramétrique, ceci n'est généralement pas possible, comme le montre l'exemple ci-dessous :

Exemple :

Si on prend $S = 3.5$, deux agrégations sont possibles : (A,B) et (A,C).



Mais B et C doivent être dans deux classes différentes, ce qui rend impossible le partitionnement de ces 3 points pour $S = 3.5$.

Pour régler ce problème, il suffit de transformer la distance initiale d (par exemple la distance euclidienne) en une ultramétrie δ . Le problème de la classification hiérarchique est alors de construire la hiérarchie indicée associée à l'ultramétrie δ^* la plus « proche » de la distance d initiale.

On peut par exemple chercher l'ultramétrie

$$\delta^* = \underset{\delta \in U}{\text{Arg min}} \{ \Delta(d, \delta), \delta \leq d \} \quad (10-35)$$

où

U est l'ensemble des ultramétries,

Δ est une mesure d'adéquation entre d et δ .

$$\delta \leq d \text{ signifie } \forall a, b \in \mathbf{E} ; \delta(a, b) \leq d(a, b) \quad (10-36)$$

Une des stratégies les plus utilisées est de prendre l'ultramétrie sous-dominante δ_d , borne supérieure de l'ensemble des ultramétries inférieures à d . Elle est définie par :

$$\forall a, b \in E, \delta_d(a,b) = \max \{ \delta(a, b) / \delta \in U, \delta \leq d \} \quad (10-37)$$

Par exemple, l'algorithme 10.2 permet de construire une hiérarchie à partir d'un critère d'agrégation D , fonction de la distance d entre les points.

1. Partir de la partition maximale de E où chaque élément est une classe.
2. Agréger les deux classes qui minimisent D (s'il y a plusieurs couples possibles, en choisir un au hasard). La valeur de f pour le palier est cette valeur de D .
3. Recommencer 2 jusqu'à ce que toutes les classes se réunissent en une seule.
4. Regrouper les paliers h et h' tels que $f(h) = f(h')$ et $h \subset h'$ (ce qui correspond à regrouper les classes adjacentes du 2 qui ont des indices d'agrégation égaux). L'application f se généralise avec $f(h \cup h') = \max(f(h), f(h'), D(h, h'))$.

Algorithme 10.2 : Construction d'une hiérarchie basée sur la distance entre classes

Si le critère d'agrégation D est égal à

$$D(h, h') = \min_{a \in h, b \in h'} d(a, b) \quad (10-38)$$

alors l'ultramétrie construite est l'ultramétrie sous-dominante et ce critère s'appelle critère d'agrégation du lien minimum. Dans ce cas, la hiérarchie indiquée obtenue est unique. Cette ultramétrie induit un effet de chaîne : deux points très éloignés sont considérés comme proches s'il existe une chaîne de points proches les reliant l'un à l'autre, ce qui correspond bien à la notion intuitive que l'on possède d'une classe. Cela suppose également que les classes sont suffisamment séparées pour ne pas engendrer d'artefact.

Pour obtenir une partition en N_c classes, il suffit de monter dans la hiérarchie jusqu'à ce qu'il ne reste plus que N_c sous-ensembles. Ou bien, on choisit une valeur S de l'indice d'agrégation et dans ce cas, la partition sélectionnée est la partition ayant le plus grand nombre de classes et dont la valuation est inférieure à S (cf. figure 10.10).

Cette méthode est sensible au bruit : l'adjonction d'individus peut conduire à des modifications importantes dans la hiérarchie.

Une application intéressante de cette méthode est proposée par Gagalowicz et Graffigne [GAGA88] pour la segmentation texturale non supervisée. Ils découpent l'image en fenêtres carrées et prennent comme attributs de chaque fenêtre l'ensemble des moyennes d'espace $MC_t(i,j)$ du second ordre pour toutes les translations t de la fenêtre (cf. § 2.4). Le défaut de l'algorithme général précédent est qu'il ne prend pas en compte la proximité des individus dans l'image. Pour pallier ce défaut et réduire le nombre de distances calculées à l'étape 2, ils ne

calculent les distances qu'entre fenêtres voisines et n'agrègent donc que des fenêtres adjacentes.

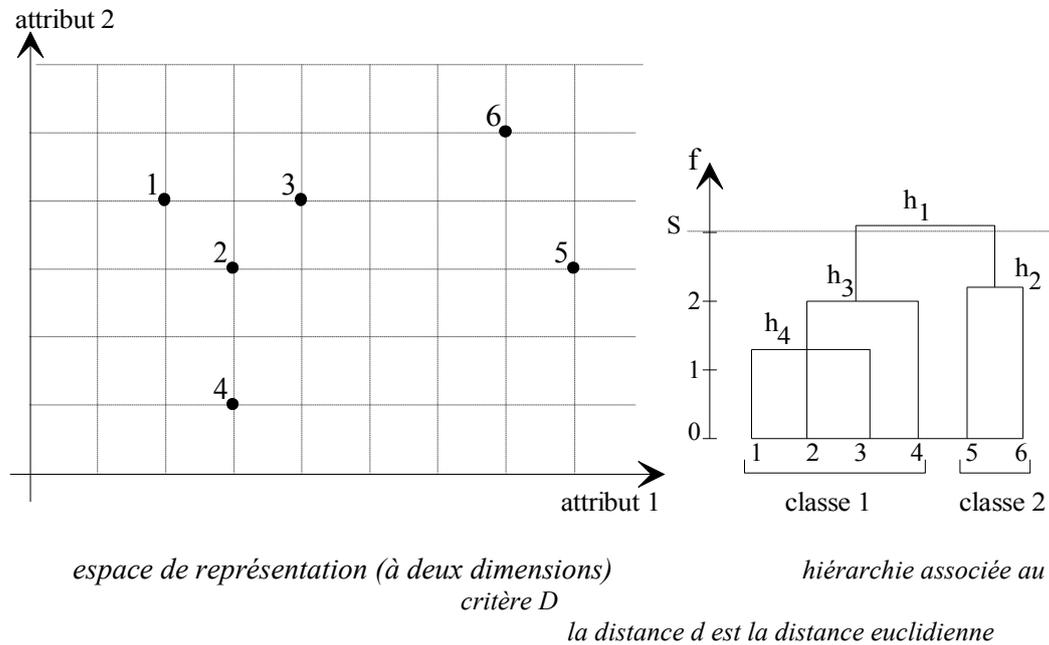


Figure 10.10 : Exemple de classification à partir d'une hiérarchie

2.2 Classification supervisée

Les méthodes de classification supervisée supposent la connaissance *a priori* de l'appartenance de chaque échantillon de l'ensemble d'apprentissage à une classe donnée, ce qui revient à supposer une connaissance *a priori* sur l'image à segmenter.

2.2.1 Construction d'une fonction discriminante

Ces méthodes permettent de déterminer quel est le découpage de l'espace des attributs, qui permet d'assurer au mieux (suivant un critère), la discrimination. L'affectation des individus de l'ensemble d'apprentissage est connue *a priori*. Il est cependant nécessaire de préciser que :

- comme dans la classification non supervisée, on utilise un espace de représentation figé (la dimension de l'espace paramétrique est fixe), l'utilisation d'attributs non représentatifs est donc problématique. Le bruit engendré par ces attributs modifie les dispersions et perturbe ainsi les résultats. Il est donc recommandé de faire une première sélection des attributs (*cf.* § 2.3).
- ces méthodes utilisent des surfaces de décision qui sont fortement contraintes. Par exemple, pour l'analyse factorielle, les surfaces de décision sont les ellipsoïdes d'inertie. Comme ces méthodes ne prennent pas en compte les dispersions, elles ont parfois pour effet de créer des attracteurs en des points qui n'appartiennent pas à la classe (par exemple, si les individus suivent une loi uniforme sur un cercle, un attracteur va se former au centre du cercle).

2.2.1.1 K-plus proches voisins

La méthode des K-plus proches voisins (ou Kppv) est un moyen simple d'estimation non paramétrique de densité. Pour calculer l'estimation de la densité r_i de la classe C_i au point x , on recherche les K (K fixé à l'avance) plus proches voisins de x dans un ensemble de référence (l'ensemble d'apprentissage dont l'affectation des individus est connue *a priori*). L'estimation de la densité est donnée par :

$$\hat{r}_i(x) = \frac{K_i(x)}{n_i V(x)} \quad (10-39)$$

où $K_i(x)$ est le nombre de points de C_i appartenant aux Kppv de x ,

n_i est le cardinal de la classe C_i ,

$V(x)$ est le volume de la plus petite boule contenant les Kppv de x .

Cette méthode se simplifie en méthode de discrimination par voisinage en affectant à x la classe majoritaire parmi les Kppv. Le résultat de la discrimination dépend de la valeur de K. C'est pourquoi il est intéressant de faire varier K afin d'obtenir les meilleurs résultats possibles. Par exemple, si une classe est fortement concentrée, celle-ci se comportera comme un attracteur si la valeur de K est grande. A l'inverse, le vote risque d'être perturbé par le bruit si la valeur de K est faible.

Les principaux inconvénients de cette méthode sont le coût de stockage (les éléments de l'ensemble d'apprentissage doivent être stockés) ainsi que le coût élevé de la recherche des Kppv (de complexité en $O(K^2 n \log n)$ où n est la taille de l'ensemble d'apprentissage) [PREP88]. De nombreuses idées pour améliorer cette méthode sont proposées dans [DUBU90].

2.2.1.2 Analyse factorielle discriminante

L'analyse factorielle discriminante a pour but de trouver les axes factoriels qui discriminent au mieux les N_c classes de l'ensemble d'apprentissage, c'est-à-dire tels que les projections (sur ces axes) des individus d'une même classe soient les plus concentrées possible alors que celles des individus de classes différentes sont les plus éloignées possible [DIDA82] [AURA90]. Ces nouveaux attributs (projections des individus sur ces axes) sont des combinaisons linéaires des attributs de départ et doivent être les plus décorrélés possible.

L'ensemble d'apprentissage E est stocké dans une matrice X de n lignes (représentant les individus) et de q colonnes (représentant les attributs). Un individu est noté $\mathbf{x}_i = (x_i^j)_{1 \leq j \leq q}$, vecteur de ses q attributs. Les valeurs du $j^{\text{ème}}$ attribut sont stockées dans le vecteur $\mathbf{x}^j = (x_i^j)_{1 \leq i \leq n}$.

Chaque individu appartient à une classe C_k , $k = 1, \dots, N_c$. On note :

$\bar{x}^j = \sum_{i=1}^n x_i^j$ la moyenne de l'attribut j,

$\bar{\mathbf{x}}$ le vecteur des moyennes de chaque attribut,

$\bar{\mathbf{x}}_k$ le vecteur des moyennes de chaque attribut des individus de C_k ,

n_k le cardinal de C_k .

Soient \mathbf{T} , \mathbf{B} et \mathbf{W} les matrices de variances totale, interclasse et intraclasse associées à E et C. Ce sont des matrices (q,q) qui se définissent par :

$$T(u, v) = \text{cov}(\mathbf{x}^u, \mathbf{x}^v) = \frac{1}{n} \sum_{i=1}^n (x_i^u - \bar{x}^u)(x_i^v - \bar{x}^v) \quad (10-40)$$

$$\mathbf{T} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})' \quad (10-41)$$

$$\mathbf{W} = \frac{1}{n} \sum_{k=1}^{N_c} \sum_{m \in C_k} (\mathbf{x}_m - \bar{\mathbf{x}}_k) (\mathbf{x}_m - \bar{\mathbf{x}}_k)' \quad (10-42)$$

$$\mathbf{B} = \frac{1}{n} \sum_{i=1}^{N_c} n_i (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}) (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})' \quad (10-43)$$

Ces trois matrices sont liées par la relation :

$$\mathbf{T} = \mathbf{W} + \mathbf{B} \quad (10-44)$$

Le problème est de déterminer le système orthonormé de vecteurs \mathbf{u}_i , $i = 1, \dots, r$ de \mathbb{R}^q , formant une base du sous-espace vectoriel dans lequel l'inertie intraclasse est minimale et l'inertie interclasse maximale. On cherche donc les vecteurs \mathbf{u}_i qui minimisent $\frac{{}^t \mathbf{u}_i \mathbf{W} \mathbf{u}_i}{{}^t \mathbf{u}_i \mathbf{T} \mathbf{u}_i}$ sous la contrainte de normalisation :

$${}^t \mathbf{u}_i \mathbf{T} \mathbf{u}_i = 1 \quad (10-45)$$

$$\text{Comme } {}^t \mathbf{u}_i \mathbf{T} \mathbf{u}_i = {}^t \mathbf{u}_i \mathbf{B} \mathbf{u}_i + {}^t \mathbf{u}_i \mathbf{W} \mathbf{u}_i, \quad (10-46)$$

ceci est équivalent à déterminer les \mathbf{u}_i qui maximisent ${}^t \mathbf{u}_i \mathbf{B} \mathbf{u}_i$, sous la contrainte ${}^t \mathbf{u}_i \mathbf{T} \mathbf{u}_i = 1$

Si on prend la métrique définie par \mathbf{T}^{-1} et comme forme quadratique \mathbf{B} , alors ce problème est analogue au problème de l'Analyse en Composante Principale (présentée au § 2.3.1) et la solution est donnée par les vecteurs propres \mathbf{u}_i de $\mathbf{T}^{-1} \mathbf{B}$ (qui vérifient $\mathbf{T}^{-1} \mathbf{B} \mathbf{u}_i = \lambda_i \mathbf{u}_i$).

Les axes factoriels discriminants sont donnés par les vecteurs :

$$\mathbf{v}_i = \mathbf{T} \mathbf{u}_i \quad (10-47)$$

(de norme 1 pour la métrique \mathbf{T}^{-1}) vérifiant :

$$\mathbf{B}\mathbf{T}^{-1}\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (10-48)$$

La valeur propre λ_i (avec $\lambda_i < \lambda_{i-1}$) représente le pouvoir factoriel de l'axe factoriel \mathbf{v}_i car :

$$\lambda_i = \frac{{}^t\mathbf{u}_i \mathbf{B} \mathbf{u}_i}{{}^t\mathbf{u}_i \mathbf{T} \mathbf{u}_i} \quad (10-49)$$

Si $\lambda_i = 1$ (i.e. ${}^t\mathbf{u}_i \mathbf{B} \mathbf{u}_i = {}^t\mathbf{u}_i \mathbf{T} \mathbf{u}_i$ et ${}^t\mathbf{u}_i \mathbf{W} \mathbf{u}_i = 0$), alors les classes se confondent avec leurs centres de gravité et la discrimination est parfaite. Si $\lambda_i = 0$ (i.e. ${}^t\mathbf{u}_i \mathbf{B} \mathbf{u}_i = 0$), alors les centres de gravité des classes se confondent et il n'est pas possible de définir un axe discriminant.

On se limite généralement aux r (r fixé) plus grandes valeurs propres car elles constituent une proportion importante de la trace de la matrice $\mathbf{T}^{-1}\mathbf{B}$:

$$\text{Tr}(\mathbf{T}^{-1}\mathbf{B}) = \sum_i \lambda_i \quad (10-50)$$

Le sous-espace vectoriel correspondant aux r premiers axes factoriels contient donc la majorité de l'information. Il y a au maximum q axes factoriels (où q est le nombre d'attributs). Il est possible de réduire la taille de l'espace discriminant par cette méthode, si les axes discriminants se confondent avec les axes des attributs ou s'ils sont contenus dans un sous-espace, car, dans ce cas, il y a des attributs dont le facteur de discrimination est nul. Généralement, on relance l'analyse factorielle sur l'espace réduit pour réajuster les axes.

Dans ce sous-espace, la moyenne des variances de classes est minimale et les moyennes de classes sont les plus dispersées possibles. L'individu \mathbf{x} a pour coordonnées dans ce sous-espace $\{\mathbf{T}\mathbf{u}_i \mathbf{x}\}$.

Une règle de décision simple consiste, par exemple, à affecter le point \mathbf{x} à la classe k la plus proche au sens de la métrique définie par \mathbf{W} , appelée également distance de Mahalanobis (cf. § 2.3.1.b), ce qui permet de tenir compte des dispersions des classes.

Cette méthode possède en outre une bonne stabilité devant l'échantillonnage (les attributs estimés étant la moyenne et l'écart-type).

Si certains attributs sont très corrélés, alors la matrice de variance totale \mathbf{T} n'est plus inversible. Il est donc nécessaire d'effectuer dans un premier temps une sélection des attributs, afin de supprimer les redondances.

Une méthode, proposée par Unser [UNSE89], permet d'utiliser ce principe dans le cadre de la classification non supervisée, par l'estimation des matrices de variances.

Application à la classification d'images d'énergie

La *méthode de Laws* consiste à construire des images d'énergie par filtrage linéaire et à classifier les images obtenues. Les images d'énergie de texture sont construites dans le domaine spatial par convolution avec des masques. Le principe est le suivant :

- 1) Filtrage de l'image originale par un ensemble de masques de convolution de petite taille, typiquement masques 5×5 à coefficients entiers ; ces masques sont séparables et, dans le cas 5×5 , obtenus à partir des masques 1×5 suivants :

$$\mathbf{L}_5 = (\quad 1 \quad 4 \quad 6 \quad 4 \quad 1)$$

$$\mathbf{E}_5 = (\quad -1 \quad -2 \quad 0 \quad 2 \quad 1)$$

$$\mathbf{S}_5 = (\quad -1 \quad 0 \quad 2 \quad 0 \quad -1)$$

$$\mathbf{W}_5 = (\quad -1 \quad 2 \quad 0 \quad -2 \quad 1)$$

$$\mathbf{R}_5 = (\quad 1 \quad -4 \quad 6 \quad -4 \quad 1)$$

Les masques se présentent sous la forme ${}^t\mathbf{E}_5 \mathbf{L}_5$, ${}^t\mathbf{R}_5 \mathbf{R}_5$, ${}^t\mathbf{E}_5 \mathbf{S}_5$, ${}^t\mathbf{L}_5 \mathbf{S}_5$, etc.

Par exemple le masque ${}^t\mathbf{E}_5 \mathbf{L}_5$ vaut :

$$\begin{array}{ccccc} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{array}$$

- 2) Application d'un filtre local non linéaire sur les images obtenues à la phase 1), par exemple moyenne sur une fenêtre assez grande (15×15 , 31×31) des valeurs absolues des valeurs des pixels. Ceci produit des images que Laws appelle des images d'énergie de texture.
- 3) Combinaison linéaire des images précédentes afin d'obtenir un nombre plus réduit d'images (3 ou 4).
- 4) Classification de ces images.

La *méthode de Ade* [ADE83a] est une utilisation directe de l'Analyse en Composante Principale. Elle consiste à prendre comme vecteur d'attributs d'un pixel, l'ensemble des niveaux de gris normalisés de ses q voisins. Les axes

principaux sont calculés sur une région d'apprentissage. A chacun des q vecteurs propres calculés est associé un masque de convolution. Les masques ainsi obtenus pour certaines textures sont très proches de ceux déterminés empiriquement par Laws, mais ont l'avantage d'être orthogonaux et d'être adaptés à la texture étudiée. L'image à segmenter est ensuite convoluée avec chacun des masques, conduisant à une série d'images représentant les composantes principales de la texture. Cette méthode a notamment été utilisée pour détecter les irrégularités dans les textures (après une phase d'apprentissage) [ADE83b].

2.2.2 Construction d'un sous-espace discriminant

Les méthodes présentées dans ce paragraphe intègrent à la fois la sélection des attributs et la construction d'une fonction de discrimination. Elles permettent de construire cette fonction dans le cas de grands espaces d'attributs et d'un nombre de classes important.

2.2.2.1 Discrimination non paramétrique

Cette méthode, qui ne suppose pas *a priori* l'adéquation des données à un modèle, a été adaptée au problème multiclassés par Celeux et Lechevallier [CELE82]. Elle génère un arbre de décision dans lequel les décisions prises à chaque niveau sont monodimensionnelles. Pour plus de clarté, on exposera ici la méthode monodimensionnelle employée ; dans ce cas on rejoint le paragraphe 1.

a) Principe général (un attribut, deux classes)

On supposera les probabilités *a priori* et le coût de mauvaise classification de chaque classe respectivement tous égaux entre eux. Soit $F_1(x)$ (resp. $F_2(x)$) la fonction de distribution de C_1 (resp. C_2).

Le principe de la méthode est le suivant : pour classifier un individu x , on le compare à un seuil ou valeur de coupure c : si $x \geq c$ alors x est affecté à la classe supérieure sinon x est affecté à la classe inférieure. On choisit c de façon à minimiser le risque de Bayes de mauvaise classification :

$$R(c) = 1 - F_1(c) + F_2(c) \quad (10-51)$$

Or, minimiser $R(c)$ revient à maximiser :

$$D(c) = \max_c |F_1(c) - F_2(c)| \quad (10-52)$$

qui est la distance de Kolmogorov-Smirnov. Si $F_1(c) > F_2(c)$, alors la région discriminante pour C_1 est $-\infty, c$, et $c, +\infty$ sinon.

En pratique, les fonctions de distribution ne sont pas connues. On utilise la distribution empirique suivante :

$$\hat{F}_j(x) = \begin{cases} 0 & \text{si } x < x_j(1) \\ \frac{i}{n_j} & \text{si } x_j(i) \leq x < x_j(i+1) \\ 1 & \text{si } x \geq x_j(n_j) \end{cases} \quad (10-53)$$

où $x_j(i)$ est le $i^{\text{ème}}$ élément de l'ensemble d'apprentissage appartenant à la classe C_j , classée par ordre croissant et $n_j = \text{card } C_j$ et on calcule :

$$\hat{D}(c) = \max_c |\hat{F}_1(c) - \hat{F}_2(c)| \quad (10-54)$$

Extensions

- *extensions aux multiattributs*

La coupure est faite avec l'attribut qui, parmi l'ensemble des attributs, maximise la distance de Kolmogorov-Smirnov entre les deux populations (les attributs des deux populations se distribuent alors en deux segments). On réitère la procédure jusqu'à ce que la population d'un segment soit inférieure à un seuil.

- *extensions aux multiclassés*

La distance de Kolmogorov-Smirnov n'est pas adaptée aux multiclassés. On s'y rapporte donc en considérant, non plus deux classes, mais deux groupes de classes complémentaires. Soit π un ensemble de r classes de C . La fonction de distribution empirique de π est estimée par :

$$\hat{F}_\pi(x) = \sum_{C_i \in \pi} \hat{F}_i(x) \quad (10-55)$$

La meilleure valeur de coupure est trouvée avec le critère suivant :

$$\hat{D}(\hat{c}) = \sup_{\pi \in \Pi} \sup_x |\hat{F}_\pi(x) - \hat{F}_{\bar{\pi}}(x)| \quad (10-56)$$

où Π est l'ensemble des partitions en deux classes de C , et $\bar{\pi}$ est le complémentaire de π dans C .

La procédure est réitérée récursivement sur chaque groupe de classes jusqu'à aboutir à un segment pur (*i.e.* ne contenant que des éléments d'une même classe) ou jusqu'à ce que la population du segment soit trop petite pour que celui-ci soit de nouveau partitionné. Dans ce cas, le segment est affecté à la classe qui y est majoritaire.

b) Règles de décisions

On construit ainsi un arbre de décision (*cf.* figure 10.11) qui peut se réécrire sous forme de règles de production du type :

SI $p_1(x) > c_1$ OU $p_2(x) > c_2$
 ALORS classe 1
 SI $p_1(x) \leq c_1$ ET $p_2(x) \leq c_2$
 ALORS classe 2

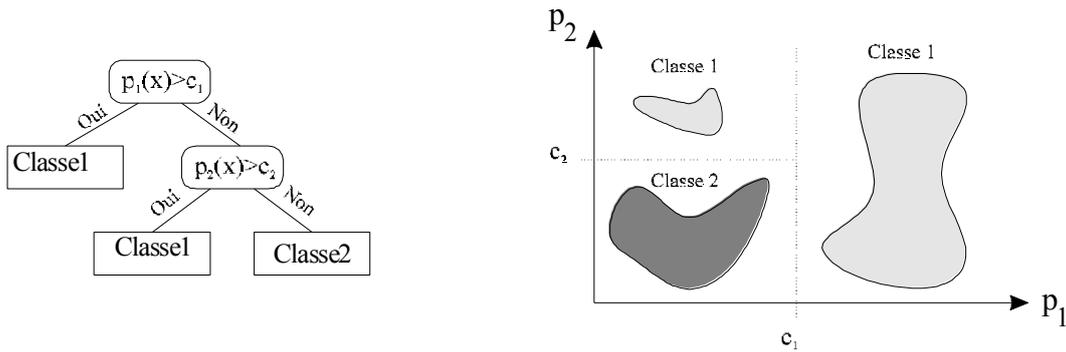


Figure 10.11 : Arbre de décision et classification correspondante

L'avantage de cette méthode est la facilité de sa mise en œuvre et sa capacité à traiter les espaces de grande dimension (*i.e.* avec un grand nombre d'attributs). Par contre, l'utilisation de seuils à chaque niveau de l'arbre transforme les zones adjacentes aux coupures en zones instables où de très faibles variations d'attribut peuvent entraîner des décisions radicalement différentes. Il convient également de se méfier des découpages trop fins qui pourraient conduire à une sur-adaptation de la méthode à l'ensemble d'apprentissage.

2.2.2.2 Approche par la méthode des paires

La méthode des paires est une approche générique qui permet de manipuler un grand nombre de classes et d'attributs avec une faible complexité. Elle possède des qualités structurelles très intéressantes qui permettent notamment d'adapter la discrimination au contexte ou de faire des mises à jour sans invalider la méthode.

Initialement présentée dans [LEVY92], cette méthode originale est utilisée dans Arthur, un logiciel d'analyse de texture [LECH91].

La méthode se déroule en deux étapes :

- *Etape de discrimination*

Cette étape a pour but de sélectionner les attributs les plus discriminants. Le problème est ramené à un problème biclasse. Pour chaque paire de classes (C_i, C_j) , un espace discriminant local est choisi dans l'espace des attributs par analyse de l'ensemble d'apprentissage. On obtient ainsi deux fonctions d'appartenance : $D_{ij}(x)$ (resp. $D_{ji}(x)$) qui indique la probabilité d'appartenance à la classe C_i (resp. C_j) de l'élément x lorsqu'elle est opposée à la classe C_j (resp. C_i).

L'idée est donc de ne plus essayer lors de la phase de discrimination de construire un espace discriminant global pour l'ensemble des classes. Pour chaque couple de classes, on cherche un « expert » hyper-spécialisé, ce qui est relativement aisé car les méthodes de discrimination biclasses sont abondantes et conduisent généralement au choix d'espaces de très faibles dimensions (un ou deux). Il est malgré tout préférable d'utiliser une méthode ayant pour sortie

un degré d'appartenance à l'une ou l'autre classe (de type logique floue) pour des raisons qui apparaîtront clairement par la suite.

La méthode des paires peut déboucher sur un nombre très important d'attributs ($q(q+1)/2$ avec un attribut par classe) : en fait, la redondance intrinsèque d'informations permet de réduire ce nombre. Par exemple les assertions : (p_1 discrimine C_1 et C_2) et (p_1 ne discrimine pas C_2 et C_3) peuvent conduire à (p_1 discrimine C_1 et C_3).

La méthode des paires sélectionne généralement moins d'attributs que les méthodes conduisant à des arbres de décision.

Les attributs apportant la meilleure discrimination (au sens du critère pour la méthode biclasse choisie) pour chaque couple de classes sont sélectionnés. La reconstruction de l'espace de discrimination global se fait par fusion des discriminations locales lors de l'étape suivante.

- *Etape de décision*

Pour classer un élément x , on calcule l'ensemble des fonctions d'appartenance pour chaque paire de classes. Les résultats sont regroupés dans une matrice de décision.

$$M(x) = (D_{ij}(x)) \text{ avec } D_{ii}(x) = 1 \quad (10-57)$$

On appelle vote en faveur de la classe C_i contre la classe C_j pour un échantillon x , la valeur de la fonction d'appartenance $D_{ij}(x)$.

La $i^{\text{ème}}$ ligne de la matrice contient la collection des votes en faveur de la classe C_i . Il est important de remarquer que chaque élément de la matrice se réfère à un ensemble d'attributs différent dans l'espace des attributs. Pour chaque paire, le sous-espace le plus favorable à la discrimination a été sélectionné.

Une fonction pseudo-floue de décision est ensuite appliquée à cette matrice :

$$F(x) = \arg \max_i \prod_j D_{ij}(x) \quad (10-58)$$

Cette fonction de vote sélectionne la classe qui est globalement la meilleure, c'est-à-dire la classe k qui maximise le produit des votes en sa faveur. Pour bien comprendre le fonctionnement de ce vote, voici quelques exemples.

Exemple 1 :

$$M(x_1) = \begin{bmatrix} 1.00 & 0.70 & 0.80 \\ 0.30 & 1.00 & 0.05 \\ 0.20 & 0.95 & 1.00 \end{bmatrix}$$

alors $F(x_1) = C_1$.

On peut remarquer un fait étonnant : l'opposition entre les classes C_2 et C_3 est utilisée alors que c'est la classe C_1 qui est décidée. Cela est pourtant nécessaire car le résultat de ce vote peut influencer le vote final. Ce phénomène est particulièrement net dans le cas des votes ambigus.

Exemple 2 :

$$\mathbf{M}(x_2) = \begin{bmatrix} 1.00 & 0.51 & 0.51 \\ 0.49 & 1.00 & 0.99 \\ 0.49 & 0.95 & 1.00 \end{bmatrix}$$

alors $F(x_2) = C_2$.

Bien que la classe C_2 ait perdu contre la classe C_1 , c'est la classe C_3 qui fait la décision entre C_1 et C_2 . Il peut donc sembler étrange, sachant que la classe finale de x sera C_1 , de se demander si cet individu appartient plutôt à C_2 ou à C_3 . Un autre point à noter est que plus le nombre de classes est important, plus la classe choisie recueille de votes en sa faveur, ce qui fiabilise le vote.

D'autres stratégies de vote peuvent également être appliquées, comme le minimax qui minimise le risque de mauvaise classification défini par :

$$F(x) = \arg \max_i \left(\min_j D_{ij}(x) \right)$$

Dans les exemples précédents, on obtient alors $F(x_1) = C_1$ et $F(x_2) = C_1$.

S'il est possible d'utiliser des fonctions d'appartenance binaires (qui ont pour effet de créer des matrices de décision binaires), celles-ci ont l'inconvénient de produire des effets de seuil. Ainsi, dans le cas de l'exemple 2, la classe 1 aurait été choisie.

Commentaire

La méthode des paires est simple à mettre en œuvre et peut être utilisée avec les méthodes de discrimination biclasse déjà existantes. Elle n'est pas optimale, mais donne des résultats tout à fait comparables à la méthode de discrimination non paramétrique. En général, le taux de reconnaissance se répartit bien entre les classes. Il est très rare de voir des classes très bien reconnues, alors que d'autres le sont très mal, l'erreur ayant tendance à se répartir à travers toute la matrice de vote.

Cette méthode a, de plus, des propriétés structurelles intéressantes liées à la séparation entre la discrimination et la décision ainsi qu'à la manipulation de paires de classes. Elle permet notamment de :

- changer la probabilité *a priori* et le coût de mauvaise classification de chaque classe en modifiant les poids dans la fonction d'appartenance,
- travailler dans un sous-espace de classes en n'utilisant que les paires de classes intervenant dans ce sous-espace,

- ajouter une nouvelle classe sans invalider le schéma de discrimination. Il suffit d'y ajouter toutes les nouvelles paires engendrées par cette nouvelle classe,
- tester si un nouvel attribut n'améliorerait pas de manière significative la discrimination. Si c'est le cas, mettre à jour l'ensemble des classes.

L'inconvénient de cette méthode est l'obligation de calculer tous les attributs discriminants pour aboutir à la décision finale, alors que les méthodes de type arbre de décision n'obligent à calculer que ceux qui sont rencontrés lors du parcours descendant de l'arbre.

Des résultats de segmentation d'images texturées par cette méthode sont donnés dans le chapitre XIII.

2.3 Méthodes de sélection des attributs

Les méthodes de sélection d'attributs jouent un rôle de filtre face au bruit apporté par les attributs non représentatifs. Cette étape est donc essentielle pour le bon fonctionnement des méthodes de discrimination citées plus haut.

2.3.1 Définition des critères de sélection

2.3.1.1 Non-supervisé

La méthode de sélection la plus utilisée (et la plus connue) est l'Analyse en Composante Principale (ACP ou transformée de Karhunen-Loeve) [DIDA82] [LEBA82]. Son principe est simple : on cherche un sous-espace de dimension $r < N_c$ dans lequel les données sont les plus étalées possible, c'est-à-dire les axes de projection sur lesquels la variance des données est maximale. Les axes principaux sont déterminés par les vecteurs propres de la matrice de covariance \mathbf{T} .

L'interprétation des vecteurs propres \mathbf{v}_i (ou axes factoriels) et des valeurs propres λ_i associées est semblable à l'analyse factorielle discriminante présentée précédemment (§ 2.2.1). λ_i est la variance portée par l'axe factoriel \mathbf{v}_i . On choisira donc les axes factoriels portant les plus grandes variances. Si ces axes factoriels sont proches des axes initiaux, on choisira les attributs correspondants.

Il faut toutefois noter que cette méthode doit être appliquée avec précaution. Il est important de noter les faits suivants :

- Si deux attributs sont très corrélés, l'axe factoriel principal pour ces deux variables est la bissectrice. L'ACP préconise de garder ces deux attributs. Il convient donc de choisir les attributs avec discernement et éviter les attributs trop corrélés.
- Si deux attributs ont des différences d'échelles importantes, c'est l'attribut qui a la plus grande échelle qui portera toujours l'axe factoriel principal car c'est sur cet axe que la variance sera la plus grande (i.e. \mathbf{T} est mal conditionnée). Pour cette raison, on utilise souvent la matrice de corrélation (chaque attribut est normalisé par sa variance totale et recentré à 0).

- Si les attributs ne sont pas choisis avec soin vis-à-vis des classes à discriminer, certains attributs pourront faire ressortir des caractéristiques inattendues provoquant un mélange interclasses tout en dispersant suffisamment les données pour être sélectionnés ce qui peut être la source d'un bruit important.

2.3.1.2 Supervisé

Pour décider si un attribut est discriminant vis-à-vis du problème, on pourra employer l'un ou l'autre des critères présentés dans ce paragraphe [CELE82] [DIDA82].

Les deux premiers critères sont basés sur l'analyse factorielle : le critère ne privilégiera donc que les attributs performants au sens de l'analyse factorielle (bonne séparation des classes dans l'espace de représentation des ellipsoïdes d'inertie). On reprendra les notations adoptées dans le paragraphe sur l'analyse factorielle (§ 2.2.1).

- *critère de la trace*

Pour estimer le pouvoir discriminant d'un ensemble de s attributs (parmi les q attributs), on utilise les matrices de variances définies au § 2.2.1, mais restreintes aux s attributs considérés. On note \mathbf{T}_s (resp. \mathbf{B}_s et \mathbf{W}_s) la matrice (de dimension $s \times s$) de variance totale (resp. interclasse et intraclasse) extraite de \mathbf{T} (resp. \mathbf{B} et \mathbf{W}).

On calcule :

$$Tr(\mathbf{T}_s^{-1} \mathbf{B}_s) = \sum_{i=1}^s \lambda_i \quad (10-59)$$

où les λ_i sont les valeurs propres de $\mathbf{T}_s^{-1} \mathbf{B}_s$. Ce critère est à maximiser.

- *critère de Wilks*

Le pouvoir discriminant d'un ensemble de s attributs est estimé par :

$$\frac{\det \mathbf{W}_s}{\det \mathbf{T}_s} = \prod_{i=1}^s \lambda_i \quad (10-60)$$

Ce critère est à minimiser.

Les critères suivants ne sont valables que pour deux classes (de centres d'inertie g_1 , g_2 et de densités de probabilité $f_1(x)$ et $f_2(x)$). Il est cependant possible de les appliquer dans le cas multiclasse en considérant deux groupes de classes.

- *distance de Mahalanobis :*

$$t(g_1 - g_2) \mathbf{W}^{-1} (g_1 - g_2) \quad (10-61)$$

Cette distance est une mesure du multiple d'écart-type séparant les centres d'inertie des deux classes, en supposant les distributions normales. Donc, plus cette distance est importante, plus la probabilité d'erreur est faible, car un

individu devra s'écarter beaucoup de sa moyenne, avant d'être confondu avec l'autre classe.

- *distance de Bhattacharya* :

$$-\ln \int_{\mathbb{R}^n} \sqrt{f_1(x) \cdot f_2(x)} dx \quad (10-62)$$

Cette distance est une mesure du recouvrement des fonctions de densités de probabilité. Plus ces fonctions sont différentes, plus l'intégrale du produit est faible et plus l'opposé de son logarithme est important.

Bien que ces critères s'appliquent au cas multidimensionnel, la sélection des attributs se fait généralement axe par axe (finalement en une dimension). Pour le dernier critère, la difficulté d'estimer les densités de probabilité conduit à limiter sérieusement la dimension de l'espace de représentation.

Il faut remarquer que le critère de sélection utilisé dépend fortement de la méthode de discrimination employée. Il s'agit alors de choisir (ou de construire) le critère qui sélectionnera les attributs et qui, conjugué avec la méthode, fournira les meilleurs résultats.

2.3.2 Procédure de sélection des attributs

La procédure optimale de sélection des attributs consisterait à tester tous les s -uplets d'attributs, et à sélectionner le meilleur, puis à recommencer avec $s+1$ attributs jusqu'à ce que la qualité de l'ensemble d'attributs décroisse.

En pratique, la complexité algorithmique est si prohibitive qu'on applique la procédure sous-optimale de sélection des attributs pas-à-pas : on sélectionne le meilleur attribut, puis on sélectionne l'attribut qui, couplé avec le meilleur attribut, fournit le meilleur couple d'attributs, puis on sélectionne l'attribut qui, couplé avec le meilleur couple d'attributs, fournit le meilleur triplet d'attributs, *etc.* Des améliorations de cette méthode ainsi que d'autres méthodes de sélection d'attributs sont proposées dans [CELE82] [DIDA82].

Le nombre d'attributs sélectionnés doit être comparable au nombre de classes. Il serait illusoire de penser qu'un grand nombre d'attributs apporte une meilleure discrimination, car l'information qu'ajoutent les attributs excédentaires est négligeable devant le bruit qu'ils apportent.

2.4 Procédure de validation

Les procédures de validation sont malheureusement trop souvent oubliées alors qu'elles doivent constituer la dernière épreuve lors de l'application d'une méthode de discrimination [CELE89]. Elles consistent à tester la fonction de classification avec des individus n'ayant pas participé à l'élaboration de la fonction, afin de vérifier la cohérence et la stabilité du résultat obtenu. Lorsqu'elles sont appliquées avec trop d'enthousiasme, les méthodes de discrimination peuvent conduire à une reconnaissance parfaite de l'ensemble d'apprentissage et donner des résultats médiocres pour la classification de nouveaux individus.

Si l'ensemble d'apprentissage est suffisamment important, on pourra réserver une partie de celui-ci pour la validation des données via l'ensemble test.

Une autre procédure (appelée validation croisée) consiste à diviser aléatoirement l'ensemble d'apprentissage en t parties égales puis à faire t apprentissages sur $t-1$ parties en gardant la dernière pour le test de validation. Cette procédure permet de tester à la fois la cohérence et la stabilité du résultat. On obtient ainsi une estimation réaliste du taux de bon classement de la classe C_i

avec $Q_i = \frac{1}{t} \sum_{j=1}^t Q_i(j)$, où $Q_i(j)$ est le pourcentage de bon classement de la classe

C_i lorsqu'on utilise la $j^{\text{ème}}$ partie comme ensemble-test.

On peut également tirer avec remise aléatoire dans l'ensemble d'apprentissage afin de constituer un grand nombre d'ensembles sur lesquels l'apprentissage va se faire. On peut ainsi estimer précisément la moyenne et l'écart-type de la fonction de probabilité de bon classement.

Une description précise de ces procédures peut être obtenue dans [INRI90].

3 Conclusion

Ce chapitre fournit un aperçu des potentialités des méthodes de classification, qui exploitent une grande diversité de paramètres. Ces derniers peuvent être calculés sur des entités allant du simple point, au voisinage, à la composante connexe, à la classe et même à l'image globale.

Ces méthodes travaillant sur des données de bas niveau, il est important d'établir un lien avec des concepts de haut-niveau, soit par coopération avec l'approche frontière, soit en associant des données issues de connaissances *a priori* sur la scène (par exemple la forme).

Approche région : méthodes markoviennes

1 Régularisation

1.1 Introduction

Nous allons ici reprendre des exemples de régularisation de type markovien, les définitions concernant les champs de Markov sont rappelées en annexe B.

L'image est un ensemble de sites, ces derniers sont les sommets d'une maille carrée de taille $N \times N$ notés S . A chaque site s est associé un niveau de gris a_s et une étiquette λ_s . Nous supposons que l'image niveaux de gris observée \mathbf{a} , avec $\mathbf{a} = \{a_s, s \in S\}$, est décrite par un champ de Markov \mathbf{A} . L'image segmentée donnée par une image d'étiquettes $\boldsymbol{\lambda}$, avec $\boldsymbol{\lambda} = \{\lambda_s, s \in S\}$, est aussi décrite comme une réalisation d'un champ markovien $\boldsymbol{\Lambda}$. L'ensemble des valeurs prises par λ_s dépend du nombre total de classes présentes dans l'image ; soit L ce nombre, alors $\lambda_s \in \{1, 2, \dots, L\}, \forall s \in S$.

La segmentation consiste à estimer $\boldsymbol{\lambda}$ connaissant l'image observée \mathbf{a} . Une approche, dans ce cas, commence dans un premier temps par définir une probabilité conditionnelle $P(\mathbf{A} = \mathbf{a} | \boldsymbol{\Lambda} = \boldsymbol{\lambda})$. Cette dernière correspond, par exemple, à une modélisation des textures, ou, en règle générale, à une modélisation du comportement des observations à l'intérieur d'une région homogène (ceci est encore appelé « modèle de déformation »). Dans un deuxième temps, on recherche la configuration $\boldsymbol{\lambda}$ qui maximise $P(\boldsymbol{\Lambda} = \boldsymbol{\lambda} | \mathbf{A} = \mathbf{a})$. En utilisant le théorème de Bayes, il vient :

$$P(\boldsymbol{\lambda} | \mathbf{A} = \mathbf{a}) = \frac{P(\mathbf{A} = \mathbf{a}, \boldsymbol{\Lambda} = \boldsymbol{\lambda})}{P(\mathbf{A} = \mathbf{a})} = \frac{P(\mathbf{A} = \mathbf{a} | \boldsymbol{\Lambda} = \boldsymbol{\lambda}) P(\boldsymbol{\Lambda} = \boldsymbol{\lambda})}{P(\mathbf{A} = \mathbf{a})} \quad (11-1)$$

Pour simplifier les notations et quand il n'y a pas d'ambiguïté, nous noterons : $P(\boldsymbol{\lambda})$ pour $P(\boldsymbol{\Lambda} = \boldsymbol{\lambda})$, $P(\mathbf{a})$ pour $P(\mathbf{A} = \mathbf{a})$, etc.

Dans l'expression (11-1), $P(\mathbf{a})$ est constant et $P(\mathbf{a} | \boldsymbol{\lambda})$ est connu puisque un modèle de déformation a été choisi. $P(\boldsymbol{\lambda})$ représente la loi de probabilité *a priori* du champ $\boldsymbol{\Lambda}$.

Comme, par hypothèse, les champs sont markoviens, on définira de manière équivalente $P(\boldsymbol{\lambda} | \mathbf{a})$ par l'énergie $U(\boldsymbol{\lambda} | \mathbf{a})$:

$$P(\mathbf{a}) = \frac{\exp(-U(\mathbf{a}))}{Z} \quad (11-2)$$

1.2 Un algorithme supervisé de segmentation d'images texturées

Cette approche implique que l'on dispose d'échantillons de chacune des textures présentes [GEMA87a] [GEMA87b].

1.2.1 Modèle de texture

Choisir un modèle de texture revient à définir la probabilité conditionnelle $P(\mathbf{a}|\lambda)$ d'avoir la réalisation \mathbf{a} sachant qu'on a la texture ℓ en tout site (λ est une réalisation où $\lambda_s = \ell \quad \forall s \in \mathcal{S}$). Comme \mathbf{A} est un champ de Markov, on peut écrire :

$$P(\mathbf{a}) = \frac{1}{Z^{(\ell)}} \exp(-U^{(\ell)}(\mathbf{a})) \quad (11-3)$$

où $Z^{(\ell)}$ est la constante de normalisation :

$$Z^{(\ell)} = \sum_{\mathbf{a}} \exp(-U^{(\ell)}(\mathbf{a})) \quad (11-4)$$

Nous utilisons, pour ce modèle, uniquement des cliques d'ordre deux appartenant au système de voisinage défini par les 48 plus proches voisins d'un site (*cf.* figure 11.1). Dans ce voisinage, on définit I types de cliques. Le couple de voisins (s,t), liés par la clique de type i est noté $\langle s,t \rangle_i$. Dans la figure 11.1, sont présentés 6 types de cliques.

Le modèle de texture proposé est décrit par :

$$U^{(\ell)}(\mathbf{a}) = - \sum_{i=1}^I \sum_{\langle s,t \rangle_i} \theta_i^{(\ell)} \Phi(a_s - a_t) \quad (11-5)$$

où la fonction Φ est définie par la formule□:

$$\Phi(\Delta) = \left(1 + \left(\frac{\Delta}{\delta} \right)^2 \right)^{-1} \quad (11-6)$$

dans laquelle $\delta > 0$ est un paramètre de normalisation.

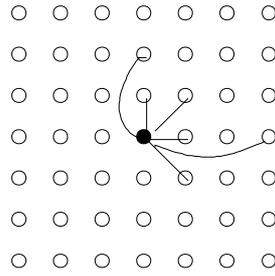


Figure 11.1 : Système de voisinages et quelques-unes des cliques

Les paramètres $\theta_i^{(\ell)}$, $i = 1, 2, \dots, I$, $\ell = 1, 2, \dots, L$ sont estimés à partir des échantillons de textures par la méthode du maximum de pseudo-vraisemblance [GEMA87a].

1.2.2 Loi jointe

Après avoir défini le modèle de textures, nous proposons la loi jointe suivante portant sur Λ et \mathbf{A} [GEMA87b]:

$$P(\mathbf{A} = \mathbf{a}, \Lambda = \lambda) = \frac{\exp\{-U_1(\mathbf{a}, \lambda) - U_2(\lambda)\}}{Z} \quad (11-7)$$

où U_2 représente le modèle *a priori* sur les étiquettes et U_1 représente le terme d'interaction entre les étiquettes et les niveaux de gris.

Pour U_2 , on choisit un terme de régularisation de type *Ising* :

$$U_2(\lambda) = -\beta \sum_{[s,t]} \mathbf{1}_{\lambda_s = \lambda_t} + \sum_{s \in S} w(\lambda_s) \quad (11-8)$$

où :

- $\beta > 0$ quantifie l'importance relative du terme de régularisation,
- $\mathbf{1}_{\lambda_s = \lambda_t}$ est une fonction indicatrice,
- $[s,t]$ est une clique d'ordre 2 horizontale ou verticale,
- $w(\cdot)$ est un seuil permettant de sélectionner l'étiquette la plus appropriée, son calcul est expliqué en fin de paragraphe.

Pour l'énergie U_1 qui décrit les interactions entre les pixels et les étiquettes, nous proposons la forme suivante [GEMA87b] :

$$U_1(\mathbf{a}, \lambda) = \sum_{s \in S} V(\mathbf{a}, \lambda_s, s) \quad (11-9)$$

où :

$$V(\mathbf{a}, \ell, s) = \frac{1}{n_F} \sum_{t \in F_s} H(\mathbf{x}, \ell, t) \quad (11-10)$$

$V(\mathbf{a}, \ell, s)$ est une moyenne sur la fenêtre F_s , centrée en s , des contributions des couples de pixels à la probabilité conditionnelle dans le modèle de texture.

Notons $\tau_1, \tau_2, \dots, \tau_I$ les vecteurs associés aux cliques d'ordre 2 contenues dans le système de voisinages des 48 plus proches voisins (*cf.* figure 11.1). s et $s+\tau_i$ sont donc deux sites voisins liés par la $i^{\text{ème}}$ clique. Nous avons pour $H(\mathbf{a}, \ell, t)$ l'expression :

$$H(\mathbf{a}, \ell, t) = -\sum_{i=1}^I \theta_i^{(\ell)} \{ \Phi(\mathbf{a}_t - \mathbf{a}_{t+\tau_i}) + \Phi(\mathbf{a}_t - \mathbf{a}_{t-\tau_i}) \} \quad (11-11)$$

La constante n_F de la formule (11-10) est calculée de manière à ce que la somme des contributions de tous les voisinages F_s se réduise à $U^{(\ell)}$ donné par :

$$U^{(\ell)}(\mathbf{a}) = \sum_{s \in S} H(\mathbf{a}, \ell, s) \quad (11-12)$$

ce qui revient à s'assurer que chaque clique apparaît exactement une fois. Par exemple, il y a $n_F = 50$ cliques horizontales pour une fenêtre F_s de taille 5×5 . En effet, il y a 25 sites et 2 cliques $\langle \mathbf{a}_t, \mathbf{a}_{t+\tau_i} \rangle$ et $\langle \mathbf{a}_t, \mathbf{a}_{t-\tau_i} \rangle$ par site.

Grâce à la relation (11-12), ce modèle est cohérent avec le modèle de texture défini en (11-4) pour une image d'étiquette constante $\lambda_s = \ell$, $\forall s \in S$. L'idée est que chaque étiquette est influencée localement par le niveau de gris des pixels voisins.

La fonction $w(\cdot)$ apparaissant dans l'expression de U_2 est calculée en observant le logarithme du rapport des vraisemblances des étiquettes k et j :

$$\begin{aligned} & \ln \left\{ \frac{\Pi(\lambda_r = k | \lambda_s, s \neq r; \mathbf{x}_s, s \in S)}{\Pi(\lambda_r = j | \lambda_s, s \neq r; \mathbf{x}_s, s \in S)} \right\} \\ &= (H(\mathbf{a}, j, r) - H(\mathbf{a}, k, r)) + \left(\beta \sum_{t \in \langle i, r \rangle} (\mathbf{1}_{\lambda_{t=k}} - \mathbf{1}_{\lambda_{t=j}}) + w(j) - w(k) \right) \end{aligned} \quad (11-13)$$

Le premier terme est un terme d'interaction avec les données, et le second est un terme spatial sur les étiquettes.

L'efficacité de ce modèle dépend de la qualité de la discrimination obtenue entre les étiquettes k et j . Celle-ci peut être contrôlée en traçant les histogrammes de ces valeurs pour des échantillons de textures k et j . Une séparation nette entre les deux histogrammes ainsi obtenus signifiera une bonne performance de la classification. Les $w(\cdot)$ sont choisis sur les échantillons comme étant les seuils de séparation des histogrammes.

1.2.3 Estimateur du MAP

Le problème est de maximiser :

$$P(\mathbf{a}) = \frac{1}{Z} \exp(-U_1(\mathbf{a}, \lambda) - U_2(\lambda)) \quad (11-14)$$

Dans ce cas particulier, étant donnée une observation $\mathbf{A} = \mathbf{a}$, nous recherchons une réalisation λ qui minimise $U_1(\mathbf{a}, \lambda) + U_2(\lambda)$.

Pour ce faire, nous avons utilisé un recuit simulé (*cf.* annexe B). Un point de départ convenable, et donc permettant de limiter fortement le nombre d'itérations, est obtenu en annulant le terme d'Ising dans l'énergie U_2 ($\beta \square = 0$). Étant donné que ce terme est le seul contenant des interactions entre des étiquettes voisines, l'estimateur du MAP de λ , avec $\beta \square = 0$, est déterminé localement en chaque site \square de \mathcal{S} et donc de manière très rapide.

1.3 Segmentation et détection de contours sur des images texturées

En général, les statistiques d'ordre un et deux sont suffisantes pour classifier les micro-textures, mais le problème est plus ardu dans le cas de macro-textures. Pour cet algorithme [AZEN92a], nous avons utilisé un ensemble de caractéristiques de textures classiques définies en particulier dans [HARA73], et des statistiques d'ordre un et deux : la moyenne, la variance et des covariances, pour caractériser les micro-textures. Nous avons ensuite construit de nouvelles caractéristiques basées sur la détection de motifs pour les macro-textures (voir figure 11.2 pour un exemple de motif).

Chacun des attributs classiques est supposé gaussien, et les paramètres du modèle sont estimés sur des échantillons.

1.3.1 Modèle de contours

Nous nous intéressons ici à un problème plus simple que celui d'une détection complète de frontières de zones texturées, avec fermeture des contours. En effet, étant donné que nous obtiendrons finalement une segmentation de l'image, nous en déduisons immédiatement les frontières finales. Cette première étape de détection de contours est destinée à extraire des informations précises et relativement sûres sur la position des éléments de contours. Cependant, puisque nous avons à faire *a priori* à des images fortement texturées, le problème n'est pas classique.

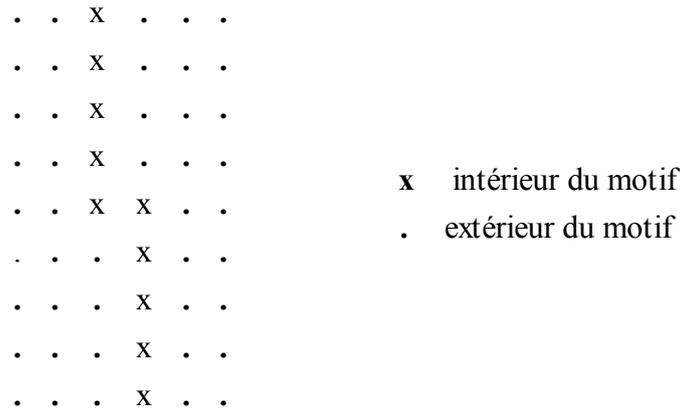


Figure 11.2 : Un exemple de motif utilisé pour les macro-textures

Nous avons utilisé une grille duale pour placer les éléments de contours, voir figure 11.3.

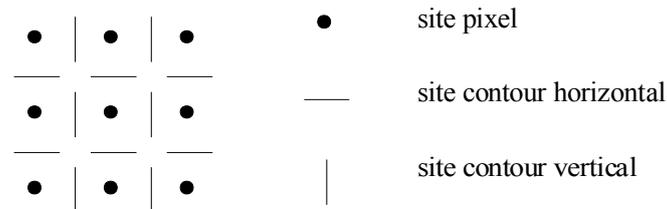


Figure 11.3 : Grilles duales utilisées pour les éléments de contours.

Deux fenêtres sont associées à chaque site contour (*cf.* figure 11.4.)

La taille de ces fenêtres peut dépendre de la présence de macro-textures. L'algorithme permet d'utiliser deux tailles de fenêtres, l'une pour les micro-textures, et l'autre pour les macro-textures. Le choix de la taille de cette fenêtre est fait de la manière suivante :

- tout d'abord, déterminer si un motif de macro-textures est présent dans la fenêtre de grande taille,
- si c'est le cas, il faut alors utiliser cette résolution, sinon, on utilise des fenêtres de petite taille correspondant à des micro-textures.

Notons que, dans chacun des cas, les deux fenêtres associées à un élément de contour sont de même taille. Une fois que les attributs ont été calculés, un test d'hypothèse gaussien classique permet de sélectionner les contours qui sont notés $(b_{\langle s,r \rangle})$. s et r sont deux plus proches voisins et la clique $\langle s,r \rangle$ caractérise le type de contours concerné.

Les éléments de contours sont fixés et ne seront plus modifiés dans la suite de l'algorithme.

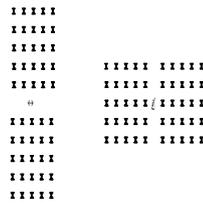


Figure 11.4 : Position des fenêtres utilisées pour la détection des éléments de contours dans les deux cas : contours verticaux et horizontaux.

1.3.2 Modèle de segmentation

Pour l'algorithme de segmentation, l'idée de base est de choisir en chaque site l'étiquette la mieux adaptée au vecteur d'attributs de texture observé. Il nous faut donc définir autour de chaque site une fenêtre dans laquelle seront calculés les attributs. En général, ce type de méthode induit une difficulté importante : à la frontière entre deux textures, une fenêtre contient un mélange des deux textures, si bien que les attributs de texture deviennent non significatifs.

Les éléments de contours peuvent être utilisés afin de pallier cet inconvénient majeur. En effet, ils permettent de déformer la fenêtre glissante à l'approche d'une frontière. D'autre part, il est aussi possible d'utiliser l'information de contours afin de stopper l'effet régularisateur de l'énergie au travers d'une frontière.

Notations :

Nous reprenons une partie des notations utilisées précédemment en les complétant :

- $\mathbf{a} = (a_s)$: niveaux de gris observés, s est le site pixel.
- M est le nombre total d'échantillons de textures (ou d'étiquettes), et l'indice k représente le $k^{\text{ème}}$ échantillon (ou étiquette).
- F_s : fenêtre d'observation centrée en s . La taille de F_s dépend de la présence de macro-textures.
- $\Lambda = (\lambda_s)$: configuration d'étiquettes qui définit la segmentation.
- V_s : voisinage de s (les huit plus proches voisins en général).
- $\mathbf{1}$ est la fonction indicatrice : $\mathbf{1}_{\lambda_s = \lambda_r} = 1$ si $\lambda_s = \lambda_r$ et 0 sinon.
- α_s = vecteur attribut calculé sur F_s .
- $\alpha_s^{(m)}$ est le $m^{\text{ème}}$ attribut. Cet attribut est calculé dans la fenêtre F_s qui sera tronquée suivant le nombre d'éléments de contours. Le calcul de $\alpha_s^{(m)}$ varie suivant deux critères :

Le premier est la taille de F_s :

S'il n'y a pas de motif de macro-texture autour de s , alors F_s est de taille correspondant à la résolution des micro-textures (la plus petite taille est de l'ordre d'un carré de 5 à 7 pixels de côté). Sinon, F_s est de la taille correspondant à la résolution des macro-textures (de l'ordre d'un carré de 15 à 25 pixels de côté).

Le second critère est la présence d'éléments de contours :

S'il y a suffisamment d'éléments de contours dans F_s , alors nous calculons la droite la mieux adaptée (au sens des moindres carrés) aux contours observés, et nous coupons la fenêtre par cette droite, sinon, les attributs sont calculés sur la fenêtre entière.

- $d(k, \alpha)$ est la distance entre un vecteur attribut α et le vecteur attribut de l'étiquette k : soient $\mu_k^{(m)}$ et $\sigma_k^{(m)}$ la moyenne et l'écart type du $m^{\text{ème}}$ attribut de l'étiquette k , alors :

$$d(k, \alpha) = \sup_m \left(\frac{\alpha^{(m)} - \mu_k^{(m)}}{\sigma_k^{(m)}} \right)^2 \quad (11-15)$$

Fonction d'énergie :

$$U_L() = \sum_{\text{sites}} \sum_{k=1}^K \mathbf{1}_{\lambda_s=k} d(k, \alpha_s) - \beta V_L() \quad \text{où :} \quad (11-16)$$

$$V_L() = \sum_s \sum_{r \in V_s} \mathbf{1}_{\lambda_s \neq \lambda_r} \mathbf{1}_{b_{<s,r>} = 0} \quad (11-17)$$

Cette énergie est partagée en deux termes : le premier concerne l'information de base extraite des données, c'est en fait un terme de classification, et le second est un terme de régularisation.

L'image segmentée choisie est l'estimateur du MAP construit à partir de cette énergie. Cet estimateur peut être approximé par mise en oeuvre d'un recuit simulé ou d'un ICM. Le résultat retenu correspond à un minimum local de l'énergie ; un exemple de résultat est présenté dans la figure 11.5.

Le choix du paramètre β détermine l'importance du terme de régularisation par rapport aux informations données par les observations. Ce paramètre est donc très important : s'il est choisi trop grand, la configuration des étiquettes pourrait s'éloigner trop des données ; mais s'il est trop petit, la segmentation finale sera insuffisamment régularisée.

Nous calculons β en utilisant une liste de cas particuliers pour lesquels les probabilités conditionnelles sont choisies par avance ([AZEN87], [AZEN92a]).

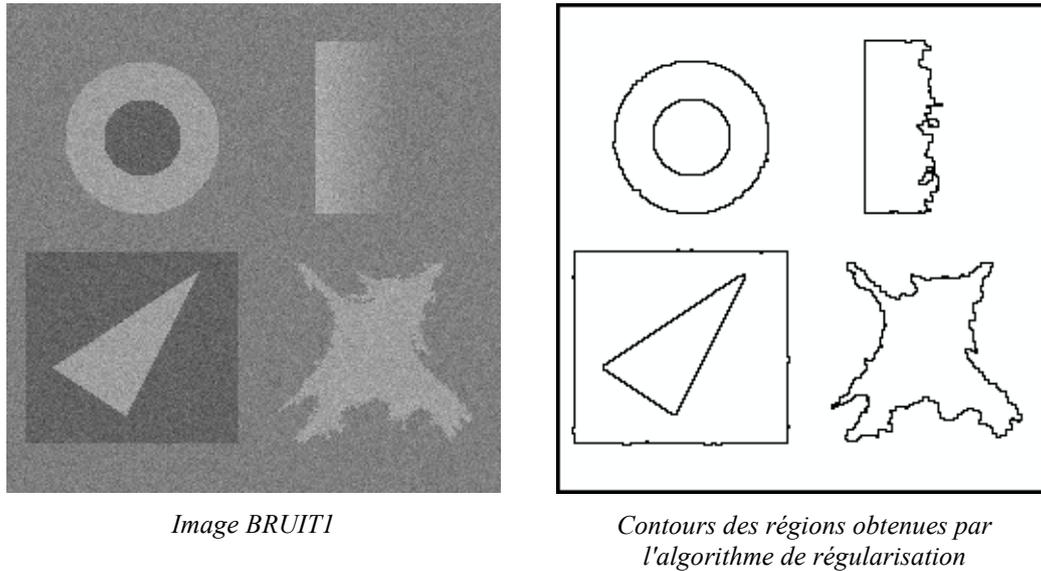


Figure 11.5 : Régularisation markovienne sur une image bruitée (bruit gaussien)

2 Relaxation

2.1 Introduction

Nous avons placé dans cette section les algorithmes stochastiques de segmentation non uniquement basés sur une régularisation stochastique ou déterministe d'une segmentation (ou classification) initiale, mais représentant réellement une auto-organisation de la configuration des étiquettes. Cette auto-organisation est de nouveau gérée par un ou des modèles markoviens.

2.2 Champ de Markov caché et segmentation non-supervisée

L'algorithme de ce paragraphe suppose que le nombre maximal de régions, disons q , est fixé. L'image observée $a = \{a_s, s \in \mathcal{S}\}$ est vue comme la réalisation d'un champ aléatoire \mathbf{A} . Le champ des étiquettes $\lambda = \{\lambda_s, s \in \mathcal{S}\}$ est vu comme la réalisation d'un champ aléatoire markovien $\mathbf{\Lambda}$ de système de voisinages $V = \{v_s, s \in \mathcal{S}; (s \notin v_s)\}$. Pour toute configuration λ , les a_s sont supposés conditionnellement indépendants :

$$P(a) = \prod_s P(a_s | \lambda_s) \quad (11-18)$$

et on note :

$$P(A_s = k | \Lambda_s = i) = P_i(k) \quad (11-19)$$

En pratique, on est amené à paramétrer ces probabilités, par exemple suivant une loi gaussienne $N(\mu_i, \sigma^2)$. Notons Φ l'ensemble de ces paramètres. La distribution de probabilité $P(\lambda)$ de $\mathbf{\Lambda}$ est définie par l'ensemble des probabilités

conditionnelles locales $\{P(\lambda_s | \lambda_r, r \in \mathbf{v}_s), s \in \mathcal{S}\}$, qui dépendent d'un vecteur β paramétrant $P(\lambda)$. λ étant un champ q -aire, on peut établir une classification des configurations locales possibles sur \mathbf{v}_s que l'on note $j = 1, \dots, J$. Par exemple dans le cas binaire ($q = 2$, $\lambda_s = 0$ ou 1), si \mathbf{v}_s est constitué des 4 plus proches voisins de s , alors si Λ est isotrope on a $J = 5$; les configurations locales étant classées suivant le nombre de sites r tels que $\lambda_r = 1$ dans le voisinage de s . On note :

$$P(\Lambda_s = i | \text{voisinage de type } j) = P_{ij} \quad (11-20)$$

Notre problème est d'estimer λ tout en ne connaissant pas les paramètres notés $\Theta = (\Phi, \beta)$. La segmentation est non-supervisée dans le sens où les caractéristiques Φ des régions sont inconnues et non-estimées au préalable. De par la formulation de notre problème, Λ peut être interprété comme un champ de Markov « caché par les observations » \mathbf{a} . Enfin, l'estimation directe de β est difficile puisque λ n'a pas été observé [YOUN89]. Ici, nous présentons un algorithme qui s'apparente au traitement des chaînes de Markov cachées présenté pour la première fois dans [BAUM72] et étendu au cas des champs de Markov dans [CHAL88].

En sur-paramétrant suivant $\Theta = (\{P_i(k)\}, \{P_{ij}\})$, une estimation *simultanée* de λ et Θ de type bayésienne peut être réalisée. L'algorithme associé est itératif et a une apparence d'algorithme EM [DEMP77]. Si $\theta^{(n)}$ désigne l'estimation de Θ après n itérations de l'algorithme, alors $\theta^{(n+1)}$ est solution de :

$$\max_{\theta} E \left\{ \ln(P_{\theta}(\cdot, \mathbf{A}) | \mathbf{a}, \theta^{(n)}) \right\} \quad (11-21)$$

où l'espérance est définie suivant :

$$E \left\{ \ln P_{\theta}(\cdot, \mathbf{A}) | \mathbf{a}, \theta \right\} = \sum \ln(P_{\theta}(\cdot, \mathbf{a})) P_{\theta}(\cdot | \mathbf{a}) \quad (11-22)$$

$P_{\theta}(\cdot, \mathbf{a})$ est la pseudo-vraisemblance que nous définissons comme suit [BESA74] :

$$P_{\theta}(\cdot, \mathbf{a}) = \prod_s P(a_s | \lambda_s) P(\lambda_s | \lambda_r, r \in \mathbf{v}_s). \quad (11-23)$$

L'expression de gauche, ci-dessus, est paramétrée par Θ car $P(a_s | \lambda_s)$ est paramétrée par Φ et $P(\lambda_s | \lambda_r, r \in \mathbf{v}_s)$ est paramétrée par β . Elle peut encore s'écrire :

$$P_{\theta}(\cdot, \mathbf{a}) = \prod_{ik} P_i(k)^{m_{ik}(\cdot)} \prod_{ij} P_{ij}^{n_{ij}(\cdot)} \quad (11-24)$$

où m_{ik} et n_{ij} sont des variables aléatoires. m_{ik} est le nombre d'occurrences dans λ telles que $\lambda_s = i$ et $a_s = k$ et n_{ij} est le nombre d'occurrences dans λ telles que $\lambda_s = i$ et le voisinage de s est de type j .

On montre alors que la solution de la maximisation en Θ est :

$$P_{ij}^{(n+1)} = \frac{E\{n_{ij} | a, \theta^{(n)}\}}{\sum_i E\{n_{ij} | a, \theta^{(n)}\}} \quad (11-25)$$

$$P_i(k)^{(n+1)} = \frac{E\{m_{ik} | a, \theta^{(n)}\}}{\sum_k E\{m_{ik} | a, \theta^{(n)}\}} \quad (11-26)$$

où les espérances sont définies suivant :

$$E\{n_{ij} | a, \theta\} = \sum n_{ij}() P_\theta(a) \quad (11-27)$$

$$E\{m_{ik} | a, \theta\} = \sum m_{ik}() P_\theta(a) \quad (11-28)$$

En pratique, on se donne une loi paramétrée des $P_i(k)$. Par exemple si $P_i(k)$ est issue de la loi gaussienne $N(\mu_i, \sigma^2)$, alors on montre que la distribution $\{P_i(k)^{(n)}, k = 1, 2, \dots\}$ se résume à :

$$\mu_i^{(n+1)} = \frac{\sum_s \gamma_s^{(n)}(i) a_s}{\sum_s \gamma_s^{(n)}(i)} \quad (11-29)$$

$$[\sigma^2]^{(n+1)} = \frac{\sum_i \sum_s \gamma_s^{(n)}(i) (a_s - \mu_i^{(n+1)})^2}{\sum_i \sum_s \gamma_s^{(n)}(i)} \quad (11-30)$$

où :

$$\gamma_s^{(n)} = P_{\theta^{(n)}}(\lambda_s = i | a) \quad (11-31)$$

est la distribution marginale *a posteriori*.

Les espérances conditionnelles $E\{ | a, \theta\}$ sont incalculables. Elles sont alors approximées par une méthode de Monte-Carlo utilisant l'échantillonneur de Gibbs, (cf. Annexe B), basé sur la distribution *a posteriori* :

$$P_\theta(a) P(a) P()$$

où $P(\lambda)$ est la distribution *a priori* dont les probabilités conditionnelles locales sont P_{ij} . Soit une suite de configurations de champs λ fournie par cet échantillonneur : $\lambda(1), \dots, \lambda(T)$, l'approximation des espérances est alors :

$$E\{n_{ij} | a, \theta\} \approx \frac{1}{T} \sum_t n_{ij}(t) \quad (11-32)$$

$$E\{m_{ik} | \mathbf{a}, \theta\} \approx \frac{1}{T} \sum_t m_{ik}(t) \quad (11-33)$$

De la même façon, on a l'approximation :

$$P_\theta(\lambda_s = i | \mathbf{a}) \approx \frac{1}{T} \sum_t 1_{\lambda_s(t)=i} \quad (11-34)$$

où $1_{\lambda_s(t)=i} = 1$ si $\lambda_s(t) = i$ et 0 sinon.

Ainsi, cet algorithme a-t-il été appelé algorithme EM gibbsien.

En même temps que Θ est estimé à l'étape n , la segmentation est aussi calculée, en estimant λ suivant la règle bayésienne suivante :

$$\lambda_s^{(n+1)} = i \text{ si } P_{\theta^{(n)}}(\lambda_s = i | \mathbf{a}) \geq P_{\theta^{(n)}}(\lambda_s = \ell | \mathbf{a}), \quad \forall \ell \neq i \quad (11-35)$$

Cette règle classique de la théorie de la décision statistique est appelée règle MPM en analyse markovienne d'image (*cf.* annexe B).

Pour compléter la description de cet algorithme, il resterait à exprimer les probabilités conditionnelles locales de $P_\theta(\lambda | \mathbf{a})$, dont nous avons besoin pour l'échantillonneur de Gibbs. Puisque chaque étape délivre une estimation de Θ , il suffit donc d'exprimer ces probabilités à l'étape $n+1$ en fonction des paramètres $\theta^{(n)}$. Pour des illustrations de l'emploi de cet algorithme, nous renvoyons le lecteur à [CHAL89]. Dans cet article, les expériences sont menées avec des lois gaussiennes $N(\mu_i, \sigma^2)$. Les résultats sur des images de petites dimensions apparaissent très satisfaisants compte tenu de la simplicité du modèle sous-jacent. Remarquons que cette méthode réalise en fait une séparation de mélange de lois. Dans les exemples traités, la séparation est bonne bien que les histogrammes des niveaux des gris soient unimodaux. Par ailleurs, toujours avec cette hypothèse de loi gaussienne, cet algorithme se montre très peu sensible à l'initialisation $\theta^{(0)} = \{\Phi^{(0)}, P_{ij}^{(0)}\}$. On choisit des distributions $P_{ij}^{(0)}$ uniformes et des paramètres $\Phi^{(0)}$ calculés à partir des quantiles de l'histogramme. Remarquons que pour traiter des images de grande dimension, il est nécessaire d'implanter l'algorithme EM gibbsien de manière locale, c'est à dire sur des imageries recouvrant l'image, afin de respecter l'hypothèse de stationnarité du modèle en ce qui concerne les P_{ij} et les $P_i(k)$. Enfin l'inconvénient majeur de cet algorithme réside dans le fait que le nombre de régions doit être petit, sans quoi le sur-paramétrage en P_{ij} pose des problèmes d'estimation statistique. Pour situer cet algorithme dans le contexte des champs markoviens cachés, on consultera [QIAN89].

2.3 Un algorithme non supervisé de segmentation

Nous allons décrire ici trois types d'algorithmes :

- Formation de germes (agglomération locale d'étiquettes).

- Croissance de régions de bas niveau (agglomération globale d'étiquettes).
- Croissance de régions de haut niveau.

Ces différents niveaux doivent être empilés afin de réaliser la segmentation finale [AZEN92b].

2.3.1 Formation automatique de germes de base

Ce modèle est utilisé pour l'initialisation du processus de croissance de régions et il est utilisé en première étape dans la chaîne de traitements. Il sera aussi utilisé alternativement avec le modèle de croissance de régions de bas niveau afin de raffiner la position des frontières. Cet algorithme sera appelé algorithme d'agglomération locale.

Ce modèle est similaire à des modèles de type restauration, incluant des éléments de bords implicites et dépend d'un seul paramètre σ qui peut être assimilé à l'écart-type toléré à l'intérieur des régions de base. Le système de voisinages utilisé est celui des 4 ou 8 plus proches voisins.

En pratique, et en chaque site s , nous examinons l'ensemble des étiquettes des voisins, et nous sélectionnons en ce site l'étiquette qui minimise le terme correspondant de la fonction d'énergie. Cette fonction d'énergie contient deux termes : un terme d'interaction avec les données et un second positionnant les bords.

Modèle : A chaque segmentation, nous associons une configuration d'étiquettes Λ . Pour ce modèle, la fonction d'énergie est de la forme suivante :

$$U() = U_1() - U_2() \quad (11-36)$$

$$U_1() = \sum_s \sum_{\lambda} \mathbf{1}_{\lambda_s = \lambda} \frac{(a_s - \mu_{\lambda})^2}{2\sigma^2} \quad (11-37)$$

où μ_{λ} est la moyenne des niveaux de gris pour la région d'étiquette λ .

$$U_2() = \sum_{\langle s,t \rangle} \left(\frac{2}{1 + (a_s - a_t)^2 / \sigma^2} - 1 \right) \mathbf{1}_{\lambda_s = \lambda_t} \quad (11-38)$$

Le premier terme U_1 contrôle l'homogénéité à l'intérieur d'une région, les variances internes étant normalisées par le paramètre σ^2 . Le second terme U_2 propage une étiquette λ si le module du gradient est petit, et pénalise l'agglomération si le gradient est grand.

Relaxation : La segmentation est initialisée par des étiquettes λ_s différentes en chaque site et $\mu_{\lambda_s} = a_s$. Nous avons utilisé un ICM (*Iterated Conditional Mode*, [BESA74]) afin d'optimiser cette énergie. Nous obtenons ainsi de petites mais très homogènes régions, respectant les contours.

2.3.2 Croissance de régions de bas niveau

Cette procédure est basée sur l'idée suivante : supposons que nous ayons obtenu une première segmentation (ceci est réalisé en particulier par l'algorithme précédent), le processus de croissance de régions est obtenu en réalisant une nouvelle segmentation de la configuration des étiquettes. Ainsi, une étape du processus de croissance de régions consistera à segmenter l'ensemble des étiquettes obtenus au niveau précédent.

A chaque niveau, la segmentation est réalisée par un champ markovien paramétré par le paramètre σ qui a la même signification que pour le modèle précédent.

Modèle : Remarquons tout d'abord que ce champ markovien est sensiblement différent du précédent : il opère sur des sites qui sont des régions et non plus sur les sites pixels. Le nouvel ensemble de sites est la liste des étiquettes obtenues au niveau précédent, il sera noté \mathbf{S}^L .

Le système de voisinages est défini de la manière suivante : deux étiquettes sont voisines si les régions correspondantes ont une frontière commune.

Soit Γ la nouvelle segmentation (Γ est indicée par les sites $\lambda \in \mathbf{S}^L$ et $\lambda \in \mathbf{R}$ signifie que le site λ appartient à la région \mathbf{R}). L'énergie a la forme suivante :

$$H(\Gamma) = H_1(\Gamma) - \alpha H_2(\Gamma) \quad (11-39)$$

$$H_1(\Gamma) = \sum_{\mathbf{R}} \left[\sum_{\langle \lambda, \mu \rangle | \lambda, \mu \in \mathbf{R}} \text{Dist}(\lambda, \mu) \right] \quad (11-40)$$

$$H_2(\Gamma) = \sum_{\mathbf{R}} \sum_{\langle \lambda, \mu \rangle | \lambda, \mu \in \mathbf{R}} f(\text{moyenne des gradients de la frontière entre } \lambda \text{ et } \mu) \quad (11-41)$$

où $\text{Dist}(\lambda, \mu)$ est la distance entre les vecteurs de caractéristiques de λ et μ (moyenne, variance, attributs de textures,...). Cette distance est de nouveau normalisée par les paramètres σ , et f est la fonction précédemment utilisée :

$$f(u) = \left(\frac{2}{1 + \frac{u^2}{\sigma^2}} - 1 \right) \quad (11-42)$$

Les deux termes H_1 et H_2 sont normalisés de manière à avoir le même ordre de grandeur. α peut être fixé égal à 1.

Optimisation : Elle est faite en utilisant un ICM qui se stabilise en général assez vite, en quelques itérations.

2.3.3 Croissance de régions de haut niveau

L'idée est cette fois d'utiliser un algorithme et une fonction d'énergie très semblables aux précédents, mais en incluant des informations de haut niveau telles que des critères géométriques sur la forme des régions.

Pour ce faire, il est relativement facile de modifier la distance entre deux étiquettes en utilisant des caractéristiques géométriques. Le type de critères géométriques utilisé sera clairement extrêmement dépendant des résultats recherchés. Par ailleurs, il ne serait pas raisonnable d'utiliser de tels critères géométriques avant d'avoir obtenu une première segmentation basée sur les informations de bas niveau.

Nous proposons des fonctions d'énergie de la forme :

$$K(\Gamma) = H_1(\Gamma) - \alpha H_2(\Gamma) - \beta K_1(\Gamma) \quad (11-43)$$

où H_1 et H_2 sont les mêmes termes que précédemment et K_1 est une fonction dépendant des attributs géométriques.

A cette étape de la procédure, nous travaillons avec un petit nombre d'étiquettes, si bien qu'une optimisation précise de la fonction d'énergie peut être obtenue.

Exemple de chaîne de traitement basée sur les trois algorithmes précédents

Il y a beaucoup de possibilités de combinaison des trois algorithmes décrits. La combinaison de base que nous proposons est la suivante :

1. *Initialisation* : algorithme d'agglomération locale (Formation des germes de base).
2. *A l'étape n* : Utiliser alternativement :
 - l'algorithme de croissance de région de bas niveau paramétré par σ_n
 - l'algorithme d'agglomération locale, paramétré par σ_n
3. *Itérer sur 2* : pour une suite croissante de valeurs de σ . (La valeur maximale de σ est fixée au départ ou dépend d'un test d'arrêt.)
4. *Haut niveau* : Algorithme de croissance de régions avec critères géométriques.

Cette procédure globale est totalement définie par la donnée de la suite des valeurs de σ .

Remarquons que l'utilisation récurrente de l'algorithme d'agglomération locale permet de raffiner les frontières des régions en utilisant à chaque pas les nouvelles caractéristiques des régions.

La figure 11.6 présente un résultat obtenu sur une image réelle.

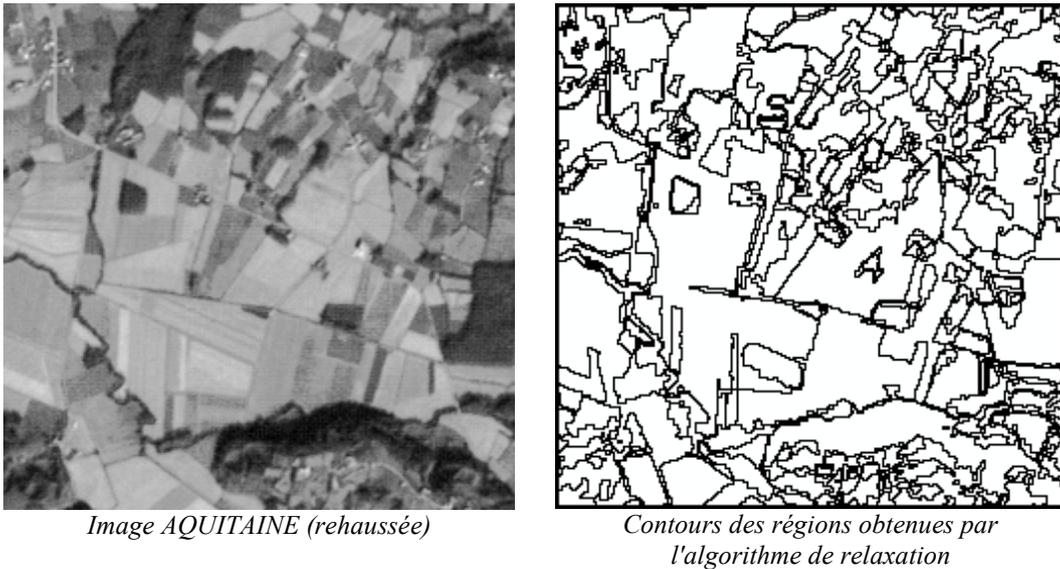


Figure 11.6 : Relaxation markovienne non-supervisée

2.4 Un algorithme semi-supervisé de segmentation d'images texturées

Ce modèle a été appelé semi-supervisé dans la mesure où il ne fait pas appel à une estimation de paramètres *a priori* sur des échantillons de textures. Cependant, un certain nombre de seuils doivent être choisis. Il est clair qu'ils ne seront adaptés qu'à un nombre relativement petit de types de textures et que donc ils représentent une forme d'information *a priori* sur les textures considérées. Pour plus de détail, nous renvoyons le lecteur à [GEMA90].

2.4.1 Modèle

L'ensemble des sites pixels est une grille $N \times N$, notée \mathcal{S}^P . L'ensemble des sites d'étiquettes est une autre grille \mathcal{S}^L qui est *a priori* plus petite que la précédente.

$$A = \{A_s\}_{s \in \mathcal{S}^P} \text{ et } \Lambda = \{\Lambda_s\}_{s \in \mathcal{S}^L}. \quad (11-44)$$

Nous n'avons pas besoin de connaître le nombre exact de textures présentes, mais seulement un nombre maximal M , alors :

$$\lambda_s \in \{1, 2, \dots, M\} \forall s \in \mathcal{S}^L. \quad (11-45)$$

Nous voulons caractériser chaque texture par un ensemble d'attributs. Ces attributs ne peuvent être calculés localement, mais sur des fenêtres glissantes de taille $(2p+1) \times (2p+1)$, centrées sur le site concerné (dans nos expériences, p prend des valeurs variant de 5 à 15). Les fenêtres considérées étant de taille relativement grande, et afin d'éviter de trop grands chevauchements de ces fenêtres, nous avons sous-échantillonné la grille d'observation des étiquettes (figure 11.7).

Pour chaque site étiquette λ_s , nous noterons \mathbf{y}_s le vecteur des niveaux de gris observés dans la fenêtre associée.

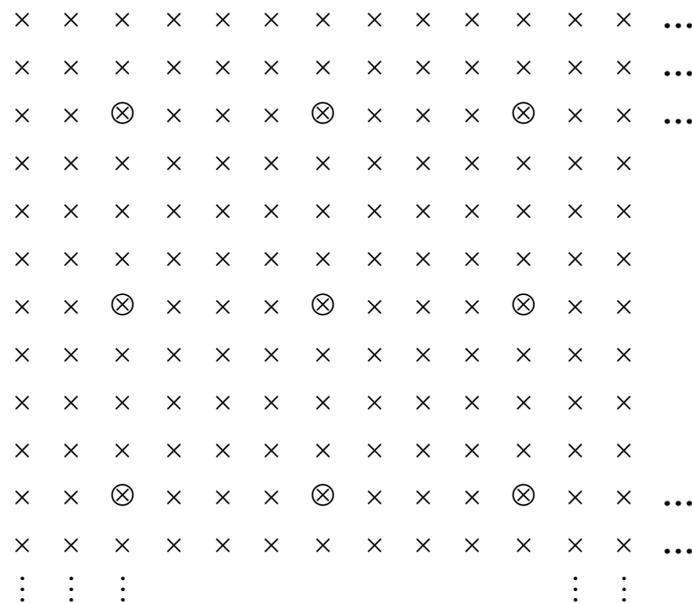
La loi jointe des processus est de la forme :

$$P(A = a, \dots) = \frac{1}{Z} \exp(-U_1(a, \dots)) \tag{11-46}$$

où U_1 est le terme d'interaction entre les étiquettes et les observations et Z une constante de normalisation.

Contrairement au modèle défini au paragraphe 1.2 de ce chapitre, nous avons choisi d'exprimer le terme de régularisation sous forme de contrainte. Ceci représente en fait une alternative valable pour chaque modèle, et il est donc possible de combiner un terme d'interaction U_1 avec un terme de type U_2 défini précédemment ou avec une contrainte tel que cela est décrit ici.

Le but est en fait de restreindre l'ensemble total des configurations Ω à un sous-ensemble de configurations Ω' qui vérifient les hypothèses *a priori*.



*x pour les sites pixels
 ⊗ pour les sites pixels associés à un site étiquette.
 G est la distance entre deux centres de fenêtres (ici G=4).*

Figure 11.7 : Grilles des sites pixels et étiquettes.

2.4.2 Définition de Ω'

La définition de Ω' peut légèrement varier suivant le type d'hypothèses *a priori* que l'on souhaite faire sur le type de segmentations recherchées. Nous allons donc donner une définition de Ω' qui peut être adaptée :

$$\Omega' = \{\omega \in \Omega : U_2(\omega) = 0\} \tag{11-47}$$

$$U_2(\omega) = V(\omega) + W(\omega) \quad (11-48)$$

$$V(\omega) = \sum_{s \in \mathcal{S}^L} \mathbf{1}_{\left(\sum_{t \in V_s} \mathbf{1}_{\omega_s = \omega_t} \right) > \xi} \quad (11-49)$$

$$W(\omega) = \frac{1}{2} \sum_{\tau_i} \sum_{s | s+\tau_i, s-\tau_i \in \mathcal{S}^L} \mathbf{1}_{\omega_s \neq \omega_{s+\tau_i} \text{ et } \omega_s \neq \omega_{s-\tau_i}} \quad (11-50)$$

où :

- ξ est un seuil donné
- $\mathbf{1}$ est la fonction indicatrice.
- V_s est un carré de \mathcal{S}^L , centré en s , de taille $(2p+1) \times (2p+1)$
- $\{\tau_i\}$ est un ensemble de translations ; nous avons généralement utilisé :

$\tau_1 = (0,1)$ (translation horizontale), $\tau_2 = (1,0)$ (translation verticale),

$\tau_3 = (0,2)$ (translation horizontale de longueur 2), et $\tau_4 = (2,0)$ (translation verticale de longueur 2).

L'énergie U_2 est ainsi composée de deux termes. Le premier terme V pénalise l'apparition de petites régions (le seuil ξ est choisi égal à $(p+1)^2$). Le deuxième terme W pénalise les frontières irrégulières.

Afin de restreindre l'espace de configurations à \mathcal{Q} , nous allons simuler suivant une distribution modifiée :

$$P(\mathbf{A} = \mathbf{a}, =) = \begin{cases} \frac{1}{z'} \exp(-U_1(\mathbf{a},)) & \text{si } \in ' \\ 0 & \text{sinon} \end{cases} \quad (11-51)$$

où z' est une constante de normalisation.

En pratique, nous avons simulé le modèle :

$$P(\mathbf{A} = \mathbf{a}, =) = \frac{1}{z} \exp(-U_1(\mathbf{a},) - \beta U_2()) \quad (11-52)$$

avec β tendant vers l'infini. Ainsi, lors des premières itérations, les configurations obtenues ne vérifient pas en général la contrainte, mais étant donné que β augmente (suivant un principe qui peut être comparé à un recuit simulé), nous obtenons finalement une configuration qui convient (la convergence de ce type d'algorithme a été démontrée dans [GEMA87c]).

Remarquons que U_2 possède de nombreux minimums locaux pour lesquels $U_2 = 0$, et que donc la contrainte ne devrait modifier que légèrement les minimums globaux de U_1 .

2.4.3 Energie U_1

Revenons maintenant à U_1 :

$$U_1(a_s) = \sum_{\langle s,t \rangle} d(\mathbf{y}_s, \mathbf{y}_t) \mathbf{1}_{\lambda_s = \lambda_t} \quad (11-53)$$

où $d(\mathbf{y}_s, \mathbf{y}_t)$ est une « distance » entre les deux vecteurs d'observations \mathbf{y}_s et \mathbf{y}_t , et $\langle s,t \rangle$ est une clique appartenant au système de voisinages que nous définirons plus loin. On aura donc tendance à accepter d'identifier deux étiquettes s et t si cette distance est « petite » et à refuser de les identifier si elle est « grande ». Bien entendu le fait de considérer que la distance est « petite » ou « grande » implique l'utilisation de seuils qu'il faudra choisir. Ainsi que nous l'avons signalé au début de ce paragraphe, les résultats obtenus dépendront fondamentalement des seuils qui auront été fixés et donc l'algorithme ne fonctionne pas totalement de manière non supervisée.

La « distance » $d(\mathbf{y}_s, \mathbf{y}_t)$ est définie par :

- $d(\mathbf{y}_s, \mathbf{y}_t) = -0.5$ si toutes les distances entre attributs de textures sont plus petites que le seuil correspondant.
- $d(\mathbf{y}_s, \mathbf{y}_t) = -0.5 + (\text{nombre d'attributs de textures pour lesquels la distance entre deux attributs est plus grande que le seuil})$ sinon.

La présence du terme -0.5 est due au fait que, sans ce terme, le minimum global de l'énergie serait 0 et serait atteint pour $\lambda_s \neq \lambda_t$, pour tous sites s et t , ce qui n'est évidemment pas le but recherché. Le modèle ainsi défini, les étiquettes λ_s et λ_t ont tendance à être les mêmes si $d(\mathbf{y}_s, \mathbf{y}_t) = -0.5$ (aucun des attributs de textures ne permet de distinguer les deux fenêtres d'observations), et ils ont tendance à être distincts si $d(\mathbf{y}_s, \mathbf{y}_t) \geq 0.5$ (au moins l'un des attributs de textures permet de distinguer les deux échantillons de texture observés).

2.4.4 Système de voisinages

Nous avons, pour ce modèle, utilisé un système de voisinages assez spécifique. Le choix de ce système a été guidé par les deux remarques suivantes :

- Si deux textures distinctes se trouvent dans deux régions non voisines d'une image, il faut être sûr que le système de voisinages utilisé permette de comparer ces deux textures, sinon, il serait possible que les deux régions se voient attribuer la même étiquette. Il faut donc un système de voisinages qui ne soit pas « local ».
- Du point de vue algorithmique, et même si les minimums globaux de la fonction d'énergie sont corrects, c'est à dire qu'ils fournissent bien les configurations recherchées, il est tout à fait possible que l'énergie ainsi écrite soit extrêmement difficile à optimiser parce qu'elle contient beaucoup de minimums locaux assez profonds. C'est exactement ce qui se passe pour l'énergie U_1 décrite ici dans le cas où on utiliserait un système de voisinages aux plus proches voisins. En effet, dans ce cas, une région homogène peut

se retrouver partagée en deux régions représentées par deux étiquettes différentes. La pénalité affectée à une telle sur-segmentation est relativement minimale dans la mesure où elle est fonction essentiellement de la longueur de la frontière alors que pour sortir de ce minimum local, il faudrait pouvoir remettre à jour tous les sites d'une des deux régions.

Nous avons donc choisi d'utiliser un système de voisinages aléatoire symétrique et régulier, de degré k fixé. Ceci signifie que chaque site a exactement k voisins, et que ce graphe est choisi suivant une loi uniforme sur l'ensemble de tels graphes [BOLO85]. Il est assez difficile de simuler de tels graphes, aussi nous avons utilisé d'autres graphes ayant les mêmes propriétés (cf. [GEMA90] pour la construction de ces graphes).

Les résultats expérimentaux utilisant de tels graphes se sont révélés très satisfaisants. En particulier, nous avons vérifié qu'aucun minimum local n'apparaissait lors de l'optimisation de U_1 , ceci restant cependant une conjecture dans la mesure où il n'a pas été possible de le démontrer.

2.4.5 Caractéristiques de textures et seuils

Notons a_i, a_j deux niveaux de gris ; ils appartiennent à $\{0, \dots, 255\}$. Pour une translation τ fixée, $P(a_i, a_j)$ est la probabilité empirique du couple $(A_s, A_{s+\tau})$ observée dans la fenêtre de taille $(2N+1) \times (2N+1)$ décrite précédemment. μ est la moyenne des niveaux de gris dans cette fenêtre. Les attributs de texture que nous avons utilisés sont les suivants [HARA73] :

- F_0 : l'histogramme de niveaux de gris dans la fenêtre.
- et, pour une translation τ fixée :

$$\text{— } F_1 \text{ (Inertie)} : \sum_{i,j} (a_i - a_j)^2 P(a_i, a_j)$$

$$\text{— } F_2 \text{ (\Phi-fonction, homogénéité locale)} : \sum_{i,j} \frac{1}{1 + (a_i - a_j)^2} P(a_i, a_j)$$

$$\text{— } F_3 \text{ (Energie)} : \sum_{i,j} P(a_i, a_j)^2$$

$$\text{— } F_4 \text{ (Entropie)} : \sum_{i,j} P(a_i, a_j) \ln P(a_i, a_j)$$

$$\text{— } F_5 \text{ (Cluster shade)} : \sum_{i,j} (a_i + a_j - 2\mu)^3 P(a_i, a_j)$$

Nous avons utilisé quatre translations définies par les vecteurs : $\tau_1 = (0,1)$, $\tau_2 = (1,0)$, $\tau_3 = (1,1)$, et $\tau_4 = (1,-1)$.

Pour l'attribut F_0 , nous avons utilisé la « distance » de Kolmogorov-Smirnov. Pour les attributs F_i , $i = 1, \dots, 5$, nous avons utilisé la valeur absolue de la différence que nous comparons à un seuil.

Les seuils peuvent être estimés *a priori* et réestimés en cours de procédure ou sur une liste d'échantillons de textures.

2.5 Un algorithme supervisé de segmentation d'images non-texturées

Nous nous intéressons ici au problème de la segmentation d'images non-texturées. Nous supposons que le nombre de classes est fixé *a priori* et que les paramètres caractérisant chaque classe peuvent être appris par avance sur des fenêtres de petites tailles.

2.5.1 Modèle

Soit $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$ la maille sur laquelle sont placés les sites pixels. A chaque site pixel est associé un site étiquette sur la même grille. Soit $\mathcal{L} = \{1, 2, \dots, K\}$ l'ensemble des étiquettes possibles correspondant aux K classes fixées *a priori* lors de l'apprentissage. Soit \mathcal{V} le système de voisinage considéré correspondant à un champ de Markov du 1^{er} ordre (*cf.* Annexe B) et c_s une clique définie sur \mathcal{V} et contenant le site s . Nous notons a_s l'intensité du pixel au site s ($\mathbf{a} = \{a_s, s \in \mathcal{S}\}$) et λ_s l'étiquette associée ($\boldsymbol{\lambda} = \{\lambda_s, s \in \mathcal{S}\}$). Enfin, nous appelons Ω l'ensemble de tous les étiquetages discrets possibles. Nous cherchons un étiquetage $\hat{\boldsymbol{\lambda}}$ qui maximise $P(\boldsymbol{\lambda}|\mathbf{a})$ (critère du MAP). En utilisant le théorème de Bayes sous l'hypothèse d'indépendance qui implique que $P(\mathbf{a}|\boldsymbol{\lambda}) = \prod_{s \in \mathcal{S}} P(a_s|\lambda_s)$, il vient :

$$P(\boldsymbol{\lambda}|\mathbf{a}) = \frac{P(\boldsymbol{\lambda})P(\mathbf{a})}{P(\mathbf{a})} \propto \prod_{s \in \mathcal{S}} P(a_s|\lambda_s)P(\boldsymbol{\lambda}) \quad (11-54)$$

De plus, nous supposons que $P(a_s|\lambda_s)$ est Gaussien :

$$P(a_s|\lambda_s) = \frac{1}{\sqrt{2\pi}\sigma_{\lambda_s}} \exp\left(-\frac{(a_s - \mu_{\lambda_s})^2}{2\sigma_{\lambda_s}^2}\right) \quad (11-55)$$

où μ_{λ_s} est la moyenne et σ_{λ_s} l'écart type de la classe $\lambda_s \in \mathcal{L}$. Ce terme correspond à la probabilité associée à l'énergie potentielle de cliques d'ordre 1 (singletons).

Pour tenir compte du contexte et favoriser l'apparition de zones homogènes, nous supposons que le terme $P(\boldsymbol{\lambda})$ est markovien :

$$P(\boldsymbol{\lambda}) = \exp(-U(\boldsymbol{\lambda})) = \exp\left(-\sum_{c \in \mathcal{C}} V_c(\boldsymbol{\lambda}_c)\right) \quad (11-56)$$

où $V_c(\boldsymbol{\lambda}_c)$ est un potentiel autologistique [GEMA84] de valeur $\pm\beta$ ($\beta \geq 0$) ne faisant intervenir que des cliques d'ordre 2 (doublets) et correspondant au modèle de Potts en physique statistique.

2.5.2 Choix du paramètre β

Le paramètre β détermine la taille des zones homogènes qui vont être créées au cours de la relaxation. Ce paramètre est donc important. Nous avons choisi une valeur adaptée à chaque type d'images traitées (β varie typiquement de 0,1 à 2).

2.5.3 Fonction énergie

La fonction énergie à minimiser s'écrit alors :

$$E() = \sum_{s \in \mathcal{S}} \left(\log \sqrt{2\pi} \sigma_{\lambda_s} + \frac{(a_s - \mu_{\lambda_s})^p}{2\sigma_{\lambda_s}^2} \right) + \sum_{c \in \mathcal{C}} V_c(c) \quad (11-57)$$

L'énergie locale s'écrit donc :

$$\varepsilon_s = \log \sqrt{2\pi} \sigma_{\lambda_s} + \frac{(a_s - \mu_{\lambda_s})^p}{2\sigma_{\lambda_s}^2} + \sum_{c_s} V_{c_s}(\lambda_{c_s}) \quad (11-58)$$

La fonction énergie $E(\lambda)$ présente plusieurs minimums. Différents algorithmes de relaxation ont été utilisés afin d'obtenir une bonne approximation du minimum global qui donnera la valeur de l'étiquetage recherché $\hat{\lambda}$.

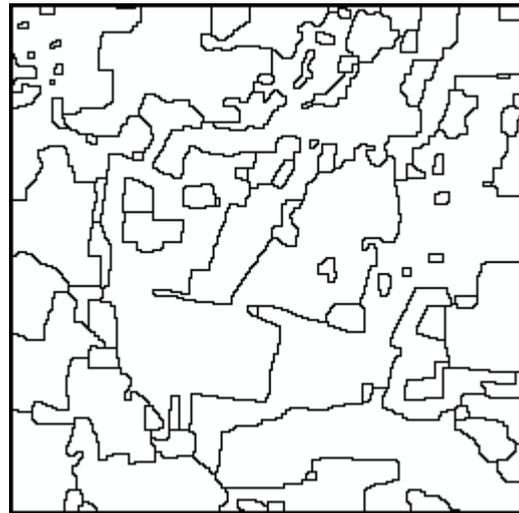
2.5.4 Relaxation

Nous avons travaillé avec deux méthodes stochastiques de relaxation : le recuit simulé avec l'échantillonneur de Gibbs [GEMA84] et l'algorithme de Métropolis, que nous avons comparées à deux méthodes déterministes : l'ICM (« *Iterated Conditional Modes* ») [BESA86] et le MMD (« *Modified Metropolis Dynamics* ») [KATO92] [KATO93]). L'algorithme du MMD est une variante pseudo-stochastique de l'algorithme de Métropolis. La différence se situe dans le choix du paramètre α qui permet d'accepter certaines configurations bien que ces dernières fassent croître l'énergie. Dans le cas de la dynamique de Métropolis, α est un nombre aléatoire choisi entre 0 et 1 selon une distribution uniforme alors que pour le MMD, α est une constante à valeur dans $[0,1]$ choisie *a priori* par l'utilisateur (*cf.* [KATO93] pour le lien entre le choix de α et le seuil de température au dessous duquel l'algorithme rejette toute croissance d'énergie). La valeur de α dépend de la qualité de l'image à traiter : α est choisi proche de 0 lorsque les données sont très bruitées, sinon α prend une valeur proche de 1 (typiquement 0,8 ou 0,9).

L'initialisation peut être identique pour toutes les méthodes de relaxation mais on peut également initialiser les algorithmes stochastiques avec des étiquettes tirées aléatoirement. Pour l'initialisation que nous avons choisie, seul le terme provenant de la partie gaussienne a été conservé, les paramètres de chaque classe étant appris par avance sur des fenêtres de dimensions adaptées à la nature de l'image.



Image AQUITAINE (rehaussée)



*Contours des régions obtenues
nombre de classes : 3
taille de la fenêtre d'apprentissage : 20×20*

Figure 11.8 : Relaxation markovienne supervisée

CHAPITRE XII

Approche région : méthodes structurales

1 Introduction

Ce chapitre est consacré aux méthodes de construction de régions qui comportent un fort aspect algorithmique du fait :

- de la taille des données à manipuler,
- des stratégies adoptées,
- de la structuration des données.

Contrairement au chapitre précédent, aucune hypothèse n'est émise sur la modélisation du processus « image » (de type markovienne, par exemple). Les hypothèses faites ici portent sur la nature de l'homogénéité des régions à extraire (qui se quantifie à l'aide d'un des attributs présentés au chapitre II), ainsi que sur la structure sous-jacente de graphe permettant d'étendre le concept de voisinage pixel à celui d'adjacence entre groupements de points.

Les stratégies employées sont ascendantes ou descendantes suivant qu'elles exploitent des mécanismes de croissance ou de division de régions. Dans le premier cas, le processus de segmentation fait intervenir initialement un niveau simple de représentation, pouvant aller jusqu'au niveau élémentaire du pixel. Dans le deuxième cas, le processus démarre à un niveau élevé qui peut correspondre à l'image entière.

Pour être efficace, la mise en œuvre de ces stratégies doit s'appuyer sur des structures de données ou des représentations de l'image comme celles présentées dans le chapitre III (maillage, graphe d'adjacence de régions, *quadtree*, ...).

Sont présentées dans ce chapitre des méthodes de croissance de région qui, partant du niveau du pixel, élaborent progressivement les régions en se basant sur des critères d'homogénéité. Le paragraphe 2 propose un algorithme général d'agrégation de pixels fondé sur un prédicat d'homogénéité à définir par l'utilisateur, alors que le paragraphe 3 se base sur l'histogramme local pour effectuer la croissance.

Le paragraphe 4 est consacré à des méthodes de division et de fusion utilisant comme structure de données différents types de graphes d'adjacence (quadtree, diagramme de Voronoï, pyramides).

2 Agrégation de pixels

La méthode d'agrégation de pixels, également appelée *croissance de régions*, est une technique ascendante qui, partant de la représentation de l'image comme un ensemble de pixels, les regroupe selon un double critère d'homogénéité et d'adjacence. Cette croissance est conduite par l'utilisation d'un mode de contrôle défini *a priori* et appelé prédicat.

Le prédicat, noté **Pred** par la suite, permet d'identifier une contrainte que doivent satisfaire les régions ; il peut être de nature géométrique (région rectangulaire, région convexe,...), de nature radiométrique (région homogène...) ou une combinaison des deux.

Exemples de prédicat d'homogénéité pour une région R :

- La variance $\sigma^2(R)$ des niveaux de gris de l'image associés aux points de la région R est inférieure à un seuil préfixé.
- La proportion $\alpha(R)$ de points dont le niveau de gris se situe hors de l'intervalle $[m(R) - \sigma(R), m(R) + \sigma(R)]$, où $m(R)$ est la valeur moyenne des niveaux de gris, ne dépasse pas un seuil préfixé.

Le prédicat est une fonction booléenne sur chaque région.

2.1 Algorithme général

Le processus crée les régions les unes après les autres et comprend, pour chaque région, une phase d'initialisation et une phase itérative [CHAS84].

Pour présenter les deux phases, on suppose que l'ensemble des sites S de l'image a été partiellement partitionné en $i-1$ régions R_j , $j = 1, \dots, i-1$. On s'intéresse à la génération de la région R_i qui sera construite dans l'ensemble

$$S^{(0)} = S - \bigcup_{j=1}^{i-1} R_j.$$

La figure 12.1 illustre cette étape de partitionnement partiel dans le cas où on aurait déjà identifié 5 régions et où le processus s'intéresse à la localisation d'une sixième région.

Le choix d'un germe $R_i^{(0)}$ peut se faire par seuillage sur un attribut (par exemple le niveau de gris), ou de toute autre manière (extraction de formes géométriques, *etc.*). Un germe peut éventuellement être réduit à un seul point.

L'algorithme 12.1 explique la croissance de la région R_i représentée par ses itérés $R_i^{(k)}$.

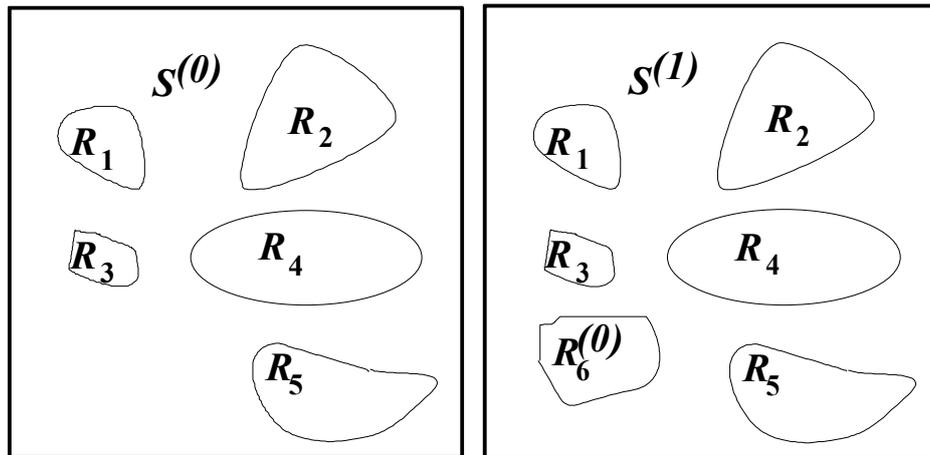


Figure 12.1 : Exemple de configuration pour la création d'une 6^{ème} région

2.2 Détail de la croissance de régions

La construction de $R_i^{(k+1)}$ s'effectue à partir de $R_i^{(k)}$, en lui adjoignant des points de $S^{(k+1)}$ défini par $S^{(k+1)} = S^{(k)} - R_i^{(k)}$, suivant la relation :

$$R_i^{(k+1)} = R_i^{(k)} \cup \{ s \in S^{(k+1)} \mid s \mathcal{R}_1 R_i^{(k)} \text{ ET } s \mathcal{R}_2 R_i^{(k)} \}$$

où \mathcal{R}_1 est une relation de type similarité radiométrique entre un site et la région croissante $R_i^{(k)}$ et où \mathcal{R}_2 est une relation de type connexité entre un point candidat et la région croissante.

Remarque

La relation \mathcal{R}_1 pour l'élaboration d'un ensemble X peut être spécifiée comme suit :

- Calcul des caractéristiques images (texture, intensité, couleur) sur l'ensemble des points agrégés à l'itération précédente, c'est-à-dire l'ensemble $R_i^{(k)} \setminus R_i^{(k-1)}$. Pour la première étape, cet ensemble est pris égal à $R_i^{(0)}$.

Le fait d'utiliser la caractéristique moyenne sur l'ensemble des points retenus à l'itération précédente $\{R_i^{(k)} \setminus R_i^{(k-1)}\}$ et non sur l'ensemble des points de l'itéré $R_i^{(k)}$ permet de satisfaire l'hypothèse d'homogénéité locale de la classe en question, tout en autorisant des variations possibles au niveau global de cette classe.

- Pour chacun des points de $S^{(k+1)}$, on calcule la distance à la caractéristique moyenne prédéfinie. Un point s de $S^{(k+1)}$ est désigné comme point candidat à l'agrégation si la valeur de distance est inférieure à un seuil prédéfini.

```

i := 0
/* a- Initialisation de la région  $R_i^{(0)}$  */
k := 0 ; /*numéro de l'itération*/
SI i = 1 ALORS  $S^{(0)} = S$  SINON  $S^{(0)} = S - \bigcup_{j=1}^{i-1} R_j$  ;

Choix d'un germe  $R_i^{(0)}$  ;

/* b- Phase itérative : construction des itérés  $R_i^{(k)}$  */
/* Mise à jour de l'ensemble  $S^{(k)}$  des points candidats à l'agrégation */
 $S^{(k+1)} = S^{(k)} - R_i^{(k)}$  ;
construction de  $R_i^{(k+1)}$  à partir de  $R_i^{(k)}$  et  $S^{(k+1)}$  ;
SI Pred( $R_i^{(k+1)}$ ) ALORS
    SI  $S^{(k+1)} = \emptyset$  ALORS ARRET /* toutes les régions vérifient le prédicat */
    SINON
        DEBUT
             $R_i = R_i^{(k+1)}$  ;
            i := i+1 ;
            ALLER en a) ;
            /*Initialisation d'une nouvelle région*/
        FIN
    SINON
        SI  $S^{(k+1)} = \emptyset$  ALORS
            ARRET /* i-1 régions vérifient le prédicat,
            la région i comprenant les points restants n'est pas homogène */
        SINON
            k=k+1 ;
            ALLER EN b) ; /* Poursuite de la croissance*/

```

Algorithme 12.1 : Agrégation de pixels

La relation \mathcal{R}_2 permet de construire un ensemble H constitué des seuls points de $\mathcal{S}^{(k+1)}$ qui sont connectés à l'itéré $R_i^{(k)}$.

L'itéré $R_i^{(k+1)}$ est alors défini par $H \cup R_i^{(k)}$.

2.3 Convergence

Le processus s'arrête selon deux scénarios :

- toutes les régions satisfont le prédicat (l'itéré $R_i^{(k)}$ vérifie le prédicat) ; la segmentation comprend i régions
- il y a $i-1$ régions vérifiant le prédicat, la $i^{\text{ème}}$ région n'est pas homogène, elle comprend les points ne vérifiant pas le prédicat.

Comme les itérés sont de surface croissante et bornée, la convergence est assurée au bout d'un nombre fini d'itérations.

2.4 Commentaires

L'algorithme présenté ci-dessus s'applique sur des images de niveaux de gris pour lesquelles l'attribut peut être un attribut de texture. La structure de données sous-jacente est le graphe d'adjacence de régions qui est construit de manière incrémentale. Par ailleurs, cet algorithme est parallélisable : les croissances à partir des différents germes $R_i^{(0)}$ peuvent être traitées simultanément par plusieurs processeurs.

Cette méthode a été utilisée dans le cas de la segmentation d'images en couleur pour lesquelles le prédicat attaché aux régions était la convexité [CHAS84].

3 Segmentation par corrélation d'histogrammes locaux

La méthode de segmentation présentée ici est fondée sur le principe d'une croissance de régions exploitant une description des textures à l'aide d'histogrammes locaux. Bien que l'étude de la distribution au premier ordre ne puisse rendre compte de toute la diversité des textures possibles, elle s'est avérée être un outil puissant dans le cas des images naturelles [LOWI83].

3.1 Généralités

On note h_D l'histogramme des intensités d'un domaine D et on considère une fenêtre W_s , glissant d'une région R_1 à une région R_2 de texture différente (cf. figure 12.2).

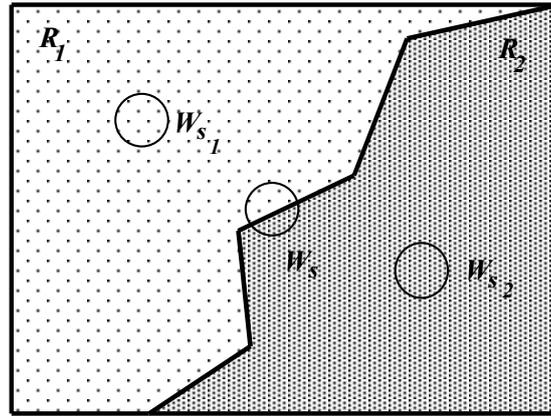


Figure 12.2 : Passage d'une texture à une autre

Les régions sont considérées comme homogènes. Si la taille des fenêtres W_s est suffisante et si $W_s \subset R$, on peut écrire que :

$$h_{W_{s1}} \approx h_{R1} \text{ et } h_{W_{s2}} \approx h_{R2} \quad (12-1)$$

Quand W_s recouvre une partie des 2 régions $R1$ et $R2$, on pose :

$$h_{W_s} = \alpha \cdot h_{W_s \cap R1} + (1 - \alpha) \cdot h_{W_s \cap R2} \quad (12-2)$$

où α représente la surface de $W_s \cap R1$ divisée par la surface de W_s (donc $0 \leq \alpha \leq 1$).

En première approximation, on a :

$$h_{W_s \cap R1} \approx h_{R1} \text{ et } h_{W_s \cap R2} \approx h_{R2} \quad (12-3)$$

et par conséquent :

$$h_{W_s} \approx \alpha \cdot h_{R1} + (1 - \alpha) \cdot h_{R2} \quad (12-4)$$

En représentant h par un vecteur dans l'espace des niveaux de gris, cette équation signifie que l'extrémité du vecteur h_{W_s} décrit le segment de droite compris entre les extrémités des vecteurs h_{R1} et h_{R2} .

On se place maintenant dans le cas simplifié où l'image est codée sur 3 niveaux de gris, a_1, a_2, a_3 . La norme de h_R est définie par :

$$\|h_R\| = \sqrt{h_R^2(a_1) + h_R^2(a_2) + h_R^2(a_3)} \quad (12-5)$$

Les iso-normes sont alors des sphères centrées en 0 ayant pour rayon la valeur de la norme. L'extrémité des vecteurs de même norme décrit un cercle (cf. figure 12.3), intersection de la sphère et du plan Π d'équation :

$$h_R(a_1) + h_R(a_2) + h_R(a_3) = 1 \quad (12-6)$$

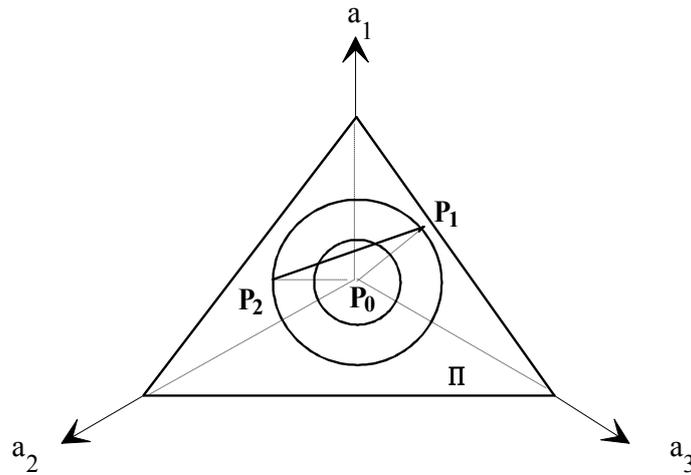


Figure 12.3 : Les iso-normes dans un espace à trois dimensions

Tous ces cercles sont centrés sur l'extrémité P_0 d'un vecteur h_0 de norme minimale obtenu pour une densité uniforme et dont les composantes sont données par :

$$h_0(a_1) = h_0(a_2) = h_0(a_3) = \frac{1}{3} \quad (12-7)$$

On se place dans le plan Π et on considère P_1 et P_2 les extrémités respectives de deux vecteurs h_{R1} et h_{R2} que l'on supposera, pour fixer les idées, de même norme r . P_1 et P_2 sont donc situés sur le cercle de rayon r (cf. figure 12.4).

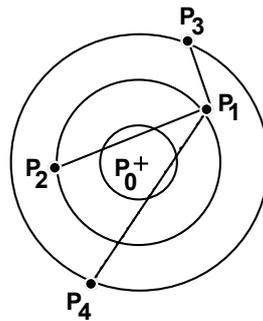


Figure 12.4 : Cercles concentriques de centre P_0 ; P_1 et P_2 extrémités de deux vecteurs de même norme, P_3 et P_4 extrémités de vecteurs de même norme (mais supérieure)

Quand W_s passe de la région R_1 à la région R_2 , l'extrémité P de son vecteur h_s décrit le segment de droite (P_1, P_2) , se situant alors sur des cercles concentriques de rayon inférieur à r (cf. figure 12.5). On considère maintenant P_3 et P_4 les extrémités respectives des vecteurs h_{R3} et h_{R4} de même norme supérieure à r . Le

passage de P_1 à P_3 se fait par une simple augmentation de la norme alors que pour celui de P_1 à P_4 , la norme de h_s passe par un minimum (cf. figure 12.5).

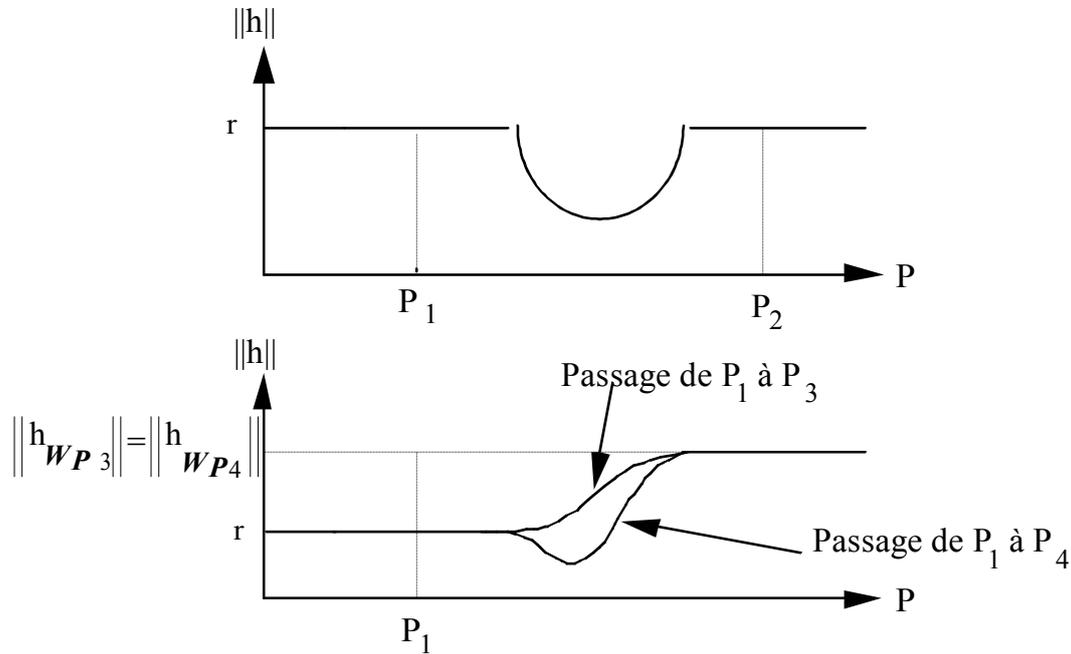


Figure 12.5 : Variations de la norme

Ainsi, ces variations de la norme euclidienne des histogrammes locaux mettent en évidence des frontières entre des régions « homogènes » mais différentes. Il est possible de définir un semi-ordre sur l'homogénéité des régions, à partir des observations suivantes :

- les maximums de la norme sont significatifs de régions très « homogènes »,
- les minimums de la norme sont significatifs de régions bruitées ou de frontières.

3.2 Principe

Compte tenu des considérations précédentes, il est possible d'élaborer des méthodes de segmentation basées sur la croissance de régions. Les germes des régions sont les points où la norme de l'histogramme local est maximale. A partir de chaque germe, on recherche de façon récursive tous les voisins ayant la même caractéristique, ici la même distribution. A chaque étape de la segmentation, on choisit de nouveaux germes (sur le même critère) dans le complémentaire des régions déjà créées.

Pour réduire la contribution des points aberrants et renforcer celle des points moyens, les histogrammes sont convolués avec une gaussienne [SECC90]. Dans la suite de ce paragraphe, *histogramme* sous-entend systématiquement *histogramme convolué avec une gaussienne*.

On suppose que les conditions (12-1) sont vérifiées. Les principes de base de deux algorithmes de croissance de régions utilisant les propriétés de la norme euclidienne des histogrammes locaux sont maintenant présentés. Ils exploitent un critère « d'homogénéité » basé sur la comparaison d'histogrammes locaux (deux points appartiennent à la même zone si *leurs histogrammes se ressemblent suffisamment*). La quantification de cette ressemblance est donnée par le calcul d'un « coefficient de corrélation » entre deux histogrammes.

Le coefficient de corrélation entre deux « vecteurs histogramme » h_1 et h_2 est donné par le produit scalaire normalisé :

$$r(h_1, h_2) = \frac{h_1 \cdot h_2}{\|h_1\| \|h_2\|}$$

$r(h_1, h_2)$ est compris entre 0 et 1. Il est maximum pour $h_1 = h_2$ et plus l'écart entre h_1 et h_2 est important, plus $r(h_1, h_2)$ est faible.

Soient deux sites s_1 et s_2 ayant respectivement pour histogrammes locaux h_1 et h_2 , si $r(h_1, h_2)$ est grand (supérieur à un seuil T), on considère que les points s_1 et s_2 appartiennent à la même région ; T fixe le degré d'homogénéité d'une zone.

3.3 Croissance de régions

Le germe potentiel d'une nouvelle région est le point non affecté à une région déjà créée et dont la norme de l'histogramme local est maximale (*cf.* § 3.1).

Les deux méthodes de segmentation reposent sur deux algorithmes proches mais néanmoins distincts :

- la *relaxation* : à partir d'un histogramme mesuré sur une zone de référence, on cherche à extraire la composante connexe à cette zone qui englobe tous les points présentant dans leur voisinage le même type de distribution.
- la *propagation* : c'est une réitération de la relaxation. Après l'extraction de la composante connexe, on définit, sur la limite de cette région, une nouvelle zone de référence qui va permettre d'extraire une nouvelle composante connexe et on recommence.

3.3.1 La relaxation

L'étape de relaxation comporte l'extraction de deux composantes connexes. Après la sélection du germe s d'une région, on extrait une première composante connexe C_s autour de s (*cf.* figure 12.6), définie par les points dont l'histogramme local est voisin de celui de s . La valeur du critère exigée pour cette première étape est assez faible mais le nombre de pixels de C_s est limité (*cf.* étapes 1, 2, 3 de l'algorithme 12.2).

Dans C_s , les pixels dont les histogrammes locaux sont les plus « corrélés » et qui sont donc les plus représentatifs de la région que l'on cherche à créer formeront la zone de référence. L'histogramme de cette zone Z_{ref} (qui englobe la

fenêtre W_s) servira d'histogramme de référence (cf. étape 4 de l'algorithme 12.2). La région R_s finalement extraite est une composante connexe autour du germe, contenant la zone de référence et une autre zone constituée des pixels les plus proches de l'histogramme de référence.

Trois seuils sont utilisés, le premier T_1 permet d'extraire la zone C_s . Le second T_2 permet de sélectionner les points qui vont intervenir dans le calcul de l'histogramme de référence. Le troisième T_3 permet l'extraction de la région R_s .

L'algorithme 12.2 présente les différentes étapes de cette méthode.

1. Pour tout t , calcul de l'histogramme h_{W_t} sur la fenêtre centrée en t .
2. Détermination du point $s = \arg \max(h_{W_t})$ dans la partie d'image non segmentée.
3. Extraction d'une composante connexe C_s autour de s :
 C_s contient un nombre défini de sites q , tels que
 $r(h_{W_q}, h_{W_s}) > \text{seuil } T_1$
4. Définition de la zone de référence $Z_{ref} \subset C_s$
 $Z_{ref} = \{ \text{sites } q \in C_s, r(h_{W_q}, h_{W_s}) > \text{seuil } T_2 \}$
 $h_{ref} = \text{histogramme sur } Z_{ref}$
5. Extraction de la région R_s contenant Z_{ref} connexe à s :
 R_s contient les sites q , tels que $r(h_{W_q}, h_{ref}) > \text{seuil } T_3$
6. Dilatation morphologique de R_s .
7. Etiquetage de R_s .
8. Retour en 2.

Algorithme 12.2 : Croissance de régions par corrélation d'histogramme et relaxation

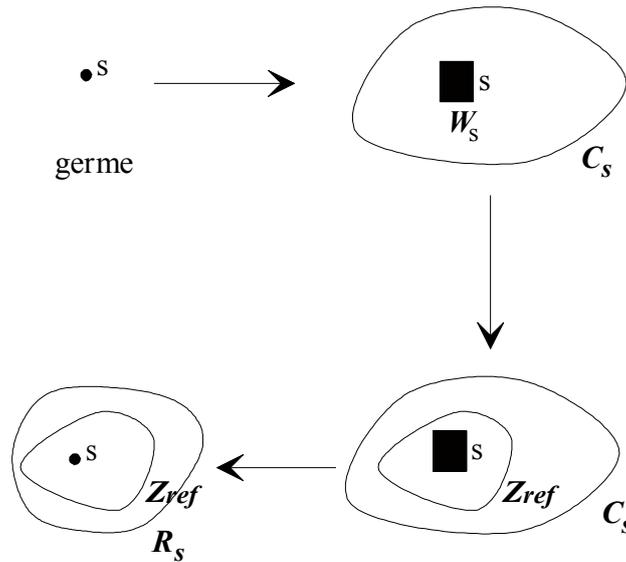


Figure 12.6 : Principe de la croissance de régions par corrélation d'histogrammes

La taille de la fenêtre W (en général carrée) dépend de la résolution de l'image traitée et peut aller de 3×3 à 11×11 .

L'étape 6 a pour but de s'assurer que le prochain germe n'est pas situé trop près de la frontière de R_s et qu'il pourra donner naissance à une nouvelle région. La dilatation s'opère en principe avec un élément structurant de petite taille (carré de taille 3×3 ou hexagonal de « rayon » 1).

Concernant les valeurs de seuil, on construit la première composante connexe C_s avec un seuil T_1 faible tant que la croissance est possible et tant que le nombre maximal de points de la zone de référence n'a pas été atteint. Ensuite, on forme Z_{ref} avec le seuil T_2 qui est le plus élevé. Ce procédé s'apparente à un seuillage par hystérésis.

On a ainsi : $1 > T_2 > T_3 > T_1 > 0$ et donc $Z_{ref} \subset C_s$ et $Z_{ref} \subset R_s$.

On peut ajouter une étape de validation de la zone de référence en comparant (avant l'étape 5) l'histogramme du germe et celui de Z_{ref} , et on ne poursuit le traitement que si $r(h_{W_q}, h_{ref})$ est suffisamment élevé.

Dans le cas où les textures sont des signaux dont l'amplitude varie lentement à l'intérieur d'une région, les histogrammes locaux et l'histogramme de référence finissent par ne plus se recouvrir, les coefficients de corrélation chutent jusqu'à devenir nuls et une région réelle peut être formée par une partition de sous-régions imaginaires : c'est le phénomène de sur-segmentation que l'on obtient sur des régions dont la taille est importante.

3.3.2 La propagation

Les premières étapes de la propagation (étapes 1 à 3) sont identiques à celles de la relaxation. Au lieu de définir la zone de référence et d'extraire la région R_s , on évalue la pertinence de la frontière de C_s . Pour cela, on cherche à déterminer les points de la frontière de C_s qui ne correspondent pas à de vrais contours.

Les points frontières f considérés sont les points de la frontière de C_s les plus décorrélés du point s . On cherche dans le voisinage de ces points f , les points qui sont également décorrélés du point s . Si on ne trouve pas de points de ce type, cela signifie que chaque point f est entouré de points ayant la même distribution que s . Donc, le point f ne correspond pas au contour de la région que l'on cherche à segmenter. f devient *point de propagation* pour la croissance de la région courante. Dans le cas contraire, le point f est considéré comme *point de contour* (cf. figure 12.8).

On va détailler de façon plus précise la détermination de ces points. Par l'extraction de la composante connexe C_s , on détermine un ensemble F de points frontières tel que :

$$r(h_{W_s}, h_{W_F}) \leq \text{seuil } T_0.$$

On considère ensuite les points situés dans un voisinage $V(f)$ du point frontière f . Deux cas se présentent alors pour f :

- $\forall q \in V(f), r(h_{W_q}, h_{W_s}) > T_1$, avec $T_1 < T_0$.

Tous les points au voisinage de f sont corrélés avec s à T_1 près, or f ne l'était pas à T_0 près. Comme T_1 est strictement inférieur à T_0 , on peut considérer que f ne représente pas une frontière de la zone que l'on cherche à segmenter : f devient point de propagation. Sur la figure 12.7, f_1 représente un tel point.

- $\exists p \in V(f), r(h_{W_p}, h_{W_s}) \leq T_1$

Il existe, dans le voisinage de f des points décorrélés avec s à T_1 près. On peut considérer que la frontière trouvée est un vrai contour puisque les zones de part et d'autre de f sont décorrélées : f devient *point de contour*. Sur la figure 12.7, f_2 représente un tel point.

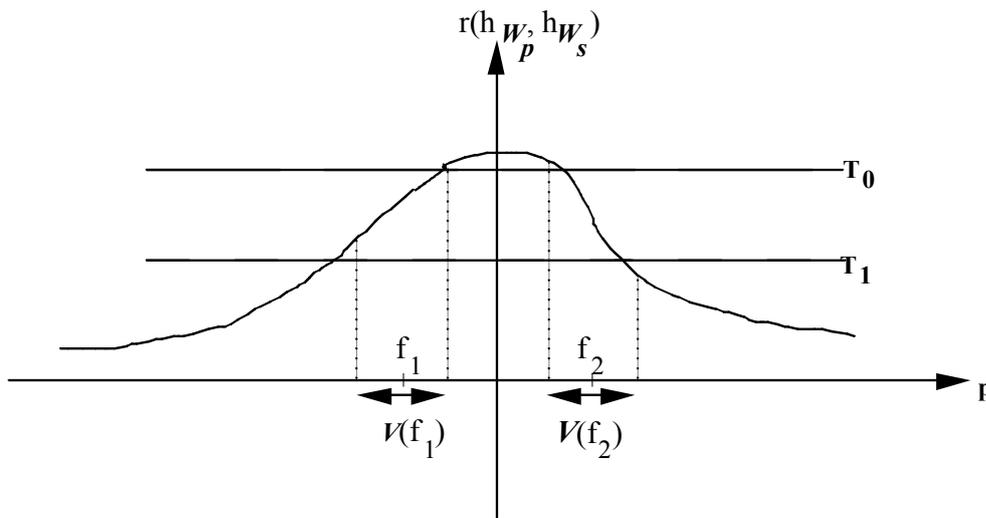


Figure 12.7 : Détermination des points de propagation et de contour

Une fois les points de propagation de la zone C_s déterminés, on cherche le prochain germe t de norme maximum. Si ce nouveau point est un point de propagation de la région R_s , alors on continue à faire croître cette région en prenant ce point comme point de départ pour l'extraction d'une composante connexe qui sera le prolongement de R_s . Si t n'a jamais été exploré, alors c'est le premier germe d'une région R_t et on commence à explorer cette région en extrayant une composante connexe à t . Le reste de la région sera segmenté par l'extraction successive des composantes connexes autour de tous les points de propagation trouvés pour cette région.

$\Delta T = T_0 - T_1$ détermine le gradient d'homogénéité que l'on accepte. On remarque que la propagation sera d'autant plus importante que ΔT sera grand : presque tous les points frontières seront des points de propagation.

L'algorithme 12.3 et la figure 12.8 décrivent la mise en œuvre de cette méthode. La figure 12.9 présente un exemple d'application sur une image (très difficile) de texture. L'algorithme extrait plus ou moins les régions, avec une imprécision sur les contours. Il est à noter qu'aucune autre méthode, présentée dans cet ouvrage, hormis la classification multidimensionnelle (cf. chapitre XIII § 2.6.2) n'a pu segmenter cette image.

1. Pour tout t , calcul de l'histogramme h_{W_t} sur la fenêtre centrée en t .
2. Détermination du point $s = \arg \max(|h_{W_t}|)$ dans la partie d'image non segmentée.
3. Extraction d'une composante connexe C_s autour de s :
 C_s contient un nombre défini de sites q , tels que $r(h_{W_q}, h_{W_s}) > \text{seuil } T_0$
- Soit F la frontière de C_s .
4. Détermination des *points de contour* :
 $\{ \text{sites } f \in F \mid \exists p \in V(f), r(h_{W_p}, h_{W_s}) \leq \text{seuil } T_1 \}$
5. Détermination des *points de propagation* :
 $\{ f \in F \mid \forall q \in V(f), r(h_{W_q}, h_{W_s}) > \text{seuil } T_1 \}$
6. Dilatation morphologique de l'ensemble des points de propagation : ensemble G
7. Recherche du prochain point t de norme maximum dans la partie non segmentée :
 SI $t \in G \cup F$
 ALORS retour en 3 pour calculer R_t qui sera une extension de R_s .
 SI t n'a jamais été exploré
 ALORS retour en 2 pour calculer une nouvelle région R_t .
8. Etiquetage de toutes les régions R_s .

Algorithme 12.3 : Croissance de régions par corrélation d'histogramme et propagation

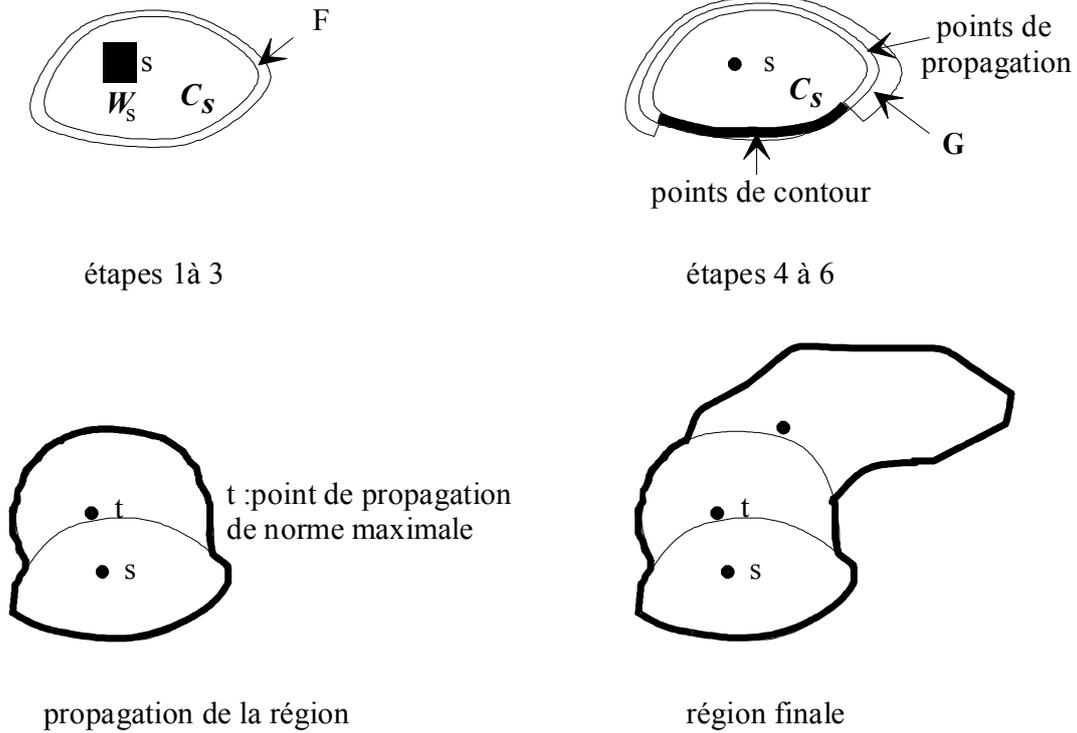


Figure 12.8 : Illustration de l'algorithme de propagation

3.3.3 Choix des seuils

Quand le seuil de corrélation T augmente, le nombre de couples (h_1, h_2) tels que $r(h_1, h_2) > T$ diminue. Les points sont moins facilement considérés comme corrélés, donc les composantes connexes extraites par croissance de régions se propagent moins loin. On constate donc que plus les seuils de corrélation sont élevés, plus le nombre de contours détectés est important.

Il existe une variante de cette méthode présentée dans [BRET89].

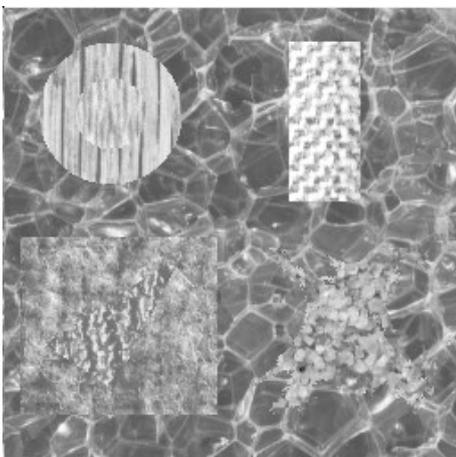
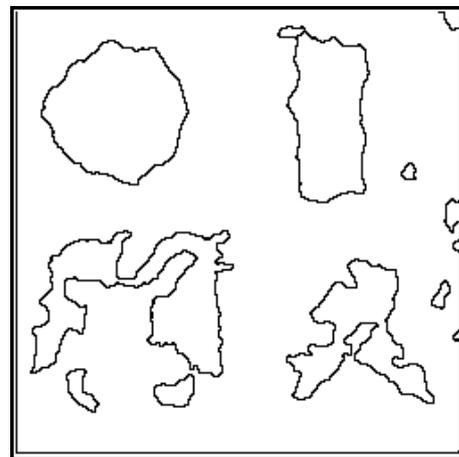


Image composite de textures



Contours des régions obtenues

Figure 12.9 : Corrélation d'histogrammes par propagation sur des fenêtres W de taille 11×11 , (gaussienne d'écart-type 1 et seuil de corrélation : 0.30)

4 Division et fusion guidées par les structures

Les méthodes de division et fusion procèdent par éclatement et regroupement de composantes de l'image décrite dans un mode de codage de type graphe d'adjacence (cf. chapitre III § 2.2.1).

4.1 Présentation générale

La segmentation initiale est représentée par un graphe d'adjacence de régions $G = (V, E)$ où V est l'ensemble des sommets et E l'ensemble des arêtes. Chaque sommet v_j est associé à une région R_j . Le graphe d'adjacence est non orienté, une arête est un couple de sommets (v_j, v_k) entre deux régions adjacentes.

4.1.1 Algorithme général de segmentation par division et fusion

Par définition de la segmentation, on a : $\bigcup_{j=1}^n R_j = S$, où n est le nombre de régions.

La forme générale des méthodes de division et fusion de régions est donnée par l'algorithme 12.4 [PAVL77] (**Pred1** et **Pred2** sont des prédicats d'uniformité).

```

/* module division */
t := VRAI ;
TANT QUE t FAIRE
    DEBUT
        t := FAUX ;
        POUR chaque sommet  $R_k$  du graphe FAIRE
            SI NON(Pred1( $R_k$ )) ALORS
                DEBUT
                    t := VRAI ;
                    partager  $R_k$  en  $N_k$  régions suivant un critère donné ;
                    mettre à jour le graphe ;
                FIN
            FIN
        FIN
/* module fusion */
t := VRAI ;
TANT QUE t FAIRE
    DEBUT
        t := FAUX ;
        POUR chaque sommet  $R_k$  adjacent à un sommet  $R_j$  FAIRE
            SI (Pred2( $R_j \cup R_k$ )) = VRAI ALORS
                DEBUT
                    t := VRAI ;
                    fusionner  $R_j$  et  $R_k$  ;
                    mettre à jour le graphe ;
                FIN
            FIN
        FIN
    FIN

```

Algorithme 12.4 : Division et fusion de régions

De ces algorithmes généraux peuvent être dérivés des algorithmes spécifiques de segmentation en choisissant les critères de fusion et les prédicats d'uniformité exploitant les attributs présentés au chapitre II.

4.1.2 Remarques sur la convergence et la complexité

- Le graphe d'adjacence limite est toujours défini, il s'agit au pire de celui correspondant au maillage ; ainsi la procédure division converge.

De même, la procédure fusion converge. Le graphe limite, quand le nombre de sommets diminue, possède un seul représentant à savoir l'image entière.

Dans l'algorithme précédent, toutes les divisions ont lieu avant les fusions, il n'y a pas de cycle possible.

- La charge de calcul d'une segmentation est liée :
 - au calcul des attributs associés aux sommets et aux arêtes du graphe.
 - à la mise à jour du graphe après chaque fusion ou division.

Il est clair que le choix des structures de données est capital en terme de coût de mise à jour. Indépendamment de ce problème d'implantation, l'actualisation des attributs des sommets est en général moins coûteuse après une fusion qu'après une division. Ceci est vérifié si, pour un attribut $A(R_i)$ associé au sommet v_i , il existe une loi de composition \heartsuit telle que :

$$A(R_i \cup R_j) = A(R_i) \heartsuit A(R_j)$$

(ce qui est le cas pour l'attribut moyenne des intensités avec la loi demi-somme).

- L'information obtenue à partir des régions est statistiquement plus fiable que l'information calculée à partir de quelques pixels. La fiabilité peut être perçue en terme de robustesse vis-à-vis du bruit.

4.2 Fusion de régions dans un graphe

Dans ce paragraphe, on se focalise sur le processus de fusion. Un coût de fusion est associé à chaque arête du graphe $G[V, E]$ (cf. figure 12.10), ce coût est intégré dans le prédicat $\mathbf{Pred}(R_j \cup R_k)$. L'ordre de fusion est primordial. Des ordres différents conduisent à des segmentations différentes, une hiérarchie de critères peut être établie (cf. § 4.3).

Une première stratégie consiste à détecter sur le graphe $G[V, E]$ l'arête de moindre coût à chaque itération. La segmentation est alors unique pour une image donnée [MUER68], mais il est difficile de contrôler une croissance équitable des régions dans l'image.

REPETER

sélectionner un sous-ensemble E' de E tel que chaque sommet v_k de V n'apparaisse qu'une seule fois dans E' ;

fusionner 2 à 2 les régions associées aux éléments de E' ;

mettre à jour V et E ;

JUSQU'A «plus de fusion possible».

Algorithme 12.5 : Fusion de régions dans un graphe

Il est possible de paralléliser l'algorithme en fusionnant à chaque itération plusieurs paires de régions définies chacune par une arête (cf. algorithme 12.5).

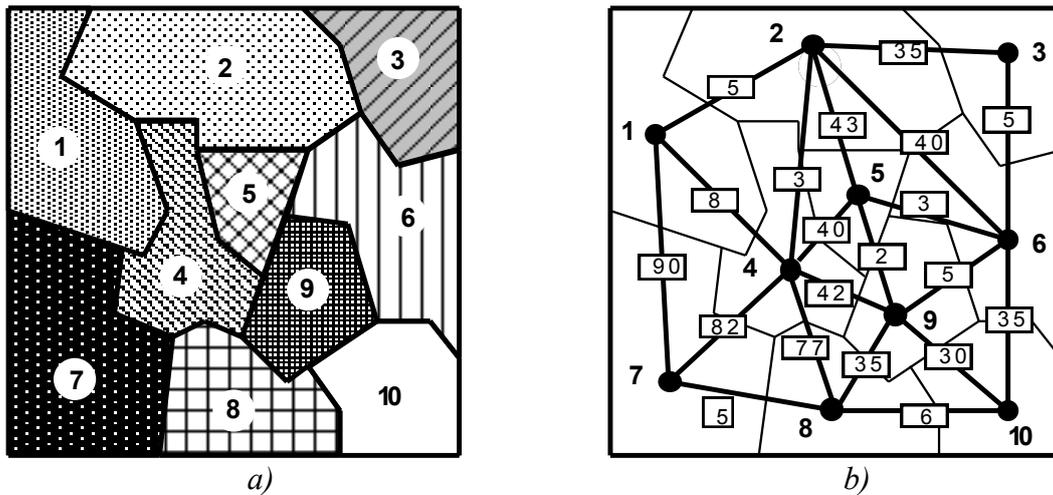
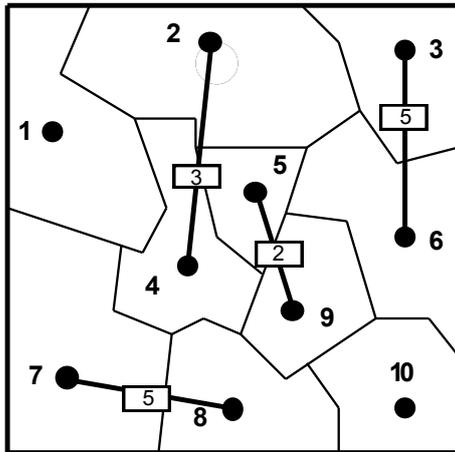


Figure 12.10 a) Partition en composantes connexes b) Graphe d'adjacence associé.
Les arêtes sont pondérées par un coût de fusion.

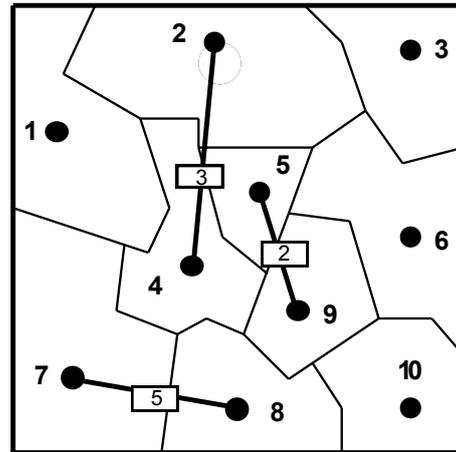
Le choix de E' doit garantir une « bonne » segmentation. Plusieurs approches ont été proposées dans la littérature :

- Soit lc le coût minimal de fusion des arêtes dans le graphe obtenu et soit t un seuil donné, les arêtes de coût de fusion inférieur à $(lc + t)$ sont affectées à E' [TILT88].
- Un seuil de coût de fusion est défini. Les minimums locaux stricts du coût de fusion, inférieurs ou égaux à ce seuil prédéfini, sont détectés sur le graphe d'adjacence et définissent E' (cf. figure 12.11a). Cette détection peut être réalisée en parallèle, mais elle échoue sur les régions déjà homogènes [BEVE89].
- Un graphe orienté $[V, A]$ est déduit du graphe d'adjacence $G[V, E]$: chaque sommet v_j sélectionne l'arc vers le sommet adjacent v_k de meilleur coût de fusion, si v_j est également le sommet de meilleur coût de fusion pour v_k alors (v_j, v_k) est une arête de E' ; c'est un test de reconnaissance mutuelle entre les régions (cf. figure 12.11b) [CLER91].
- Un seuil sur le coût de fusion est choisi, les arêtes de E' devront avoir un coût de fusion inférieur à ce seuil. On construit E' de la façon suivante : sélection de l'arête (v_j, v_k) de moindre coût, suppression des arêtes aboutissant aux 2 sommets v_j et v_k , sélection parmi les arêtes restantes de l'arête de coût minimal. On procède ainsi de suite jusqu'à avoir traité toutes les arêtes. Ou bien, pour ne pas avoir à rechercher l'arête de coût minimal, on peut se contenter de traiter les arêtes dans l'ordre où elles apparaissent dans la structure de données au lieu de rechercher systématiquement l'arête de coût minimal (cf. figure 12.12) [CHAN91].

Une fois E' défini, la fusion peut se faire en parallèle pour toutes les paires de régions à fusionner. Puis, le graphe d'adjacence doit être mis à jour avant l'itération suivante.

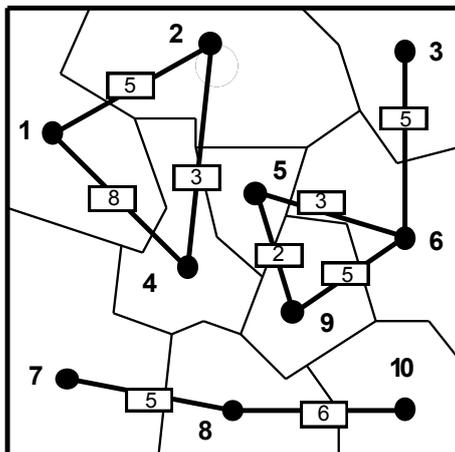


a) minimums locaux stricts du coût de fusion

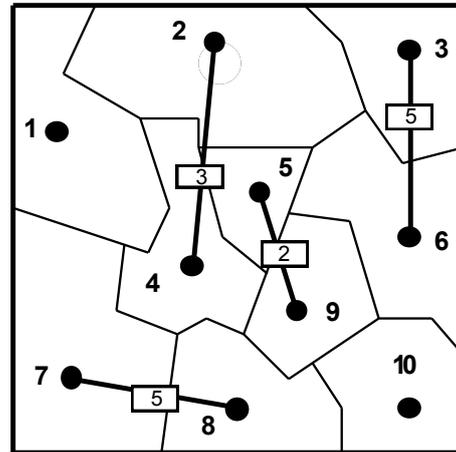


b) accord mutuel entre sommets

Figure 12.11 : Sélection de E' , à partir de la figure 12.10 (seuil fusion = 5)



a) restriction de E aux arêtes de coût de fusion inférieur au seuil (=15)



b) résultat de l'extraction itérative des arêtes

Figure 12.12 : Sélection de E' , à partir de la figure 12.10

4.3 Méthode de fusion multicritère

En reprenant le formalisme précédent, il est possible d'améliorer le résultat d'une segmentation en sélectionnant, dans la phase de fusion, le couple candidat (R_k, R_j) qui vérifie le prédicat d'uniformité et minimise une fonction de qualité.

Monga et Gagalowicz [MONG87][GAGA86] ont développé un algorithme qui utilise une séquence hiérarchisée de critères de fusion :

$$S = \{(\text{Pred}_1, \text{Qual}_1), \dots, (\text{Pred}_n, \text{Qual}_n)\}$$

Cet algorithme opère sur le graphe d'adjacence de régions. Chaque critère est un couple $(\text{Pred}_i, \text{Qual}_i)$ où Pred_i est un prédicat et Qual_i une fonction locale de qualité, comme par exemple :

$$\text{Pred}_1(\mathbf{R}) = \text{MAX-MIN} < t_1$$

$$\text{Qual}_1(\mathbf{R}) = \text{MAX-MIN}$$

MAX et MIN représentent respectivement le maximum et le minimum des niveaux de gris de la région \mathbf{R} , $\mathbf{R} = \mathbf{R}_k \cup \mathbf{R}_j$ et t_1 est un seuil prédéfini.

Un premier graphe d'adjacence de région est construit à partir d'une segmentation obtenue, par exemple, à partir d'une exploitation d'un arbre quaternaire (« *quadtree* ») [HORO74]. On opère, ensuite, autant d'étapes de fusion qu'il y a de critères dans la séquence hiérarchisée \mathcal{S} .

A chaque étape i de la séquence, les couples candidats $(\mathbf{R}_j, \mathbf{R}_k)$ sont classés suivant la valeur de la fonction de qualité $\text{Qual}(\mathbf{R}_j, \mathbf{R}_k)$ et les fusions se déroulent suivant l'ordre établi.

Cette méthode permet d'adapter la séquence de paramètres à la nature de l'image. Cependant, elle nécessite le réglage d'un nombre de seuils qui peut être important car il est lié au nombre de critères de la séquence.

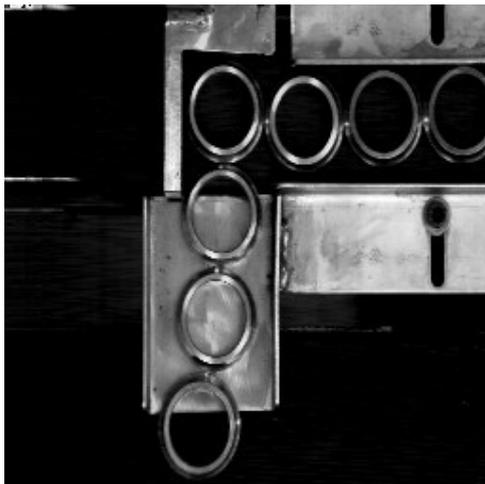
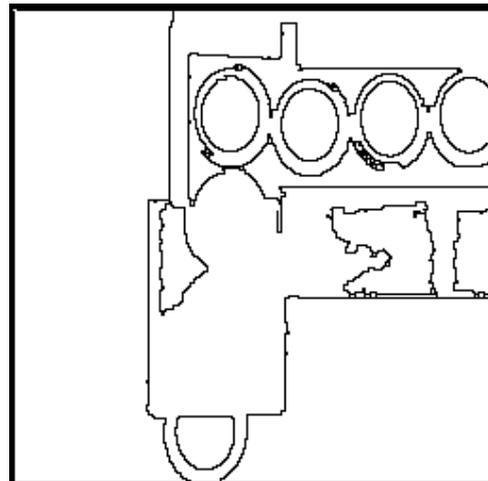


Image originale



Contours des régions extraites

Figure 12.13 : Fusion sur 4 critères
moyenne, variance, gradient le long de la frontière, surface

4.4 Partitionnement de Voronoï

Cette méthode consiste à chercher une partition de l'image en régions de Voronoï homogènes ; cette partition est décrite par un graphe d'adjacence [CHAS91]. A chaque polygone est associé un germe. Les germes sont initialement positionnés de manière aléatoire et ensuite introduits de manière guidée par l'image.

Le diagramme de Voronoï a été utilisé pour la segmentation de textures [AHUJ85] [TUCE90] [CHEN91].

Dans le processus de segmentation, cette phase de partitionnement sera suivie d'une phase d'agrégation de régions.

Le processus de type division-fusion permet de générer une partition de l'image en polygones de Voronoï. Il se compose d'une phase d'initialisation, d'une phase de division et d'une phase de fusion. Pour cela, il faut se définir un prédicat d'homogénéité et un ou plusieurs critères de fusion. Par la suite, le prédicat d'homogénéité sera lié à la mesure de la variance des niveaux de gris des points de la région et le critère de fusion fera intervenir la notion de germe inutile, notion précisée plus bas.

Au niveau de la mise en œuvre algorithmique, l'approche incrémentale de calcul du diagramme de Voronoï, présentée en annexe A est adaptée à la nature itérative de la méthode de segmentation.

L'algorithme proposé comprend trois phases :

Initialisation de la position des germes

Les coordonnées associées aux germes sont générées selon un processus de Poisson afin de réaliser une répartition uniforme des germes sur l'image (les coordonnées en ligne et en colonne sont générées de manière indépendante).

Phase itérative de division

Tant qu'il existe des polygones non homogènes, on effectue les opérations suivantes :

- Ajouter des germes dans les polygones non homogènes.

Pour chaque polygone ne satisfaisant pas le prédicat d'homogénéité, on ajoute un nombre de germes égal au nombre de sommets du polygone. On note g_0 le germe identifiant un tel polygone et s_i les sommets du polygone. Les nouveaux germes g_i seront placés, par exemple, sur chaque segment g_0s_i de telle sorte que $g_0g_i = \frac{2}{3}g_0s_i$.

- Mettre à jour le diagramme de Voronoï pour les germes ajoutés.

Phase itérative de fusion par suppression des polygones « inutiles »

La phase de fusion intervient lorsque de nouveaux germes ne peuvent plus être ajoutés. Elle consiste à réorganiser les germes compte tenu de leur environnement, plus précisément en comparant les polygones de Voronoï voisins. Pour cela, on fait appel à la notion de germe « inutile » :

Soit g un germe, on note P_g le polygone associé à g , et m_g la moyenne des niveaux de gris associés à P_g . Soit v le germe d'un polygone voisin au polygone P_g , on note $d(g, v)$ la longueur de l'arête commune aux polygones P_g et P_v .

Un germe g est *inutile* si son polygone associé vérifie les conditions suivantes :

- Le polygone P_g associé au germe g est homogène.
- Pour tout polygone voisin homogène, la différence entre la moyenne du polygone P_g et celle du polygone voisin est inférieure à un seuil préfixé noté *err*.
- La somme des longueurs des arêtes séparant le polygone P_g et les polygones voisins non homogènes, normalisée par le périmètre du polygone P_g , est inférieure à un seuil préfixé (on mesure ainsi l'importance relative des polygones non homogènes dans le voisinage du polygone P_g).

La suppression des germes inutiles ne perturbe pas la représentation de l'image et elle en fournit une meilleure approximation avec moins de polygones. Cette réorganisation au terme de la phase de fusion, permet de rendre la partition peu sensible à la phase d'initialisation des germes (*cf.* figure 12.14).

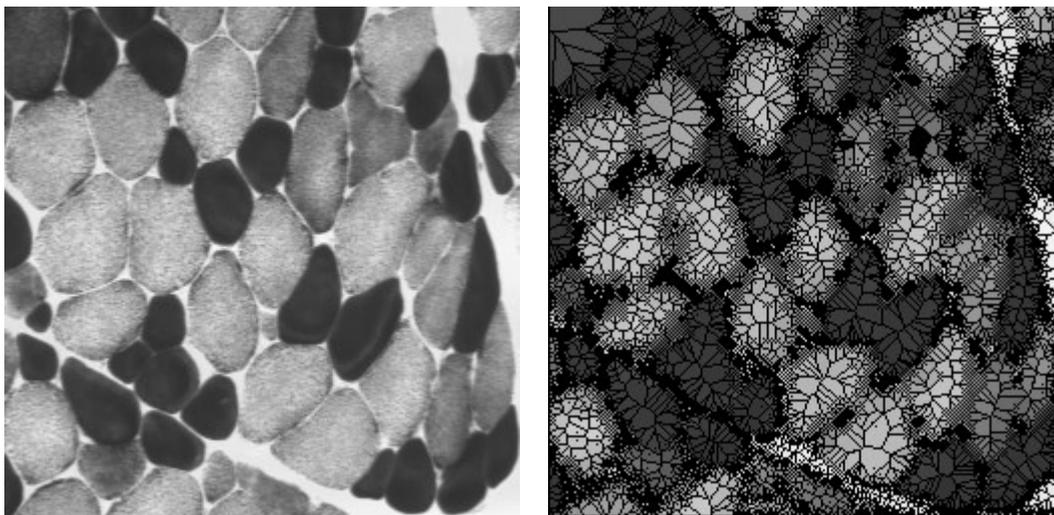


image originale

régions de Voronoï

Figure 12.14 : Partitionnement de Voronoï, les polygones et les germes sont tracés en noir, chaque région est affectée du niveau de gris moyen.

4.5 Arbre quaternaire (« *quadtree* »)

L'initialisation des algorithmes de segmentation fondés sur la structure de *quadtree* exploite la notion de coupure [PAVL77] [HUNT79] :

La coupure d'une arborescence est un ensemble de nœuds ayant les propriétés suivantes (*cf.* figure 12.15) :

- deux nœuds d'une coupure ne peuvent pas appartenir à un même chemin allant de la racine vers la feuille.

- une coupure est telle que tout nœud supplémentaire lui fait perdre sa qualité de coupure.

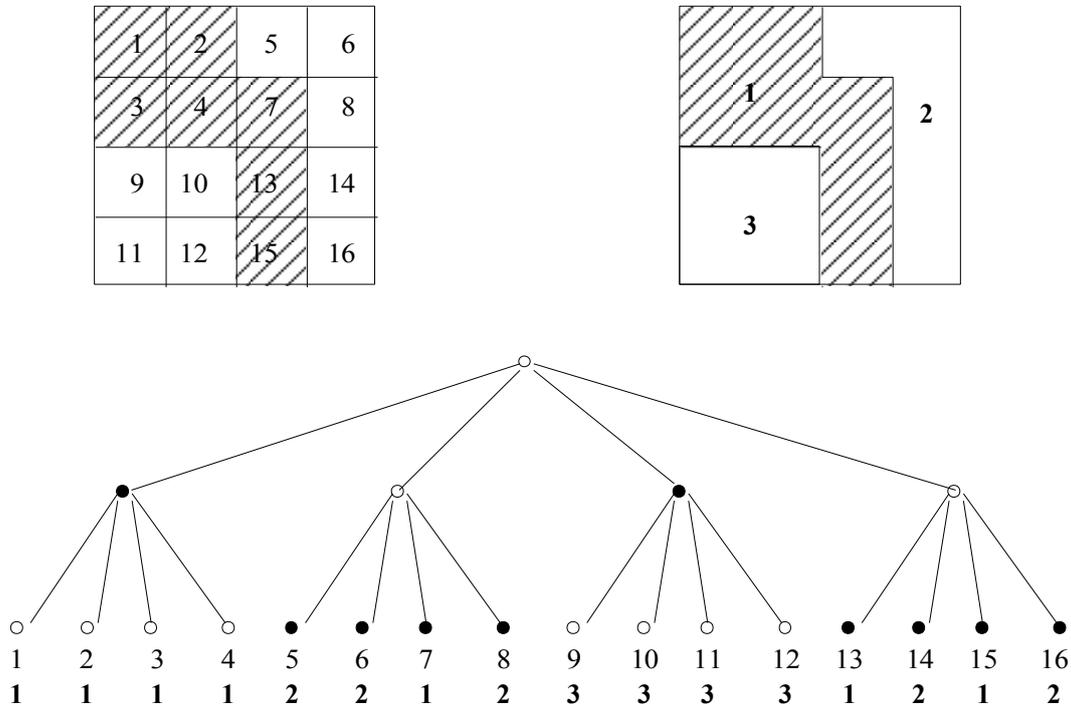


Figure 12.15 : Coupure dans un quadtree
 image de 16 blocs comportant 3 régions et quadtree associé ;
 les numéros de régions sont indiqués en gras, les nœuds noirs forment la coupure

Théorème :

Une coupure dans un *quadtree* définit une segmentation.

Preuve :

On se place aux extrémités du *quadtree*. Chaque feuille est marquée avec l'étiquette de sa région et placée dans un ensemble C . Ensuite, on effectue récursivement le traitement qui consiste à remplacer, quand c'est possible, chaque groupe de 4 nœuds de même étiquette par leur parent. On arrête quand il n'y a plus de regroupement possible.

De cette manière, tout chemin allant de la racine vers une feuille contient au moins un nœud dans C puisque les feuilles sont placées initialement dans C et qu'elles ne sont retirées que si elles ont un parent. Ainsi, aucun nœud dans C ne peut posséder plusieurs parents ou plusieurs successeurs et tout chemin allant de la racine vers une feuille ne contient qu'un élément de C et un seul ; il n'existe donc pas d'élément de C n'appartenant pas à un tel chemin.

Algorithme bidimensionnel de fusion et division

L'algorithme utilise les structures d'arbre de segmentation et d'ensemble de coupure. L'image doit être carrée, de dimension $N \times N$, avec $N = 2^L$. L'arbre de segmentation a la structure suivante :

- Chaque nœud de l'arborescence est une région carrée (appelée *bloc*), caractérisée par les coordonnées $[x, y]$ de son sommet supérieur gauche et par sa taille t .
- Les feuilles sont des blocs de taille $t = 1$, situés à une profondeur $L = \log_2 N$. La racine est à une profondeur zéro, sa taille est $t = N$, les coordonnées de son sommet sont $[1, 1]$.
- Les nœuds b de l'arbre sont définis de manière récursive :

Un nœud de profondeur k , a une taille $t = N/(2^k)$, son sommet supérieur gauche a pour coordonnées $[x, y]$. Ses quatre successeurs sont de taille $t/2$ et les coordonnées de leur sommet sont :

$$\begin{array}{cc} [x, y] & [x+t/2, y] \\ [x, y+t/2] & [x+t/2, y+t/2] \end{array}$$

Pour illustrer le fonctionnement de l'algorithme, on a choisi les attributs suivants associés à chaque sommet b :

- M : le maximum de la luminance sur b
- m : le minimum de la luminance sur b .

L'arbre n'est pas complètement mémorisé. Seul l'ensemble de coupure est stocké sous forme d'un tableau de taille maximum $N \times (\text{taille du vecteur d'attributs attaché à chaque bloc})$.

Pour la présentation de l'algorithme (*cf.* algorithmes 12.6, 12.7 et 12.8), le vecteur d'attributs est stocké dans 5 tableaux de dimension $\log_2 N$.

Pour un bloc de niveau k , on a :

$\text{taille}[k]$:	taille du bloc,
$X[k]$:	abscisse du sommet,
$Y[k]$:	ordonnée du sommet,
$M[k]$:	maximum de luminance,
$m[k]$:	minimum de luminance.

On utilise également une matrice d'étiquettes \mathbf{W} de taille $N \times N$.

```

/* initialisation*/
/* l0 : niveau de départ , t0 : taille initiale du bloc*/
    j := l0 ; t0 := N/2l0 ; t := t0 ;
/* déplacement */
    k := 1 ;
/* nombre initial de blocs*/
    n := 4l0 ;
    FUSION ;
    j := l0 ;
    ECLATEMENT ;

```

Algorithme 12.6 : Programme principal de fusion et division de régions utilisant un quadtree

```

Procédure FUSION
TANT QUE j > 0 FAIRE
    DEBUT
        DE i := 1 A n INCREMENT 4l0-j+1 FAIRE
            SI (taille[i]=taille[i+k]=taille[i+2k]=taille[i+3k]=t)
                ALORS
                    SI Pred1(bi ∪ bi+k ∪ bi+2k ∪ bi+3k) ALORS
                        DEBUT
                            /* fusion des 4 successeurs */
                            taille[i] := 2t ; taille[i+k] := 0 ; taille[i+2k] := 0 ;
                            taille[i+3k] := 0 ;
                            M[i] := MAX {M[i],M[i+k],M[i+2k],M[i+3k]} ;
                            m[i] := MIN {m[i],m[i+k],m[i+2k],m[i+3k]} ;
                            Mise à jour de l'image des étiquettes W ;
                        FIN
                    /*passage au niveau supérieur*/
                    j := j-1 ; t := 2t ; k := 4k ;
                FIN
    FIN

```

Algorithme 12.7 : Fusion de régions utilisant un quadtree

```

Procédure ECLATEMENT
  DE i := 1 A n FAIRE
    SI( 0 < taille[i] ≤ t0 ) ALORS
      TANT QUE (NON(Pred2(bi))) FAIRE
        DEBUT
          t := taille[i]/2 ; taille[i] := t ;
          taille[n+1] := t ; taille[n+2] := t ; taille[n+3] := t ;
          calculer M[i],M[n+1],M[n+2],M[n+3] ;
          calculer m[i],m[n+1],m[n+2],m[n+3] ;
          X[n+1] :=X[i]+t ; Y[n+1] :=Y[i] ;
          X[n+2] :=X[i]+t ; Y[n+2] :=Y[i]+t ;
          X[n+3] :=X[i] ; Y[n+3] :=Y[i]+t ;
          n := n+3 ;
        FIN
  FIN

```

Algorithme 12.8 : Division de régions utilisant un quadtree

4.6 Approches pyramidales

Parmi les méthodes de croissance de régions, les techniques relevant des structures dites pyramidales [JOLI92b] sont à considérer comme une construction d'une pyramide de graphes d'adjacence entre régions segmentées (cf. chapitre III § 3.2). Quoiqu'apportant, au fur et à mesure de la construction de la pyramide (de la base vers l'apex), des représentations de l'image initiale à échelle de plus en plus réduite, ces approches ne doivent pas être confondues avec les représentations d'images par des pyramides gaussiennes [BURT83] ou transformations en ondelettes [MEYE92], ces dernières relevant davantage de la théorie du traitement du signal.

La stratégie pyramidale consiste à fusionner les régions par groupes en parallèle, pour toute l'image, à chaque itération. On construit ainsi une pyramide de graphes d'adjacence. Un niveau donné de la pyramide met en jeu des régions sensiblement équivalentes en taille.

Ces méthodes se rattachent aussi aux recherches sur la parallélisation de la phase de fusion, dont une chronologie sera donnée plus loin.

En considérant la construction de la pyramide de graphes comme un problème de réduction de graphes, il est intéressant d'utiliser des notions de théorie des graphes tel que le *stable* (ou « *maximum independant set* », cette notion est détaillée dans le paragraphe 4.6.1).

La réduction doit bien sûr tenir compte du « contenu » du graphe, à savoir l'image. Cette adaptation à l'image peut être mise en œuvre de plusieurs manières.

4.6.1 Pyramides de graphes

Plutôt que de fusionner des couples de régions, des fusions de « petits groupes » de régions vont être provoquées. A chaque itération, ces fusions par groupes sont effectuées sur tout le graphe d'adjacence, ce qui définit la pyramide de graphes [MONT91].

La pyramide au niveau λ est un graphe non orienté $G[\lambda] = (V[\lambda], E[\lambda])$, où $V[\lambda]$ est l'ensemble des sommets et $E[\lambda]$ l'ensemble des arêtes. $G[0]$ représente l'image initiale avec une maille en 8-connexité¹⁰ ; les $G[\lambda]$ vont être construits successivement à partir de $G[0]$.

Définir $G[\lambda+1]$ à partir de $G[\lambda]$ est un problème de contraction de graphe où il faut :

- choisir les sommets de $V[\lambda+1]$ parmi ceux de $V[\lambda]$; ils sont appelés les *survivants* ;
- affecter tout non survivant du niveau λ à un survivant du niveau $\lambda+1$; c'est la génération des liens fils-père ;
- créer les arêtes de $E[\lambda+1]$ pour conserver l'adjacence entre les régions.

L'opération doit être parallèle pour tous les sommets d'un niveau ; de plus, les traitements doivent être locaux pour chaque sommet. L'apex de la pyramide est atteint lorsqu'aucune réduction supplémentaire n'est possible.

Une solution est fournie par la recherche d'un stable dans $G[\lambda]$, problème bien connu en théorie des graphes [LUBY86]. L'ensemble des sommets survivants est calculé pour être un stable ; il possède les propriétés suivantes (*cf.* figure 12.16) :

- tout sommet non survivant au niveau λ possède au moins un sommet adjacent qui va survivre au niveau $\lambda+1$, ce qui permet de créer, par traitement local, des liens fils-père qui garantissent la construction de régions connexes ;
- deux sommets adjacents ne peuvent survivre tous les deux, ainsi le nombre de sommets survivants décroît de manière significative, assurant la réduction de la pyramide.

De plus, il est possible de calculer un stable de manière parallèle sur le graphe et par des traitements uniquement locaux grâce à un procédé faisant intervenir des tirages aléatoires [MEER89a], [MEER89b].

Pour survivre, un sommet doit vérifier certaines contraintes vis-à-vis des sommets qui lui sont adjacents. La sélection des survivants et la création des liens fils-pères sont stochastiques mais répondent aux contraintes de parallélisme, de localité et de connexité. Le stable assure une bonne « répartition » des survivants sur le graphe.

¹⁰*cf.* chapitre 3 § 2.1.2

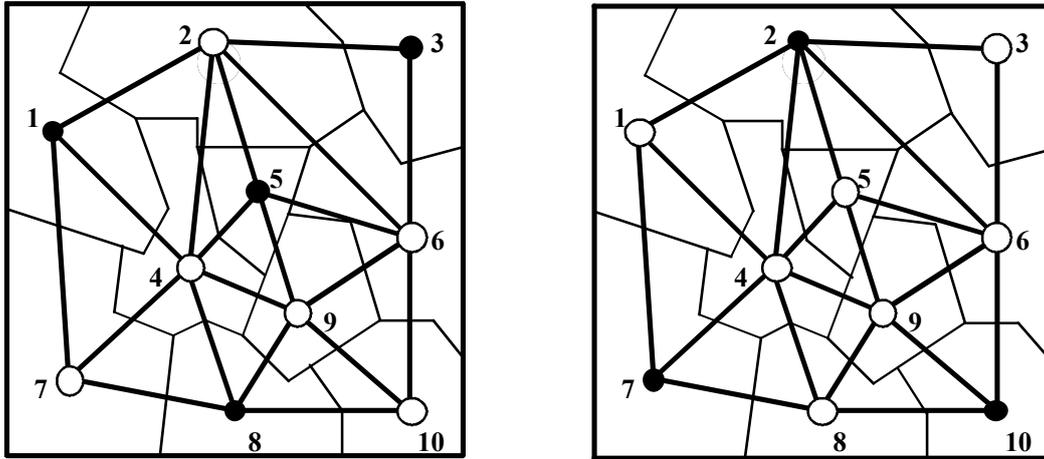


Figure 12.16 : Deux exemples de stables pour le graphe d'adjacence $G[\lambda]$ associé à la figure 12.10 a ; les sommets noirs appartiennent au stable.

Enfin pour calculer $E[\lambda+1]$, une arête n'est créée entre deux sommets de $G[\lambda+1]$ que s'il existe au moins deux sommets adjacents parmi les deux sous-ensembles de sommets fils dans $G[\lambda]$ (cf. figure 12.17).

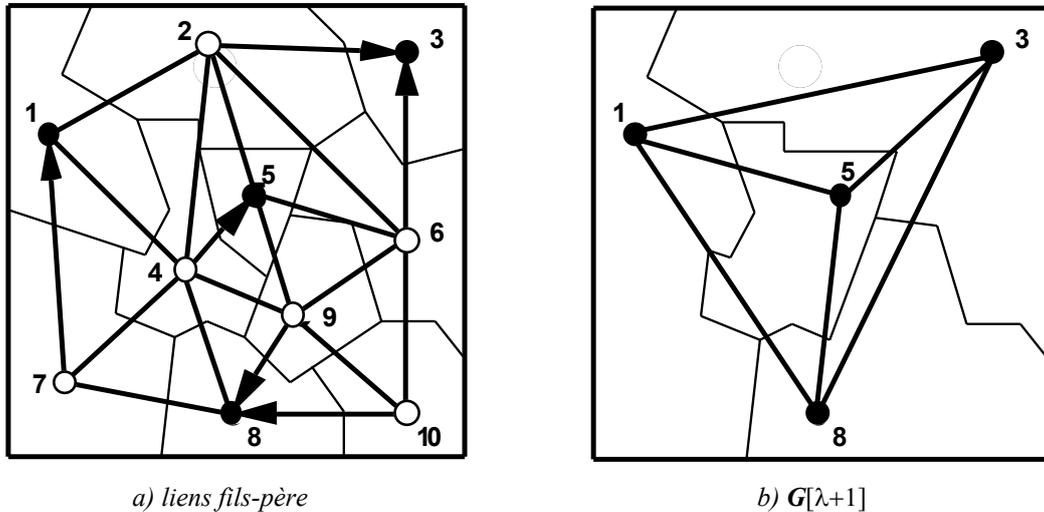


Figure 12.17 : Calcul de $G[\lambda+1]$ à partir de $G[\lambda]$ et du stable de la figure 12.16 a

Si la structure est construite selon le procédé décrit, sans aucune contrainte supplémentaire, on définit, de niveau en niveau, des pavages complètement irréguliers (et aléatoires) du support de l'image. Au dernier niveau, il ne reste qu'un sommet qui représente tout le support image. La construction doit être adaptée au contenu de l'image initiale.

Le procédé de recherche d'un stable se fonde sur la vérification de contraintes locales à chaque sommet et s'appuie sur la notion de graphe de similarité déduit

de $G[\lambda]$ que l'on appelle $G'[\lambda] = (V[\lambda], A[\lambda])$. Pour construire $G'[\lambda]$, il suffit de conserver dans $G[\lambda]$ les sommets « similaires » vérifiant les contraintes. Pour illustrer cette notion de stable, on définira la similarité à l'aide d'un seuil absolu T appliqué sur les différences entre niveaux de gris. Il est possible de généraliser la notion de similarité en utilisant des distances entre attributs associés aux sommets.

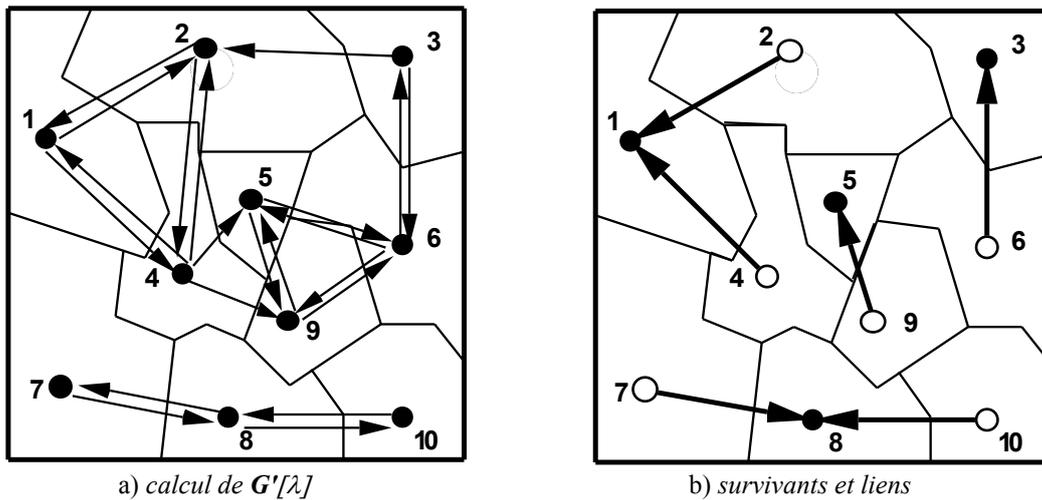


Figure 12.18 : Exemple de $G[\lambda+1]$ calculé à partir de $G[\lambda]$ associé au graphe valué de la figure 12.10 b

Soient c_0 un sommet, $(c_i)_{i=1, \dots, r}$ ses sommets adjacents et g_i les niveaux de gris associés. Dans le graphe G' , c_0 est relié à c_i s'il lui est similaire c'est-à-dire si $\delta g_i = |g_0 - g_i| \leq T$. Il est important de noter que G' est un graphe orienté.

Ceci revient également à travailler sur un graphe de similarité $G'[\lambda]$. Si un sommet non survivant « a le choix » entre plusieurs sommets comme père (et tous lui étant bien sûr similaires grâce au procédé de construction), il « se lie » au plus similaire. La figure 12.18 présente un exemple de construction de $G'[\lambda]$ à partir du graphe valué de la figure 12.10b dans laquelle les poids des arêtes correspondent aux différences δg_i . Le sommet 4 est susceptible d'être le fils des sommets 1, 2, 5 et 9 (avec un seuil de 50) et il est susceptible d'être le père des sommets 1 et 2 (avec des seuils égaux à 10).

Pour obtenir des résultats satisfaisants, l'environnement d'un sommet c_0 doit être analysé finement. Pour cela, après analyse des niveaux de gris des sommets adjacents à c_0 , on définit un seuil de similarité $S[c_0]$. Dans ce cas c_i est similaire à c_0 si

$$\delta g_i = |g_0 - g_i| \leq S[c_0] \leq T.$$

Puisque $S[c_0]$ est spécifique à c_0 le critère n'est plus symétrique (deux sommets adjacents analysent deux environnements différents ; le voisin c_k d'un sommet c_0 peut être considéré comme similaire à c_0 sans que la réciproque soit vraie), par

conséquent les graphes \mathbf{G}' sont orientés. Ce procédé assure qu'un sommet se situe en priorité par rapport aux voisins qui lui ressemblent le plus.

Le seuil de similarité $S[c_0]$ varie entre les deux valeurs extrêmes notées 0 et T. Il peut être affecté au contraste moyen maximum défini de la manière suivante :

soit $\{\delta g_{(i)}\}$ la suite ordonnée des $\delta g_i = |g_0 - g_i|$. Il existe 2 indices s et t tels que :

$$\delta g_{(0)} = 0 \leq \delta g_{(1)} \dots \leq \delta g_{(s)} \leq S[c_0] < \delta g_{(s+1)} \dots \leq \delta g_{(t)} \leq T < \delta g_{(t+1)} \dots \leq \delta g_{(r)} \quad (12-8)$$

On calcule les 2 suites

$$A_j = \sum_{i=1}^j \frac{\delta g_{(i)}}{j} \quad \text{et} \quad B_j = \sum_{i=j+1}^t \frac{\delta g_{(i)}}{t-j} \quad \text{pour } 1 \leq j \leq t-1 \quad (12-9)$$

On choisit $S[c_0]$ égal à l'écart le plus grand présent dans la suite $(B_j - A_j)_j$; il est déterminé comme suit :

$$u = \arg \max_j (B_j - A_j)$$

$$S[c_0] = \delta g_{(u)}$$

4.6.2 Compléments sur la stratégie de fusion

La stratégie de fusion présentée ci-dessus s'effectue en parallèle sur des groupes de régions.

D'autres développements reposent sur ce principe ou lui font suite :

- Le principe de la pyramide irrégulière peut être appliqué sur un arbre de recouvrement minimum extrait de l'image initiale pour élaborer une segmentation sous contrainte [MATH93] [MATH95].
- La variable aléatoire qui assure l'extraction du stable peut être biaisée pour forcer les régions considérées comme intéressantes à être extraites [JOLI92a].
- Les différentes segmentations obtenues par la composante aléatoire de l'extraction du stable peuvent être exploitées pour construire, par accumulation de résultats, une image de consensus de segmentation [MONT93].
- Un mécanisme de coopération entre régions et contours peut être introduit pour améliorer la robustesse du résultat à l'aspect aléatoire du procédé et aux variations de contraste [MONT92].

Enfin d'autres travaux peuvent être cités :

- La parallélisation de la phase de fusion (entre paires de régions) peut être reformalisée dans le contexte d'une structure pyramidale.
- Les structures pyramidales irrégulières peuvent être exploitées sur des images de contours assurant ainsi un meilleur contrôle des voisinages [KROP90].

CHAPITRE XIII

Comparaison des résultats : approche région

De même que le chapitre IX a présenté des tests comparatifs des méthodes d'extraction de contour, ce chapitre a pour but de comparer un certain nombre des algorithmes de segmentation par l'approche région qui ont fait l'objet des chapitres X à XII. Ces algorithmes sont comparés, de manière qualitative sur le même jeu d'images que dans le chapitre IX. Pour la comparaison quantitative, une série d'images bruitées et texturées a été fabriquée. Mais, contrairement à l'approche contour, il n'existait pas dans la littérature de méthode de comparaison d'images segmentées. Il a donc fallu définir une mesure - en fait une distance - entre images de régions ; celle-ci est présentée dans le paragraphe 1 ci-dessous.

Les méthodes testées ont été implantées dans des laboratoires différents (ETIS, INRIA Sophia-Antipolis et Rocquencourt, TIMC, LAMII, Matra Cap Systèmes et LAMII) et sur des machines différentes ; c'est pourquoi nous ne fournissons pas de tableau comparatif des temps de calcul.

1 Définition des critères de comparaison

Le problème de la comparaison quantitative des méthodes de segmentation en régions se présente sous la forme suivante : on dispose d'une image de test et des images de régions obtenues par les différentes techniques de segmentation que l'on veut comparer.

Deux situations peuvent se produire selon que la segmentation idéale est connue ou non. La première situation correspond, par exemple, à une image synthétique **A** dont toutes les régions sont connues et les résultats de segmentation sont obtenus sur l'image **A** bruitée ; la seconde concerne la majorité des cas : la segmentation idéale n'est pas connue *a priori*.

1.1 Une distance entre deux images de régions

Vinet [VINE91] a proposé une mesure de dissimilarité entre deux images d'étiquettes dans le cadre de la stéréovision. A partir de cette mesure, on construit une distance entre deux images d'étiquettes ; distance qui va être utilisée dans ce chapitre pour comparer deux images de régions.

1.1.1 Mesure de Vinet

On considère une image, de maillage S , et deux segmentations en régions de cette image, notées R et U . R_1, \dots, R_m sont les régions de la première segmentation, U_1, \dots, U_n celles de la deuxième. On note N le nombre de pixels de l'image ($N = \text{card } S$). Le principe de cette mesure consiste à déterminer les couples de régions assurant un recouvrement maximum entre les deux segmentations et à caractériser la dissimilarité par la proportion de pixels ne participant pas à ce recouvrement. Pour cela, on définit la table de superposition des étiquettes T par :

$$T(i,j) = \text{card}(R_i \cap U_j) \quad i=1, \dots, m ; j=1, \dots, n \quad (13-1)$$

puis on recherche le couple de régions (R_{i_1}, U_{j_1}) de recouvrement maximal ($T(i_1, j_1) \geq T(i, j) \quad \forall i, j$). On note C_1 la quantité $T(i_1, j_1)$ et on recommence la recherche des régions de recouvrement maximal. On détermine :

$$(R_{i_2}, U_{j_2}) \text{ tel que } T(i_2, j_2) \geq T(i, j) \quad \forall i \neq i_1, j \neq j_1$$

et ainsi de suite jusqu'à (R_{i_k}, U_{j_k}) tel que :

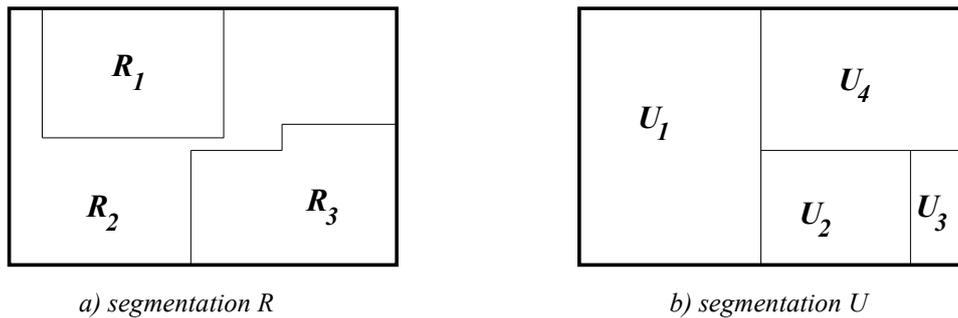
$$T(i_k, j_k) \geq T(i, j) \quad \forall i \neq i_1, i_2, \dots, i_{k-1} \quad \forall j \neq j_1, j_2, \dots, j_{k-1}.$$

Le nombre de couples de régions est $k = \min(m, n)$.

La mesure de dissimilarité est ensuite calculée par :

$$dv(R, U) = \frac{N - \gamma}{N} \text{ avec } \gamma = \sum_{i=1}^k C_i \quad (13-2)$$

On présente en figure 13.1 un exemple de calcul de cette mesure, les couples assurant un recouvrement maximum à chaque étape sont, dans l'ordre : (R_2, U_1) , (R_3, U_2) et (R_1, U_4) , on a $dv(R, U) = 0.53$. Le recouvrement est de 47 pixels.



T	U_1	U_2	U_3	U_4
R_1	20	0	0	3
R_2	27	0	0	23
R_3	0	17	7	3

c) table de superposition

Figure 13.1 : Mesure de Vinet entre 2 segmentations, card $S = 100$

1.1.2 Une distance entre deux images de régions

L'inconvénient de l'opérateur de Vinet est qu'il ne vérifie pas l'inégalité triangulaire. Pratiquement, cela signifie que si on modifie le classement d'un pixel dans une segmentation donnée, γ (cf. eq. 13-2) peut varier de plusieurs unités.

La table de superposition des étiquettes peut être vue comme le tableau des coûts associés à un problème de *couplage* (cf. annexe A § 1.3) entre les régions de deux images d'étiquettes. Les régions de ces deux images constituent les sommets d'un *graphe non orienté, valué et biparti* (cf. annexe A § 1.2) ; deux sommets sont reliés par une arête si le recouvrement entre les régions est non vide. Le poids associé à une arête de ce graphe est la valeur du recouvrement entre les deux régions correspondantes. Par exemple, à la table c) de la figure 13.1, on peut faire correspondre le graphe valué biparti de la figure 13.2.

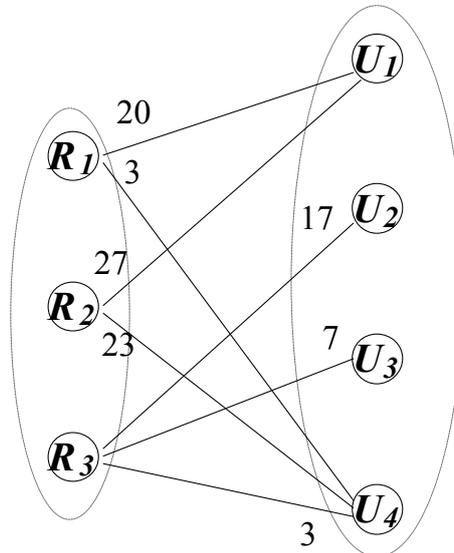


Figure 13.2 : Graphe valué biparti associé à la table c) de la figure 13.1

La démarche de Vinet pour trouver les pixels bien classés consiste à faire un couplage itératif dans ce graphe en choisissant à chaque fois l'arc de poids le plus élevé. Une modification de cette stratégie de couplage consiste à choisir le couplage de poids total maximum. La distance entre deux images de régions est constituée par le total de pixels mal classés dans une image par rapport à l'autre et, avec cette nouvelle stratégie, on obtient une distance au sens mathématique du terme. Pour déterminer un couplage maximal, on peut, par exemple, utiliser l'algorithme de Kuhn [KUHN55] ou celui d'Edmonds [EDMO65]. Dans le cas général, il n'y a pas unicité de la solution.

Si on applique cette distance à l'exemple de la figure 13.1, les couples déterminés sont : (R_1, U_1) , (R_2, U_4) et (R_3, U_2) , la distance entre R et U est 0.40, le recouvrement est de 60 pixels et est ici maximal.

1.2 Utilisation du contraste inter-région

Lorsque la segmentation idéale n'est pas connue, la technique précédente ne peut être appliquée. Le contraste, tel qu'il a été introduit dans le chapitre II (§ 7) [ZEBO88], [JOUR89], peut être utilisé pour des images faiblement texturées. Un contraste proche de 1 correspond à une image très bien segmentée.

2 Comparaison sur des images de synthèse

2.1 Fabrication des images de synthèse

Afin d'évaluer les performances des opérateurs présentés, des images de test de dimension 256×256 ont été fabriquées à partir d'une image originale comportant sept régions de formes géométriques différentes (*cf.* figure 13.3).

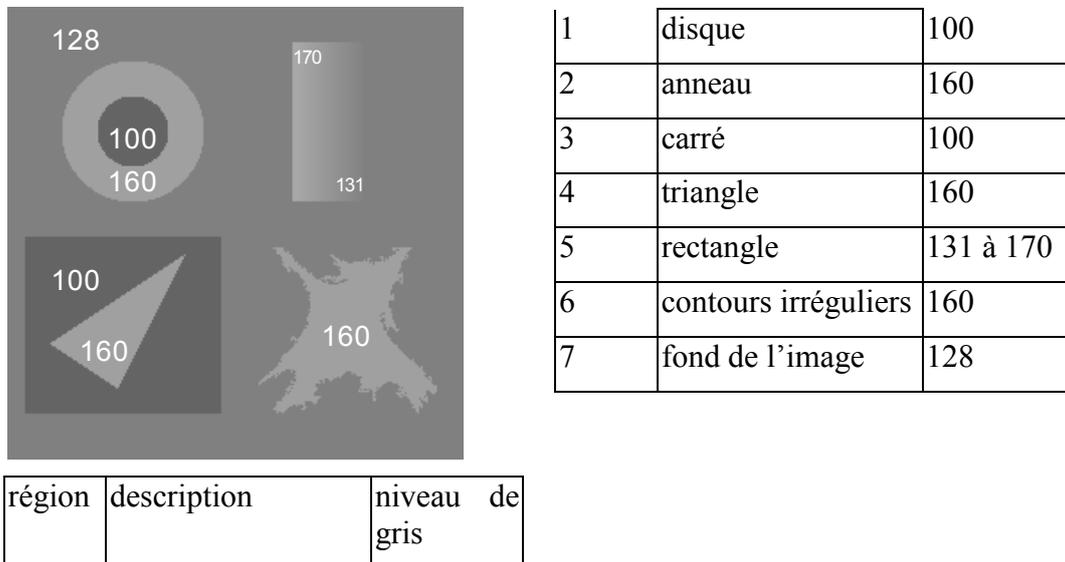


Figure 13.3 : Image originale SAVOISE avec le niveau de gris des régions

Une première série d'images de test a été réalisée en superposant à l'image originale des bruits blancs de différentes distributions : gaussienne, exponentielle, uniforme et triangulaire, de moyenne nulle et d'écart-type dix. Ce qui donne quatre images de bruit : BRUIT1, BRUIT2, BRUIT3 et BRUIT4. L'image BRUIT1 est présentée figure 11.5. Les autres images lui sont visuellement très proches, c'est pourquoi elles ne sont pas présentées.

Le profil des niveaux de gris de la ligne 64 est représenté en figure 13.4 pour ces différentes images.

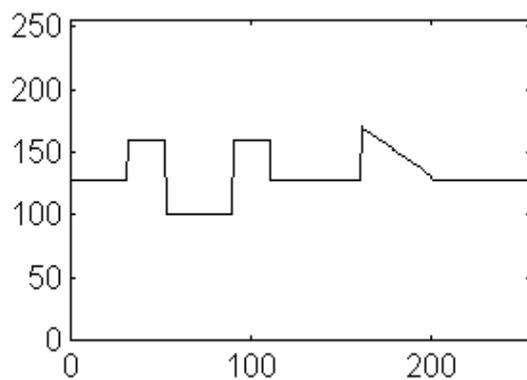
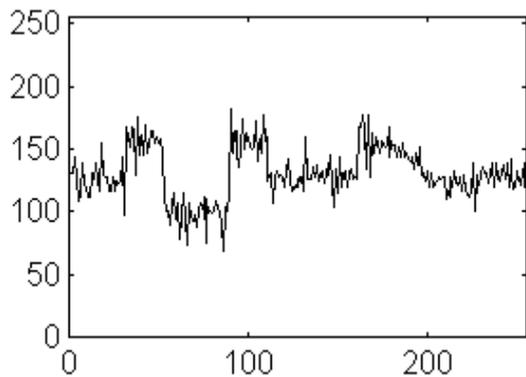
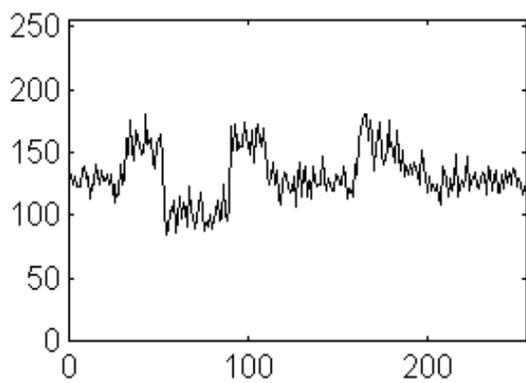


image non bruitée



BRUIT1 (gaussien)

BRUIT2 (exponentiel)

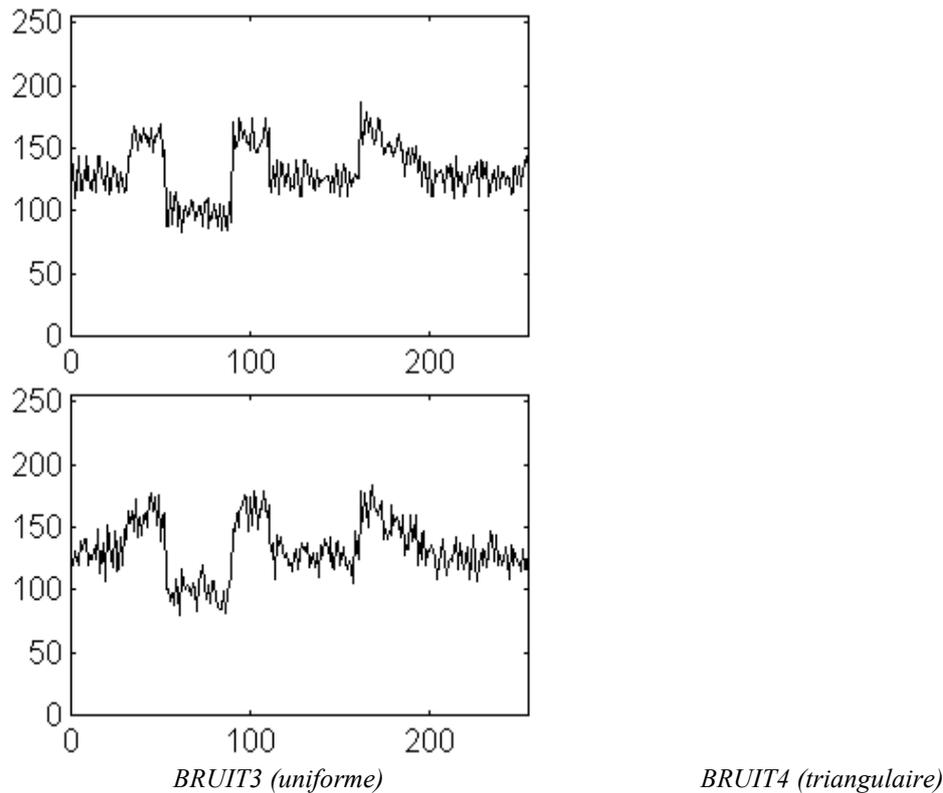


Figure 13.4 : Profil de ligne des images de test

Une deuxième série d'images a été synthétisée en remplissant les différentes régions de l'image SAVOISE avec des textures issues de l'album de Brodatz [BROD66].

Les textures utilisées sont les suivantes :

- HERBE [Brodatz D9]
- LAINÉ [Brodatz D19]
- LIERRE [Brodatz D24]
- SABLE [Brodatz D29]
- EAU [Brodatz D38]
- BOIS [Brodatz D68]
- BULLES [Brodatz D112].

2.2 Méthodes testées

Six méthodes ont été testées sur les images de bruit :

- la régularisation markovienne (*cf.* chapitre XI § 1.3)
- la relaxation non supervisée, approche markovienne (*cf.* chapitre XI § 2.3)
- la relaxation supervisée, approche markovienne (*cf.* chapitre XI § 2.5)

- la corrélation d'histogrammes, par propagation (*cf.* chapitre XII § 3)
- les pyramides adaptatives (*cf.* chapitre XII § 4.6)
- l'analyse multidimensionnelle (*cf.* chapitre X § 2).

Pour les images de textures seule la classification multidimensionnelle est présentée, les autres méthodes n'étant pas adaptées à ce type d'image.

2.3 Paramètres de réglage des méthodes

Régularisation markovienne

L'apprentissage a été réalisé sur des fenêtres carrées sélectionnées interactivement, l'attribut de texture utilisé est la moyenne des niveaux de gris dans des fenêtres de taille 3x3. β est nul à la première itération puis prend la valeur 1000.

Le fait de ne considérer que la moyenne comme attribut de texture implique que l'on ne recherche que des régions de niveau de gris moyen constant. Ceci n'est pas, *a priori*, adapté à la zone rectangulaire présentant un dégradé uniforme de niveaux de gris.

Relaxation non supervisée, approche markovienne

Le seul paramètre utilisé pour cet algorithme, σ , représente approximativement l'écart-type moyen accepté à l'intérieur d'une région. On a, pour chaque image de bruit, utilisé la même séquence croissante de valeurs de σ : de 10 à 17.

Le modèle contient explicitement un terme modélisant le comportement des variations de niveaux de gris à l'intérieur d'une région : ils suivent une loi gaussienne de variance σ^2 dont la moyenne dépend de la région. Par conséquent, de nouveau, la zone rectangulaire présentant un dégradé uniforme de niveaux de gris n'est pas ce que l'on pourrait appeler une région homogène pour cet algorithme et ne peut pas, *a priori*, être segmentée correctement. Les résultats obtenus, tout de même cohérents, s'expliquent par le fait que l'algorithme est assez souple et que le terme contenant un gradient implicite dans l'énergie globale, facilite la création de cette région.

Relaxation supervisée, approche markovienne

La fenêtre d'apprentissage est un carré de côté 20 pixels, le nombre de classes est de 4.

Corrélation d'histogrammes

La croissance se fait par propagation, la fenêtre de corrélation est de taille 3x3, l'écart-type de la gaussienne est de 1 et le seuil de corrélation T_0 vaut 0,3.

Pyramides adaptatives

L'image est transformée en un graphe de 8-connexité, sur la base de laquelle la pyramide adaptative est construite. Le seuil est le même pour toutes les images et a été fixé à 4.

Analyse multidimensionnelle

L'analyse multidimensionnelle a été appliquée à travers le programme Arthur [LECH91]. Ce programme effectue une classification et une segmentation supervisée des images texturées par la méthode des paires (*cf.* chapitre X § 2.2.2.2). Environ 500 attributs de divers types sont calculés sur la base d'apprentissage (pour une description de ces attributs, voir le chapitre II) :

- paramètres statistiques
- paramètres issus des matrices de cooccurrence dans différentes directions
- longueurs de plage
- paramètres spectraux
- attributs fractals : lacunarité, dimension fractale généralisée, attributs multifractals
- *etc.*

Pour les images de bruit les paramètres retenus sont les moments centrés d'ordre 0 et 3, la lacunarité et le moment invariant numéro 2 sur une fenêtre de taille 4x4.

Pour les images de texture les paramètres utilisés sont la lacunarité, le contraste et l'entropie calculés à partir des matrices de cooccurrence, le coefficient α du modèle AR (modèle auto-régressif), la longueur de plage RF2 non sommée sur θ en vertical. L'apprentissage a été effectué sur une fenêtre de taille 24x24 ; il nécessite une heure de calculs sur station SPARC2 pour chaque essai, plusieurs essais ont été effectués avec différentes tailles de fenêtres d'apprentissage. La classification dure de 2 à 3 minutes.

2.4 Analyse des résultats

Les résultats des différentes méthodes testées sont des images d'étiquettes ou de régions. Pour une meilleure lisibilité, les planches de résultats (*cf.* figures 13.5 à 13.8), ne présentent que les contours des régions, arbitrairement placés dans l'une ou l'autre des régions. Les contours épais indiquent donc la présence de toutes petites régions.

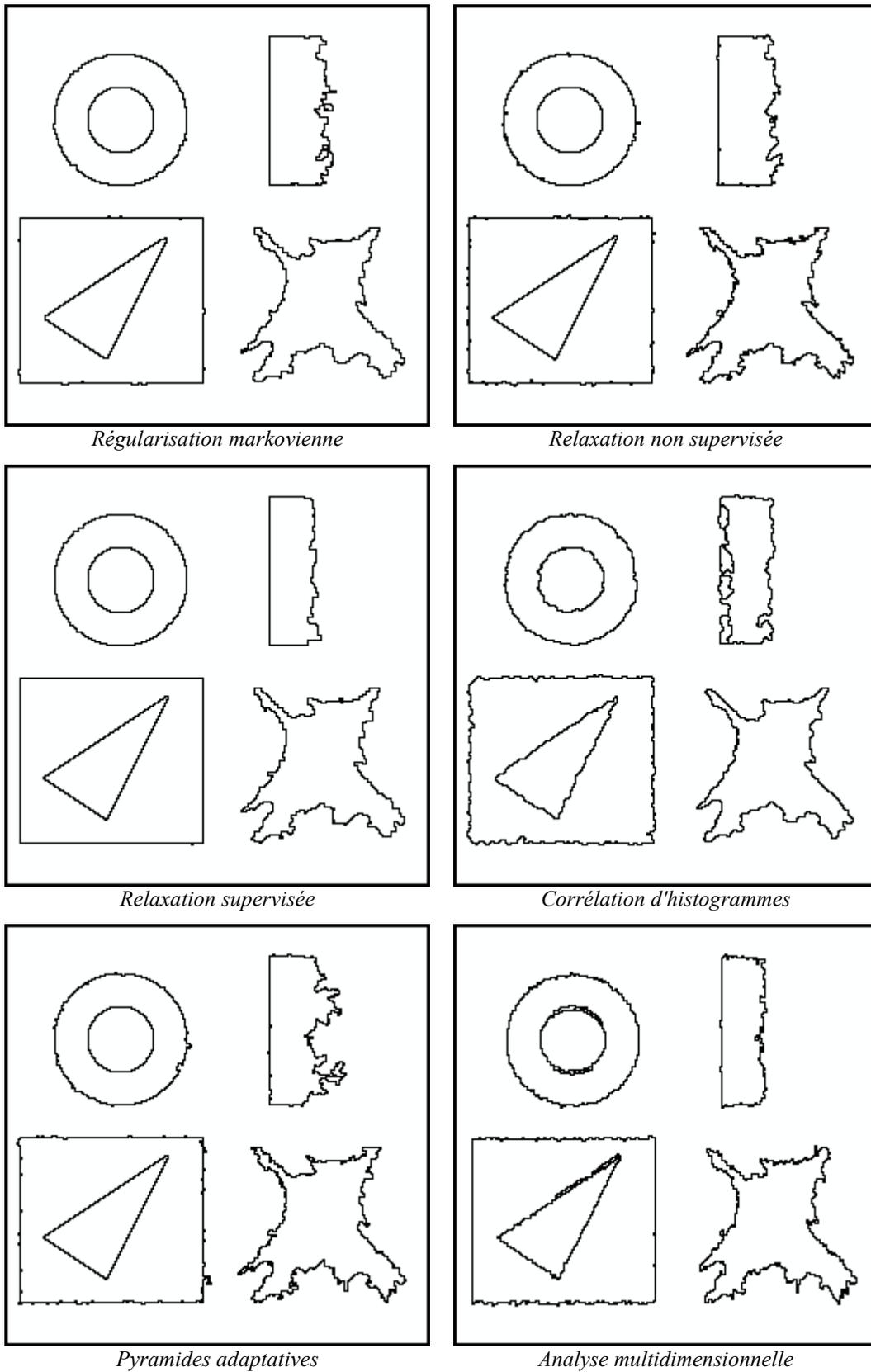


Figure 13.5 : Image BRUITI

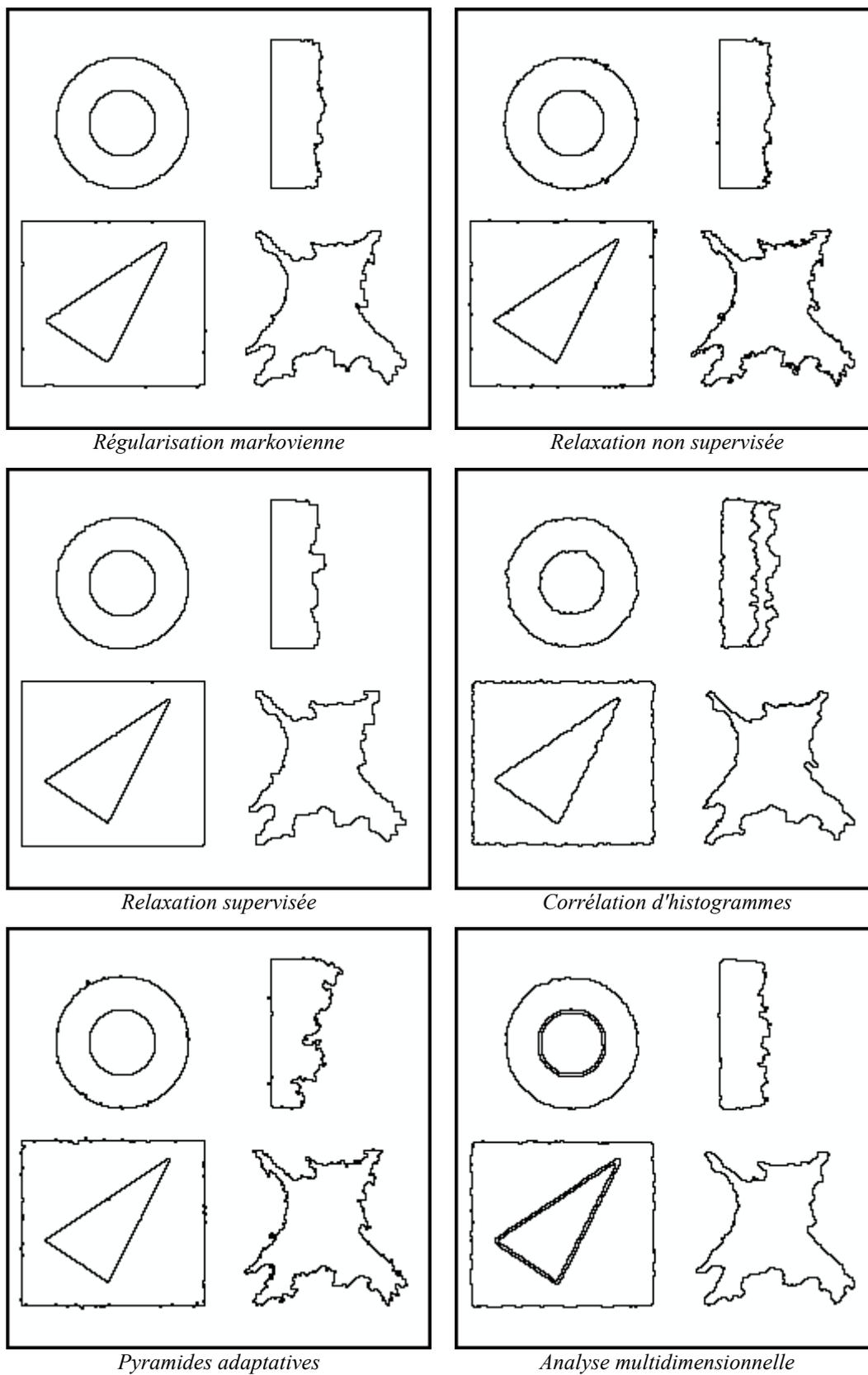


Figure 13.6 : Image BRUIT2

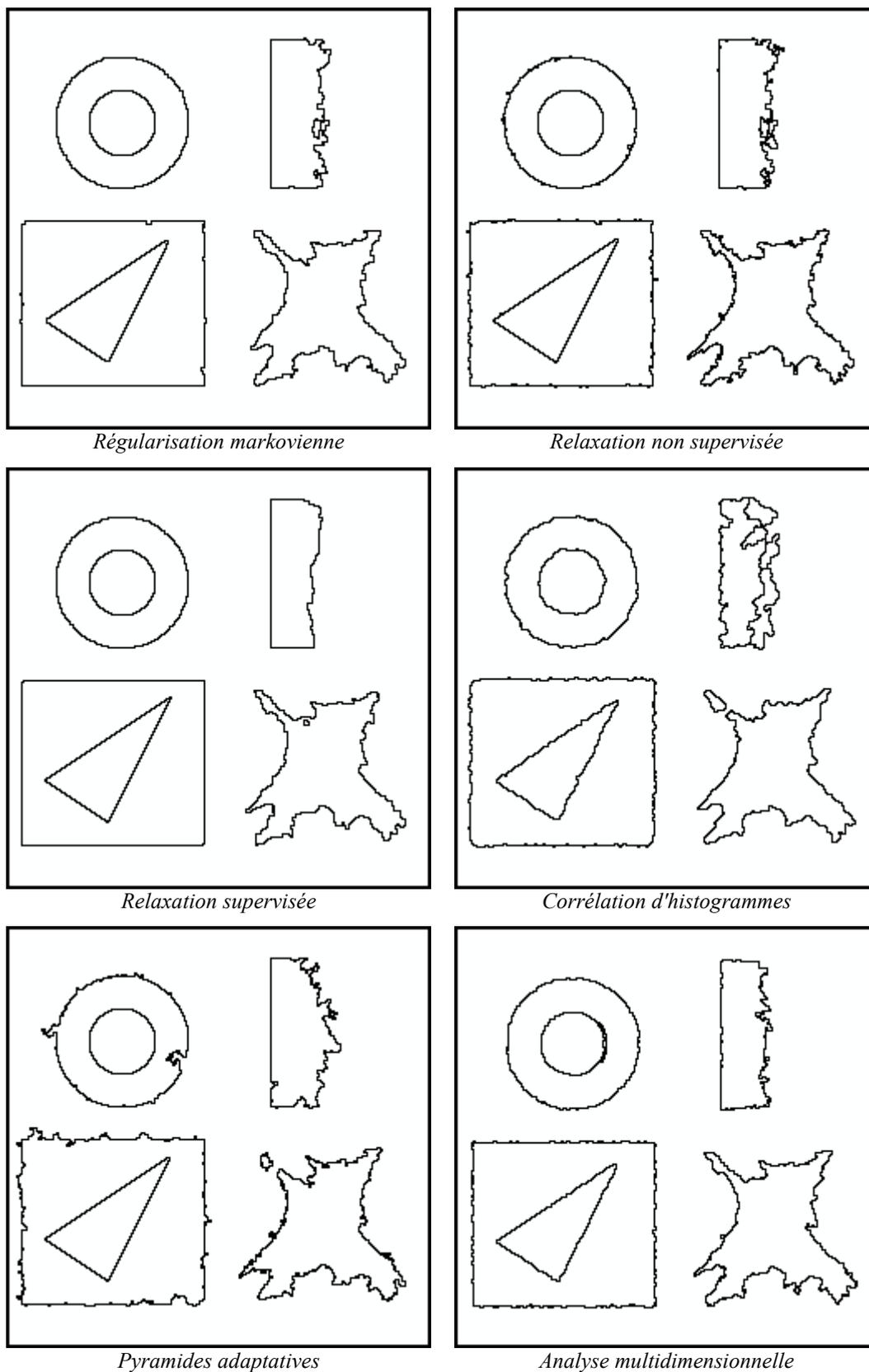


Figure 13.7 : Image BRUIT3

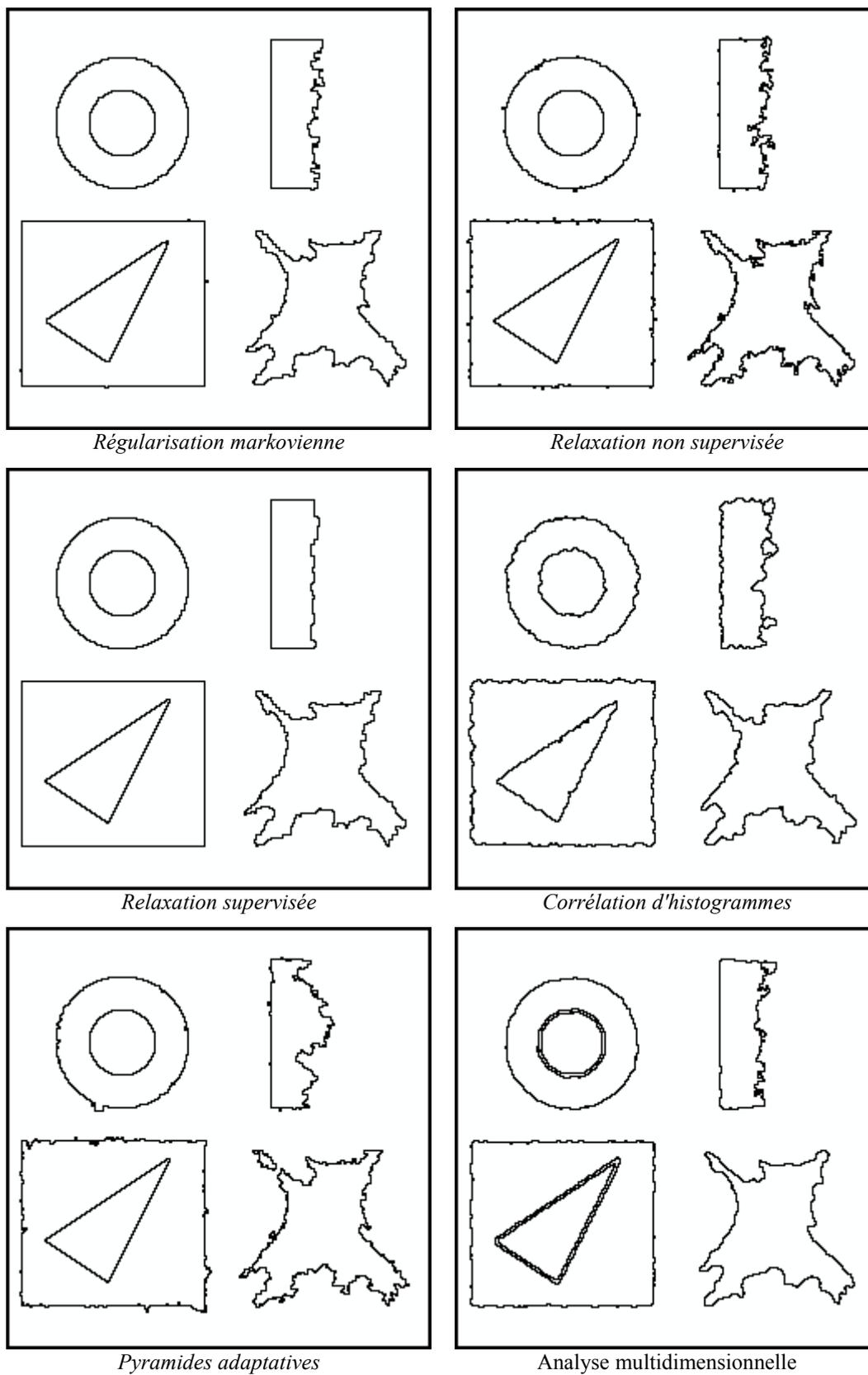


Figure 13.8 : Image BRUIT4

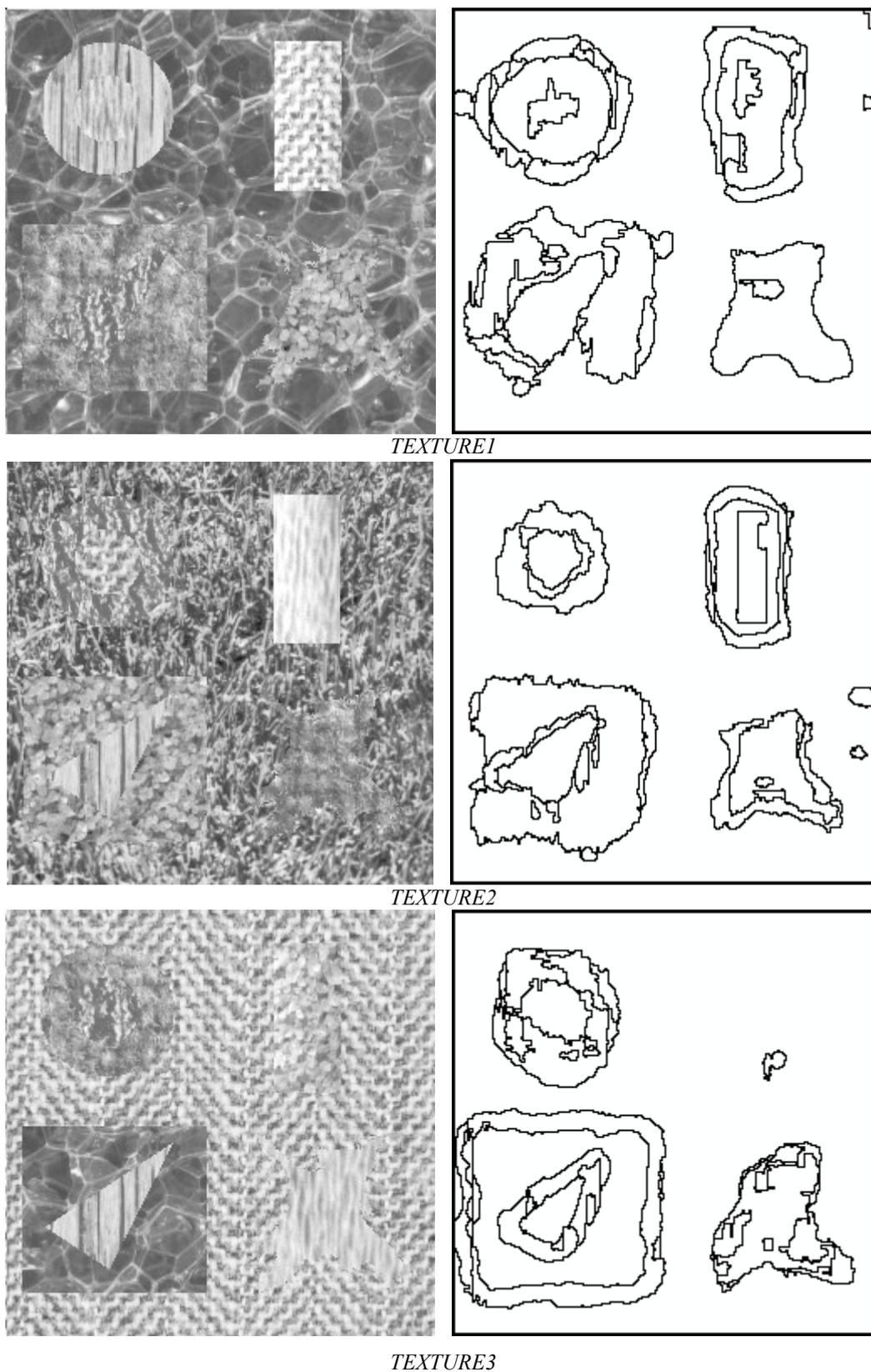


Figure 13.9 : Analyse multidimensionnelle sur images de texture

2.4.1 Images bruitées (analyse quantitative)

Les tableaux 13.1 à 13.4 contiennent les distances calculées par la méthode de Vinet modifiée. On rappelle que cette distance est calculée entre deux images d'étiquettes, la première image est l'image originale SAVOISE (*cf.* figure 13.3), la seconde est le résultat obtenu par l'une des différentes méthodes testées. Les résultats ont été normalisés par le nombre de pixels considérés puis multipliés par 100 pour avoir des tableaux plus lisibles. On rappelle qu'une valeur proche de 0 correspond à une image bien segmentée.

Le dernier tableau donne le contraste associé aux images de régions, ici une valeur proche de 1 correspond à une bonne segmentation.

Distance avec l'image de référence

On remarque que la relaxation supervisée, approche markovienne, détecte le mieux la couronne (haut gauche) et le triangle dans le carré (bas gauche). La couronne est moins bien détectée par l'analyse multidimensionnelle. Les régions les moins bien segmentées se trouvent dans la partie droite des images (dans les colonnes correspondantes des tableaux, on a les valeurs les plus élevées) ; la moitié droite des images est plus difficile à segmenter que la moitié gauche. La régularisation markovienne segmente le mieux la partie supérieure droite des images.

Pour les images BRUIT1, BRUIT2 et BRUIT4 la relaxation non supervisée, approche markovienne, donne le meilleur résultat global ; on retrouve par cette mesure les remarques visuelles : bonne détection de la région irrégulière, bonne surface pour le rectangle. Pour l'ensemble des images de bruit, les approches markoviennes et les pyramides adaptatives conduisent aux meilleurs résultats.

La corrélation d'histogrammes et l'analyse multidimensionnelle donnent les résultats les moins bons ; ces techniques ne semblent pas convenir pour des images bruitées. L'analyse multidimensionnelle est plus adaptée à la segmentation d'images texturées ; quant à la corrélation d'histogrammes, une phase de prétraitement semble nécessaire pour pouvoir l'appliquer à des images fortement bruitées.

Pour les images de texture (TEXTURE1 à TEXTURE3) aucun calcul de distance n'a été effectué car une seule technique de segmentation est présentée.

IMAGE BRUIT1	haut gauche	haut droit	bas gauche	bas droit	global
Régularisation markovienne	0.21	4.20	0.20	1.55	1.53
Relaxation non supervisée	0.20	4.44	0.41	0.85	1.47
Relaxation supervisée	0.09	8.16	0.01	1.14	2.33
Corrélation d'histogrammes	1.72	9.26	2.45	3.10	4.12
Pyramide adaptative	0.29	6.34	0.50	1.07	2.04
Analyse multidimensionnelle	3.82	9.43	6.10	4.94	6.06

Tableau 13.1 : Distances entre l'image originale SAVOISE et l'image BRUIT1 segmentée

IMAGE BRUIT2	haut gauche	haut droit	bas gauche	bas droit	global
Régularisation markovienne	0.14	6.71	0.21	1.55	2.14
Relaxation non supervisée	0.23	6.75	0.35	0.85	2.03
Relaxation supervisée	0.10	7.72	0.03	1.53	2.33
Corrélation d'histogrammes	1.89	12.49	3.35	3.55	5.30
Pyramide adaptative	0.18	7.05	0.44	0.78	2.10
Analyse multidimensionnelle	2.42	8.32	4.19	2.72	4.40

Tableau 13.2 : Distances entre l'image originale SAVOISE et l'image BRUIT2 segmentée

IMAGE BRUIT3	haut gauche	haut droit	bas gauche	bas droit	global
Régularisation markovienne	0.15	5.76	0.23	1.63	1.93
Relaxation non supervisée	0.27	5.94	0.67	0.98	1.95
Relaxation supervisée	0.12	8.31	0.02	1.48	2.47
Corrélation d'histogrammes	1.44	12.19	2.56	3.70	4.95
Pyramide adaptative	1.34	5.99	1.67	2.28	2.73
Analyse multidimensionnelle	3.51	8.85	6.38	4.63	5.82

Tableau 13.3 : Distances entre l'image originale SAVOISE et l'image BRUIT3 segmentée

IMAGE BRUIT4	haut gauche	haut droit	bas gauche	bas droit	global
Régularisation markovienne	0.16	7.89	0.09	1.55	2.41
Relaxation non supervisée	0.23	7.98	0.95	0.75	2.34
Relaxation supervisée	0.09	8.61	0.00	1.45	2.52
Corrélation d'histogrammes	1.73	9.12	2.77	3.29	4.21
Pyramide adaptative	0.34	9.12	1.03	2.02	3.11
Analyse multidimensionnelle	2.51	9.51	3.95	2.63	4.63

Tableau 13.4 : Distances entre l'image originale SAVOISE et l'image BRUIT4 segmentée

Contraste

Les images de test BRUIT1 à BRUIT4 sont des images non texturées, aussi la mesure du contraste associé à l'image des étiquettes (*cf.* chapitre II § 7) a un sens. Cette mesure globale montre que, pour les images de bruit, la meilleure segmentation est obtenue par la relaxation supervisée qui donne un contraste très proche de 1. Pour ce type d'images, l'analyse multidimensionnelle fournit les moins bons résultats parmi les méthodes testées ; il faut bien noter que cette méthode est plus adaptée aux images texturées, comme on le verra plus loin (*cf.* § 2.6.2).

Contraste	BRUIT1	BRUIT2	BRUIT3	BRUIT4
Régularisation markovienne	0.963	0.963	0.963	0.967
Relaxation non supervisée	0.991	0.991	0.989	0.990
Relaxation supervisée	0.993	0.991	0.994	0.993
Corrélation d'histogrammes	0.954	0.955	0.940	0.951
Pyramide adaptative	0.988	0.991	0.984	0.984
Analyse multidimensionnelle	0.858	0.945	0.760	0.949

Tableau 13.5 : Contraste associé aux images d'étiquettes obtenues après segmentation

2.4.2 Images bruitées (analyse visuelle)

Les *méthodes markoviennes* donnent de très bons résultats sur les quatre images de bruit : très bonne segmentation des formes géométriques simples (disques, carré et triangle). On peut noter la non détection de la pointe de l'angle aigu du triangle, sauf pour la relaxation supervisée qui le détecte sur BRUIT3. La région aux contours irréguliers est bien reconnue mais les contours obtenus sont lisses sauf pour la relaxation non supervisée qui donne les meilleurs résultats. Le rectangle présentant un dégradé uniforme de niveaux de gris est correctement décelé sauf pour la transition la plus faible pour laquelle aucune méthode ne fournit de résultat satisfaisant. La régularisation et la relaxation non supervisée déterminent le mieux ce rectangle (la surface détectée est proche de la surface réelle). Le nombre de régions déterminé est correct sauf pour la relaxation supervisée qui détermine une région supplémentaire pour l'image BRUIT3.

La *corrélacion d'histogrammes* détecte correctement les différentes régions. Les frontières des formes géométriques sont hachés, celles de la région irrégulière sont lissées. Le rectangle est découpé en de nombreuses régions.

La méthode des *pyramides adaptatives* donne des régions aux contours légèrement bruités. Des aspérités apparaissent sur les frontières des formes géométriques simples. Le triangle est très bien détecté sur les quatre images à l'exception de la pointe de l'angle aigu. Le nombre de régions détectées est correct, on note une coupure de la région aux contours irréguliers pour BRUIT3 et BRUIT4.

L'analyse *multidimensionnelle* détecte mal les transitions entre deux régions, la région aux contours irréguliers voit ses frontières lissées.

2.4.3 Images de texture

Pour ces images, ce n'est plus l'intensité moyenne qui caractérise les régions mais des attributs de texture. Il est très difficile, même pour un observateur humain, de reconnaître et de localiser précisément les régions. Les méthodes testées dans ce chapitre n'ont pas fourni de résultats exploitables sur ces images, soit parce qu'elles ne sont pas adaptées aux images texturées, soit parce qu'elles ont échoué en raison de la faible taille des régions par rapport à la résolution des textures. Seule la méthode de classification multidimensionnelle a permis la détection de la plupart des régions, même si leurs frontières ne sont pas localisées avec précision.

Cette situation, bien qu'extrême (il est rare de voir réunies autant de difficultés dans une seule image !), confirme la nécessité et l'intérêt du développement de méthodes numériques de segmentation.

3 Comparaison sur des images réelles

Les six images utilisées sont décrites au chapitre IX § 3.1.

3.1 Méthodes testées

Sept méthodes ont été testées :

- la fermeture de contours, après détection de contours par l'opérateur Deriche dérivée première, (*cf.* chapitre VI § 1)
- la relaxation supervisée, approche markovienne (chapitre XI § 2.5)
- la relaxation non supervisée, approche markovienne (chapitre XI § 2.3)
- la corrélation d'histogrammes (chapitre XII § 3)
- les pyramides adaptatives (chapitre XII § 4.6)
- la méthode de Nakagawa (chapitre X § 1.3.1)
- la méthode de Fisher (chapitre X § 1.1.2.1).

3.2 Paramètres des méthodes

Les régions monopixel ont été fusionnées avec l'une des régions les plus proches géométriquement.

Fermeture de contours

Les valeurs utilisées pour Deriche dérivée première sont présentées chapitre IX, tableau 9.12. La fermeture de contours a été réalisée avec une longueur de 7 et une profondeur de 3 sauf pour AQUITAINE où la longueur utilisée est 3 (chapitre VI § 1.2.2). Les contours pendants (qui n'ont pas rejoint un autre contour pendant la phase de fermeture) ont été éliminés.

Relaxation supervisée, approche markovienne

Le nombre de classes est 3 sauf pour COULOIR où on a utilisé 4 classes. La convergence de l'échantillonneur de Gibbs a été atteinte après 100 à 250 itérations suivant les images.

Relaxation non supervisée, approche markovienne

	ANGIOGRAPHI E	AQUITAINE	BAGUES	COULOIR	IRM	MUSCLE
σ croît de	7 à 16	7 à 17	7 à 17	7 à 10	7 à 9	5 à 25

Le modèle de régions utilisés est celui à niveaux de gris constants.

Corrélation d'histogrammes

Les histogrammes sont calculés sur des fenêtres carrées 5x5 pour AQUITAINE, BAGUES et MUSCLE ; sur des fenêtres carrées 9x9 pour les autres images. La gaussienne utilisée dans la convolution des histogrammes a un écart-type de 1.0. Le seuil de corrélation T_0 est donné dans le tableau suivant :

	ANGIOGRAPHI E	AQUITAINE	BAGUES	COULOIR	IRM	MUSCLE
seuil T_0	0.35	0.35	0.30	0.35	0.40	0.35

Pyramides adaptatives

Pour le processus bas-haut deux seuils sont utilisés : thd1 (valeur locale relative) et thd2 (valeur globale). Ce sont les seuils de conservation des arcs du graphe d'adjacence de départ. Pour le processus haut-bas, on a besoin des 3 coefficients du filtre des cellules (ce filtre est utilisé lors de la segmentation qui est un processus haut-bas de propagation de l'information du sommet vers la base). Un dernier paramètre employé est la valeur de seuillage des arcs (thd_value) ; plus ce seuil est élevé, moins on obtient de régions.

	ANGIOGRAPHIE	AQUITAINE	BAGUES	COULOIR	IRM	MUSCLE
thd1	1	2	2	2	1	1
thd2	2	4	4	4	2	2
coefficient	10	10	10	10	6	10
s	5	5	5	5	3	5
du	1	1	1	1	1	1
filtre						
thd_value	20	15	25	15	15	20

Méthode de Nakagawa

On a utilisé des fenêtres de taille 64x64.

Méthode de Fisher

	ANGIOGRAPHIE	AQUITAINE	BAGUES	COULOI R	IRM	MUSCLE
nombre de classes	5	4	9	9	5	5

3.3 Résultats sur les images de la banque du GDR 134

Image ANGIOGRAPHIE

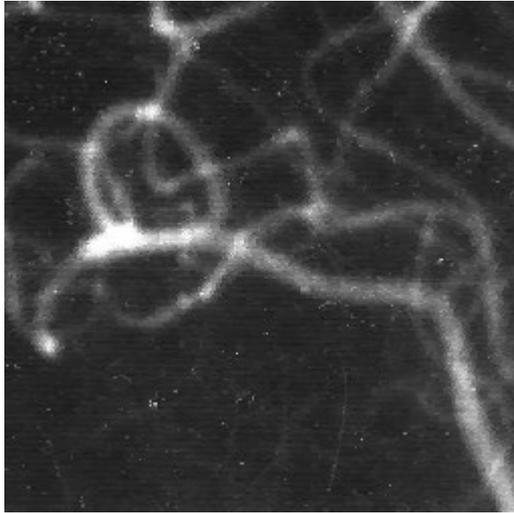
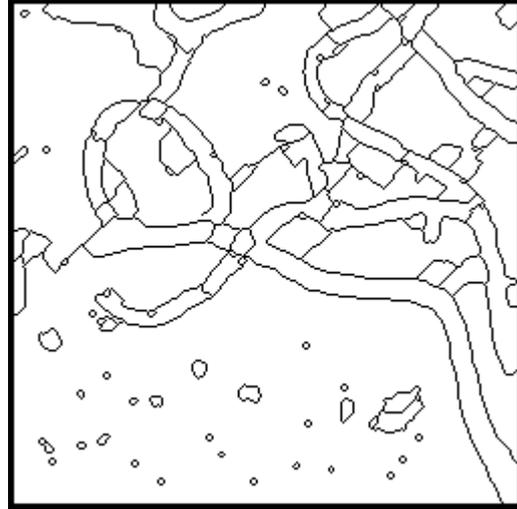
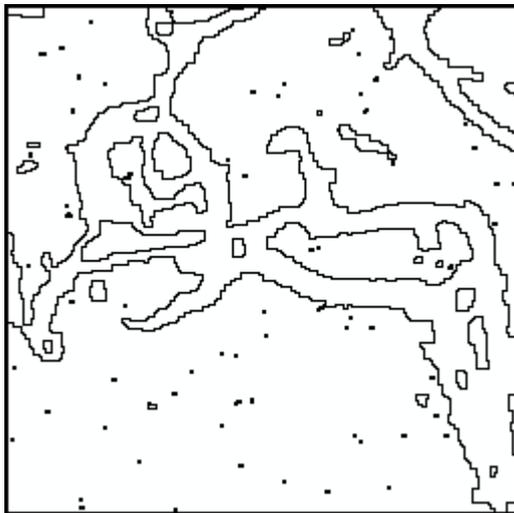


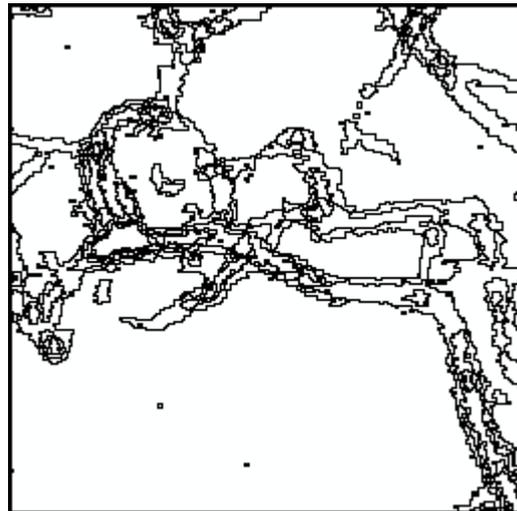
Image originale



Fermeture de contours

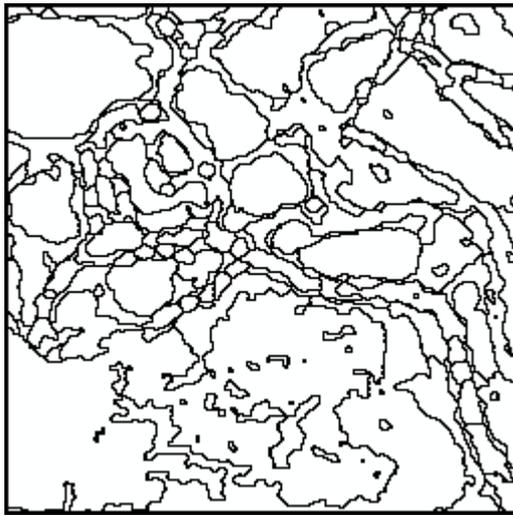


Relaxation supervisée

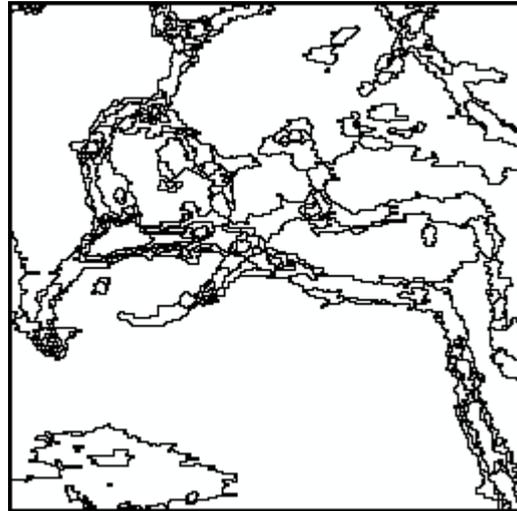


Relaxation non supervisée

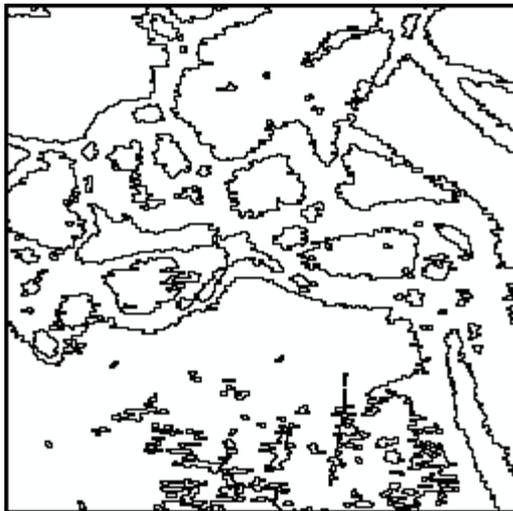
Figure 13.10 a : Résultats de segmentation par des méthodes de l'approche région sur l'image ANGIOGRAPHIE



Corrélation d'histogrammes



Pyramides adaptatives



Méthode de Nakagawa



Méthode de Fisher

Figure 13.10 b : Résultats de segmentation par des méthodes de l'approche région sur l'image ANGIOGRAPHIE (suite)

La *corrélation d'histogrammes* et la méthode de *Fisher* détectent beaucoup de bruit et conduisent à des images difficilement exploitables. La *relaxation non supervisée* et les *pyramides adaptatives* produisent beaucoup de petites régions dans la zone des contours des vaisseaux. La localisation des contours est en général très imprécise, sauf pour la *fermeture de contours* et la *relaxation supervisée* qui donnent des images très propres, la première ayant par contre l'inconvénient de couper les vaisseaux par des faux contours.

ANGIOGRAPHIE est une image très bruitée qui nécessite différents pré-traitements pour conduire à de bons résultats. Les méthodes sont assez sensibles

au bruit impulsionnel sauf la *relaxation non supervisée* et les *pyramides adaptatives*.

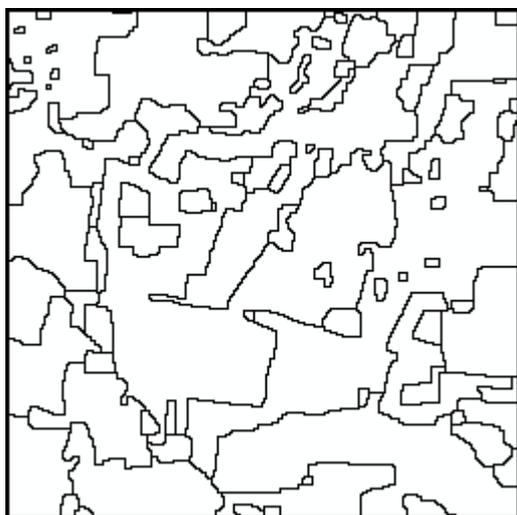
Image AQUITAINE



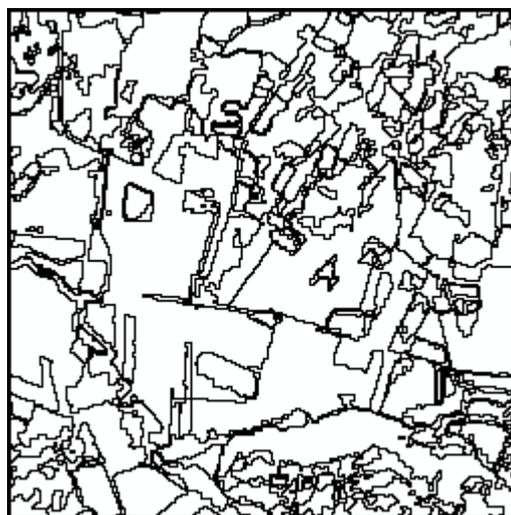
Image originale (rehaussée)



Fermeture de contour



Relaxation supervisée



Relaxation non supervisée

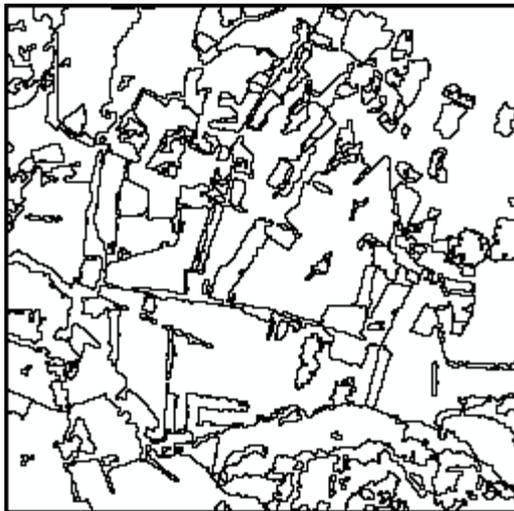
Figure 13.11 a : Résultats de segmentation par des méthodes de l'approche région sur l'image AQUITAINE



Corrélation d'histogrammes



Pyramides adaptatives



Méthode de Nakagawa



Méthode de Fisher

Figure 13.11 b : Résultats de segmentation par des méthodes de l'approche région sur l'image AQUITAINE (suite)

Les différentes méthodes conduisent à des images segmentées contenant plus ou moins de régions. On note la présence fréquente de toutes petites régions. La *relaxation supervisée* conduit à une sous segmentation. Dans la *relaxation non supervisée*, les *pyramides adaptatives*, *Nakagawa* et *Fisher*, des régions allongées correspondent aux voies de communication (fleuve et route).

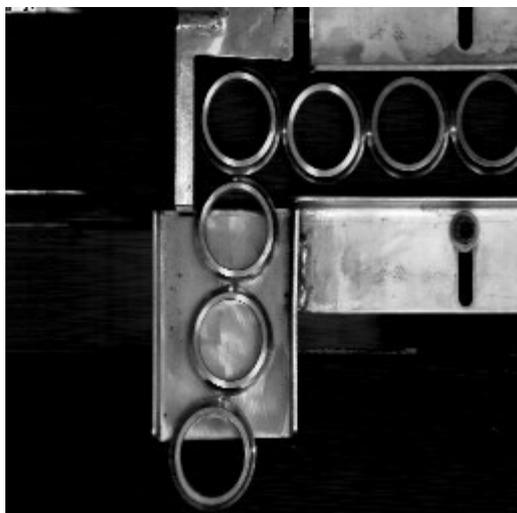
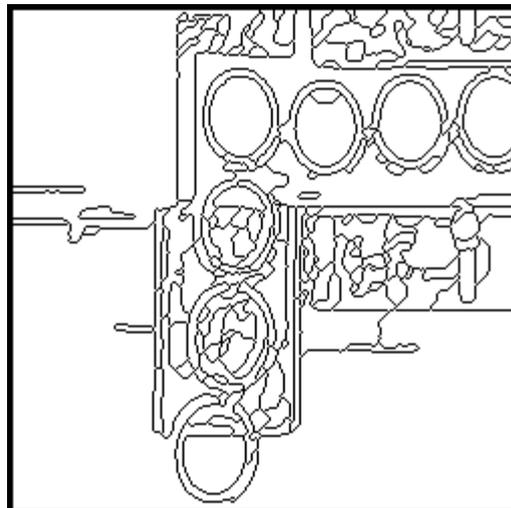
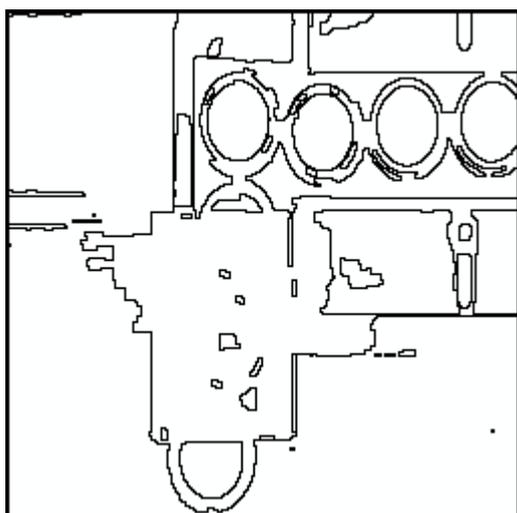
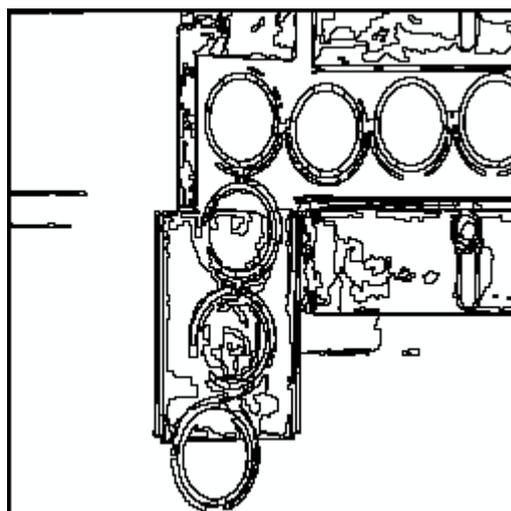
Image BAGUES*Image originale**Fermeture de contours**Relaxation supervisée**Relaxation non supervisée*

Figure 13.12 a : Résultats de segmentation par des méthodes de l'approche région sur l'image BAGUES

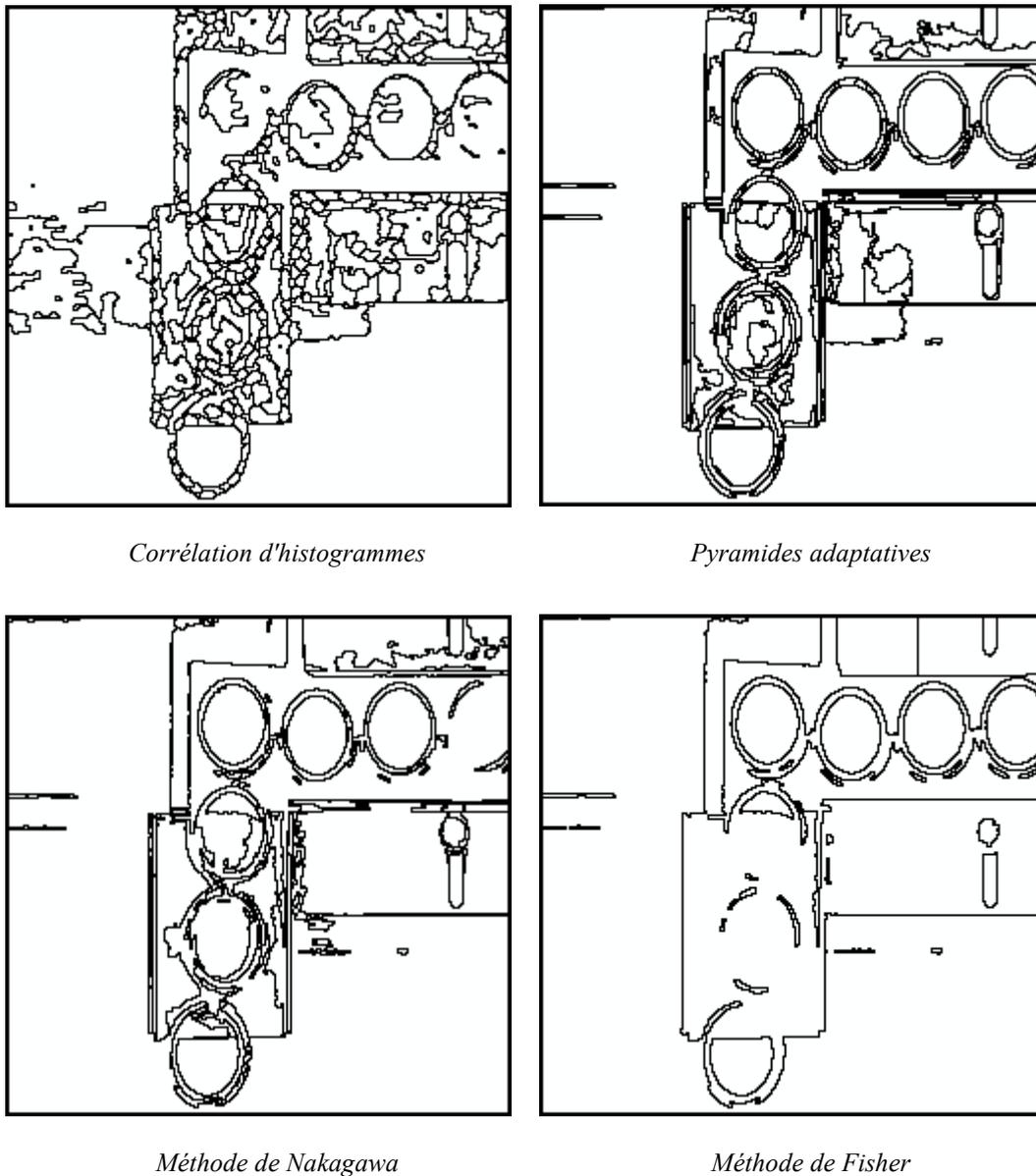


Figure 13.12 b : Résultats de segmentation par des méthodes de l'approche région sur l'image BAGUES (suite)

Les bagues sont bien détectées par *Fisher*, la *relaxation non supervisée* et par la *fermeture de contours*. La *relaxation supervisée* n'a pas détecté les bagues du bas. La *relaxation non supervisée*, la *fermeture de contours* et la *corrélacion d'histogrammes* déterminent beaucoup de petites régions liées aux reflets sur les parties métalliques (ainsi qu'au bruit dans le fond pour la *corrélacion d'histogrammes*). *Nakagawa* donne un bon résultat sauf pour la bague située en haut, à droite.

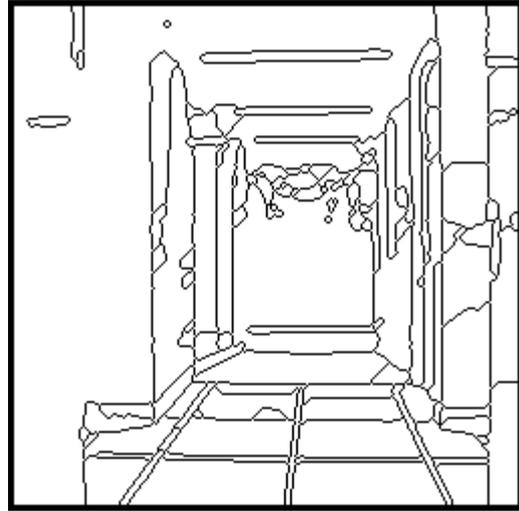
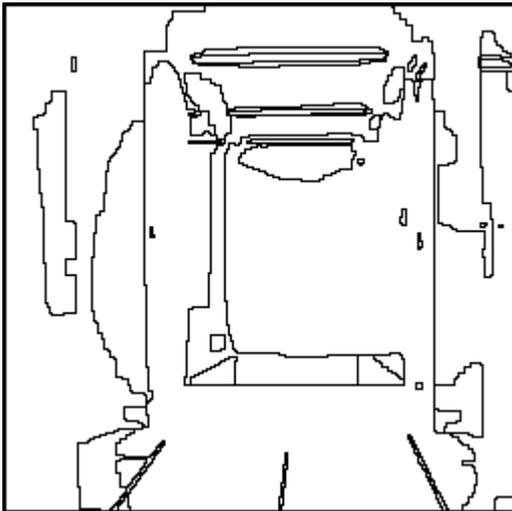
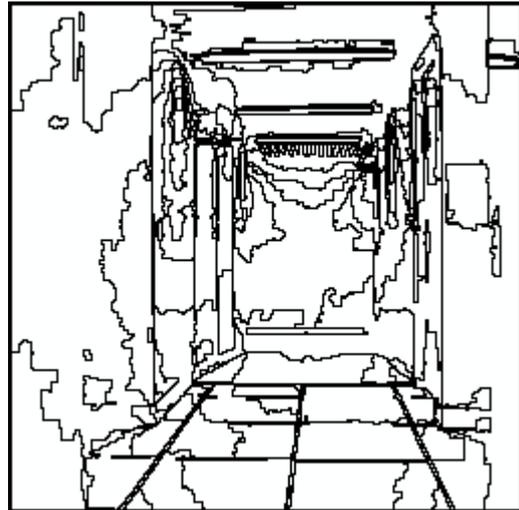
Image COULOIR*Image originale**Fermeture de contours**Relaxation supervisée**Relaxation non supervisée*

Figure 13.13 a : Résultats de segmentation par des méthodes de l'approche région sur l'image COULOIR

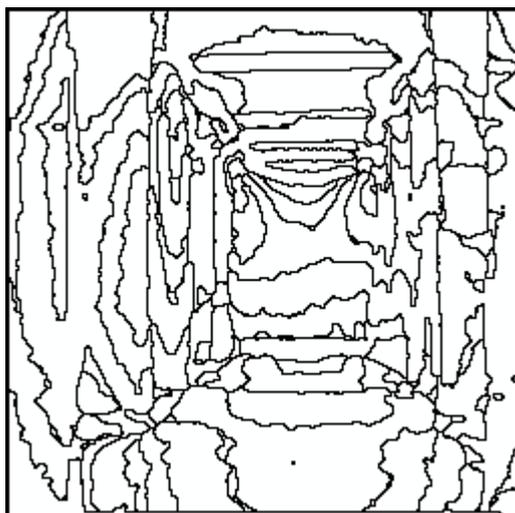
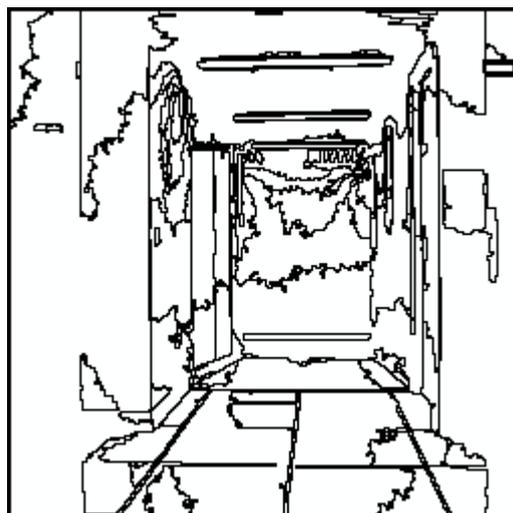
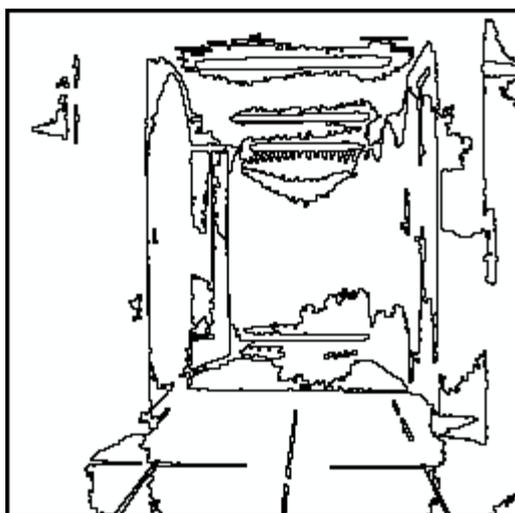
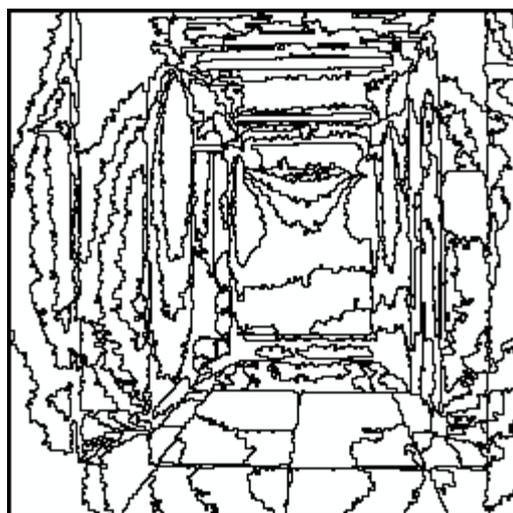
*Corrélation d'histogrammes**Pyramides adaptatives**Méthode de Nakagawa**Méthode de Fisher*

Figure 13.13 b : Résultats de segmentation par des méthodes de l'approche région sur l'image COULOIR (suite)

La *corrélation d'histogramme* et *Fisher* sont très gênés par les variations d'éclaircement. La *relaxation supervisée* et *Nakagawa* détectent peu de régions. La *fermeture de contours* donne un assez bon résultat.

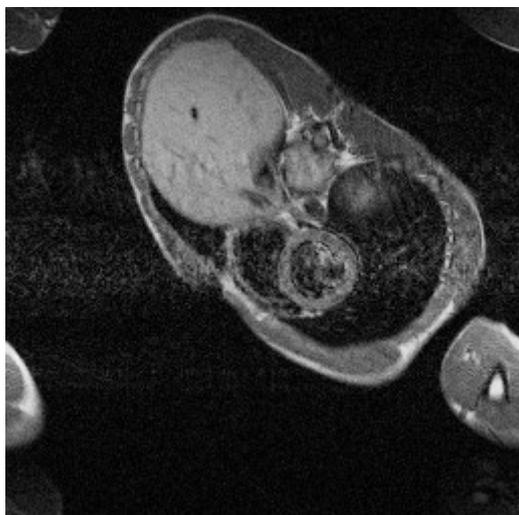
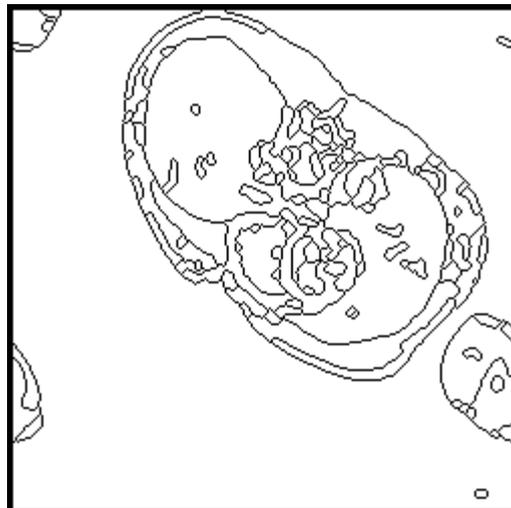
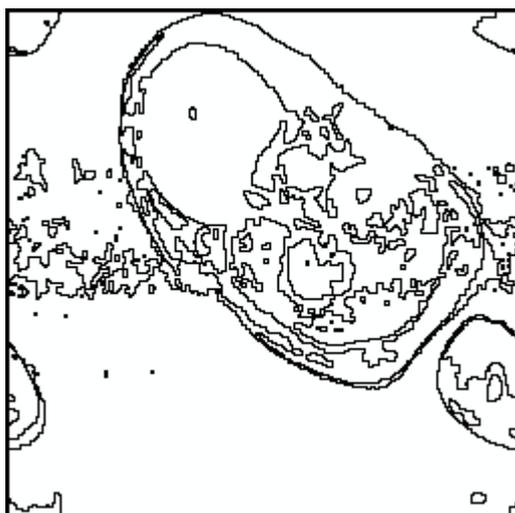
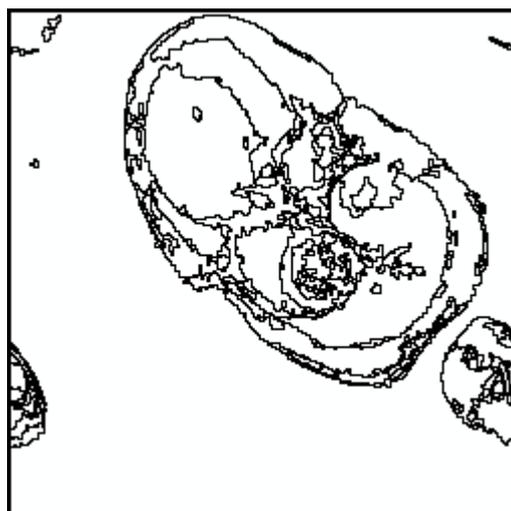
Image IRM*Image originale (rehaussée)**Fermeture de contours**Relaxation supervisée**Relaxation non supervisée*

Figure 13.14 a : Résultats de segmentation par des méthodes de l'approche région sur l'image IRM

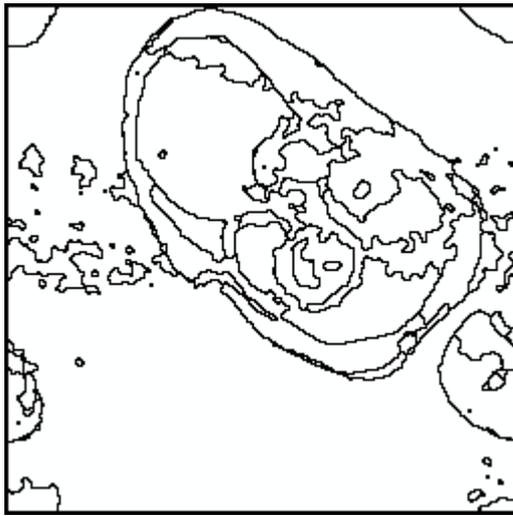
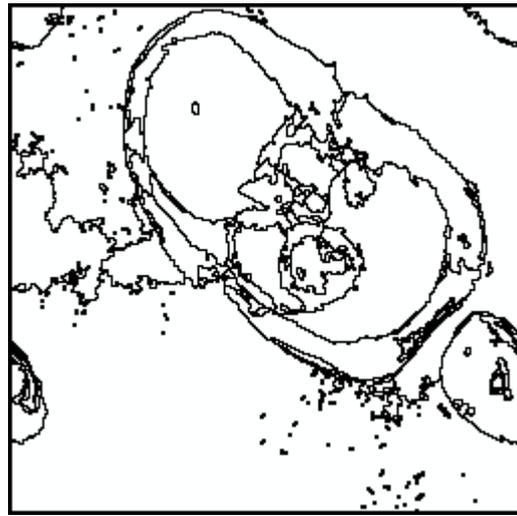
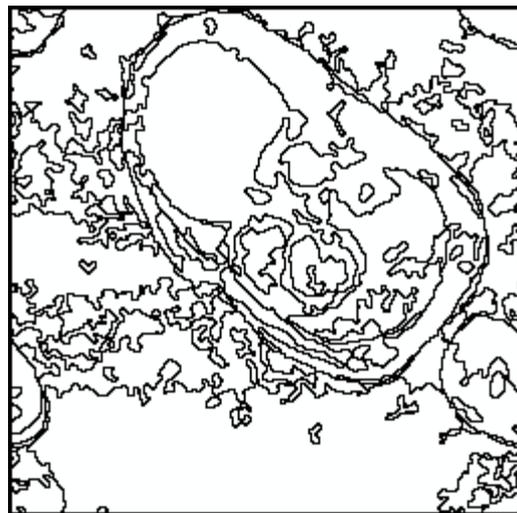
*Corrélation d'histogrammes**Pyramides adaptatives**Méthode de Nakagawa**Méthode de Fisher*

Figure 13.14 b : Résultats de segmentation par des méthodes de l'approche région sur l'image IRM (suite)

Fisher détermine trop de régions. Les autres méthodes donnent des résultats corrects dans l'ensemble nécessitant, toutefois, pour une éventuelle application, des phases de post-traitement. Les zones non homogènes à l'intérieur des organes gênent très peu *Nakagawa* ou les *pyramides adaptatives*. Les variations de luminance en haut du foie induisent des artefacts pour la *relaxation non supervisée* et la *corrélation d'histogrammes*. Le bruit de fond crée également des fausses régions pour certaines méthodes.

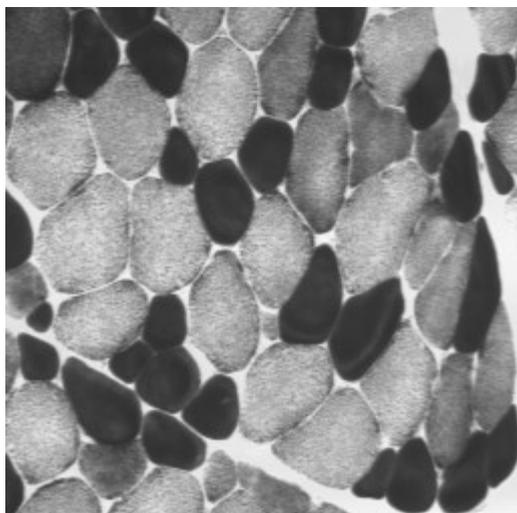
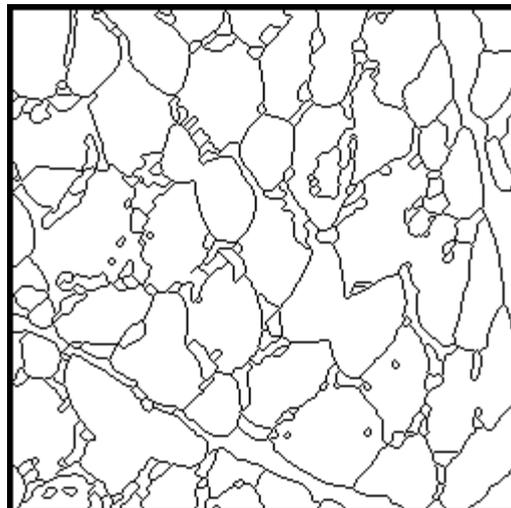
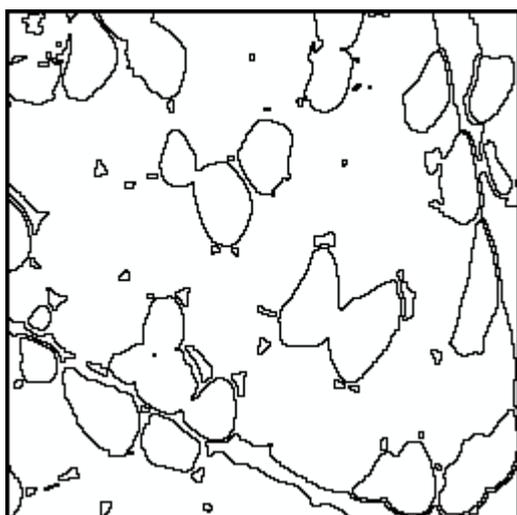
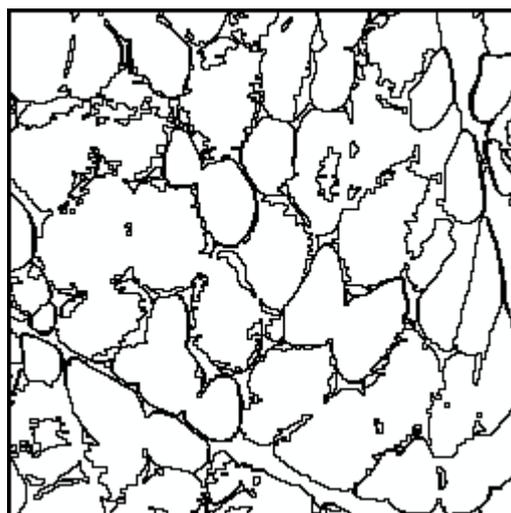
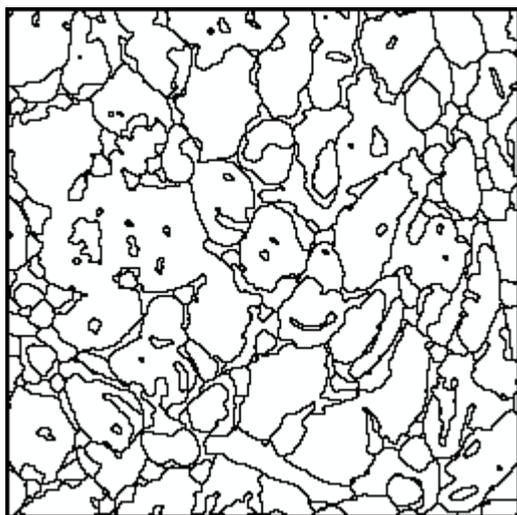
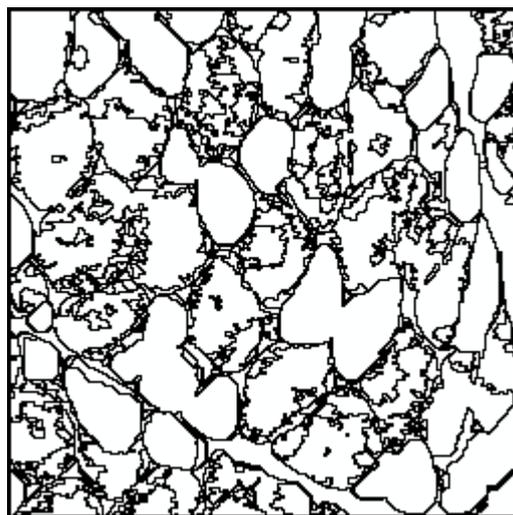
Image MUSCLE*Image originale**Fermeture de contours**Relaxation supervisée**Relaxation non supervisée*

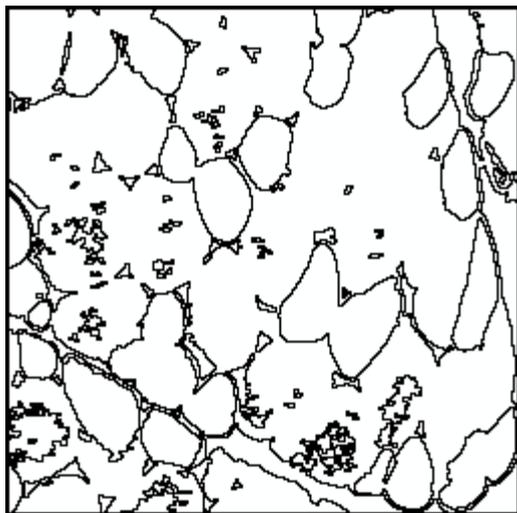
Figure 13.15 a : Résultats de segmentation par des méthodes de l'approche région sur l'image MUSCLE



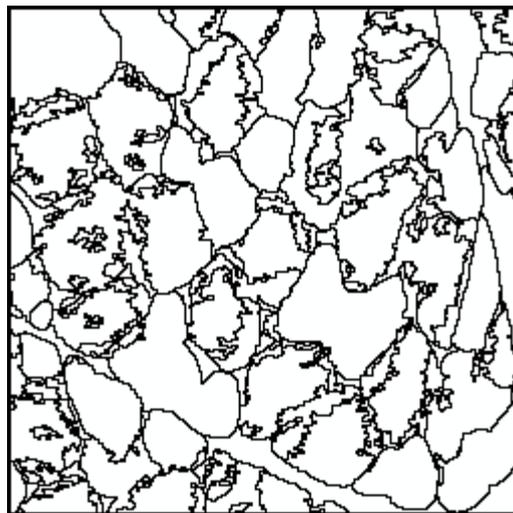
Corrélation d'histogrammes



Pyramides adaptatives



Méthode de Nakagawa



Méthode de Fisher

Figure 13.15 b : Résultats de segmentation par des méthodes de l'approche région sur l'image MUSCLE (suite)

La *relaxation supervisée* et *Nakagawa* détectent uniquement les fibres noires. Les méthodes ont des difficultés à s'adapter aux deux types de régions que comporte l'image : les limites des fibres texturées, quand celles-ci sont détectées, sont assez imprécises (nombreuses petites régions). Les *pyramides adaptatives* et *Fisher* détectent mieux les fibres sombres que les fibres claires (texturées) qui sont découpées en petites régions. Les méthodes extraient plus facilement les amas de fibres de même teinte que chaque fibre isolément. La *corrélation d'histogrammes* partage certains amas, mais avec une localisation des contours très imprécise. La *fermeture de contour* parvient également dans quelques cas à séparer les fibres de même nature.

3.4 Comparaison

Nombre de régions détectées

	Angio-graphie	Aquitaine	Bagues	Couloir	IRM	Muscle
Fermeture de contours	129	227	200	131	136	233
Relaxation supervisée	32	93	43	48	99	71
Relaxation non supervisée	93	195	122	113	44	91
Corrélation d'histogrammes	153	204	157	113	47	195
Pyramides adaptatives	34	240	102	92	41	192
Nakagawa	46	113	53	36	67	52
Fisher	224	196	20	180	89	144

Analyse quantitative des résultats

Calcul du contraste

	angiographie	aquitaine	bagues
Fermeture de contours	0.12	0.31	0.58
Relaxation supervisée	0.63	0.58	0.79
Relaxation non supervisée	0.21	0.20	0.45
Corrélation d'histogrammes	0.24	0.26	0.53
Pyramides adaptatives	0.24	0.38	0.76
Méthode de Nakagawa	0.51	0.56	0.79
Méthode de Fisher	0.17	0.47	0.82

	couloir	irm	muscle
Fermeture de contours	0.40	0.35	0.41
Relaxation supervisée	0.68	0.63	0.60
Relaxation non supervisée	0.27	0.59	0.11
Corrélation d'histogrammes	0.21	0.56	0.27
Pyramides adaptatives	0.63	0.67	0.47
Méthode de Nakagawa	0.62	0.67	0.58
Méthode de Fisher	0.29	0.22	0.44

Les valeurs des tableaux précédents ne sont pas utilisables pour les images trop texturées, par exemple, pour l'image MUSCLE les fibres sont très texturées et les valeurs associées à ces régions n'ont aucun sens. Par contre, pour l'image BAGUES qui n'est pas texturée, on peut noter que les méthodes déterminant correctement les bagues et, détectant peu de régions dans le fond, conduisent à des valeurs élevées (*Fisher*, *relaxation supervisée*, *Nakagawa* et *pyramides adaptatives*). ANGIOGRAPHIE et IRM ne sont pas trop texturées, les valeurs du contraste calculées confirment l'analyse visuelle, sauf pour la *fermeture de contours*. Il faut noter que pour cette dernière méthode, on a conservé dans l'image des régions les frontières détectées ce qui fausse légèrement la mesure effectuée. Le bon résultat de la relaxation supervisée sur ANGIOGRAPHIE est dû au fait qu'on a une seule région pour l'ensemble des vaisseaux détectés.

Les images initiales n'ont pas été prétraitées, aussi le bruit contenu dans les différentes régions et le flou des frontières peuvent conduire à des valeurs faibles. Le contraste, pour être utilisé efficacement, doit s'appliquer à des images de références peu bruitées et peu texturées.

3.5 Conclusion

Les résultats présentés montrent une sensibilité au bruit impulsif de la *relaxation supervisée*, de la *fermeture de contours* et de *Nakagawa*. Les *pyramides adaptatives* sont peu sensibles au bruit.

En milieu bruité, pour des transitions franches, on peut utiliser les *pyramides adaptatives* ou la *relaxation supervisée* (en absence de bruit impulsif).

En présence de transitions lentes, la méthode de *Fisher* et la *corrélation d'histogrammes* produisent des artefacts (faux contours); la *relaxation non supervisée* et les *pyramides adaptatives* créent de nombreuses petites régions dans les zones de transition.

La *fermeture de contours* coupe des zones homogènes en plusieurs régions, la *relaxation supervisée* conduit à de grandes régions.

Certains problèmes se règlent par un posttraitement, tel que la fusion des régions adjacentes sur un critère de taille ou de moyenne des intensités par exemple.

En guise de conclusion

Après cette étude comparative des méthodes de segmentation, qui s'est voulue la plus objective possible, aucune technique ne se détache pour prétendre au titre de « meilleure » méthode. En effet, chacune d'entre elles possède des avantages et des inconvénients, est plus adaptée à un type d'image ou à un problème donné qu'une autre. On ne peut même pas privilégier l'approche contour ou l'approche région de manière définitive. L'approche contour fournit généralement des contours bien localisés quand les transitions sont franches. Quand les frontières sont floues, certains opérateurs détectent et positionnent un seul contour (Deriche, Shen ou la relaxation markovienne supervisée), d'autres créent plusieurs contours ou de nombreuses petites régions dans la zone de transition.

Les variations lentes d'éclairage produisent plus d'artefacts avec les méthodes région qu'avec les méthodes contour. Ce problème, qui constitue une difficulté majeure des applications industrielles du traitement d'image, pourrait trouver une solution dans une méthode coopérative contour-région, les artefacts ayant alors peu de chances d'être positionnés au même endroit par les deux types d'algorithmes.

L'extraction des zones texturées trouve des solutions dans les deux types d'approches, dès lors que l'intensité moyenne différencie les régions. Le problème devient très ardu dans le cas contraire ; il faut alors faire appel à des attributs locaux dont, aucun ensemble, jusqu'à ce jour, ne s'est révélé discriminant.

L'avenir de l'analyse d'image est vraisemblablement dans la coopération de traitements, des travaux ont déjà été faits dans ce domaine. Citons la croissance de région guidée par les contours, les méthodes de fusion de résultats ou la coopération segmentation-interprétation, qui intègre des informations sémantiques liées au domaine d'application.

La solution des problèmes d'analyse d'images passe donc par l'enchaînement de certaines des « briques » que nous avons présentées dans cet ouvrage : prétraitement, extraction de contours ou de régions, posttraitement. Ces enchaînements ne sont probablement pas uniques et nécessitent de faire coopérer plusieurs opérateurs (de l'approche région et de l'approche contour). Il reste encore à l'utilisateur à définir ces enchaînements, à choisir les traitements en fonction de la nature des images et des buts à atteindre et enfin à régler les paramètres. La segmentation n'est donc pas un traitement isolé, elle s'intègre dans une approche « système » de l'analyse d'images. Nous espérons que cet ouvrage contribuera à une meilleure connaissance des traitements de bas-niveau et aidera l'utilisateur à bien exploiter la richesse en information des images.

ANNEXE A

Eléments de théorie des graphes

Diagramme de Voronoï

La structure de graphe est très souvent employée en analyse d'images. Quelques définitions de base sont rappelées ici, pour des développements plus complets, le lecteur pourra consulter par exemple [GOND79]. La deuxième partie de cette annexe est consacrée à la construction d'un graphe particulier, le diagramme de Voronoï.

1 Définitions générales de la théorie des graphes

1.1 Notions de graphe

Un *graphe* orienté est un couple $G = (X, U)$, X étant un ensemble fini d'éléments et U un sous-ensemble de $X \times X = \{(i, j) \mid i \in X, j \in X\}$.

Les éléments de X sont les *sommets* ou *nœuds* du graphe.

Les éléments de U sont les *arcs* du graphe.

La figure A.1 représente le graphe G avec :

$X = \{a, b, c, d, e, f\}$ et $U = \{(a, a), (a, b), (b, c), (c, b), (c, e), (d, c), (d, e)\}$

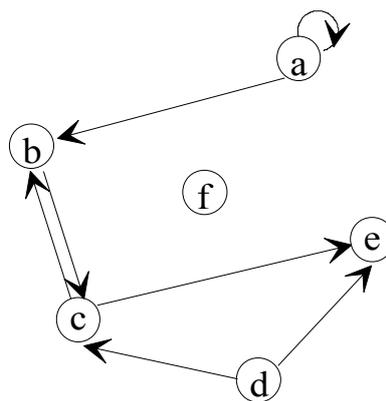


Figure A.1 : Exemple de graphe orienté

- L'ordre du graphe G est le cardinal de X ; pour l'exemple précédent, l'ordre de G est 6.

- Soit $u = (i, j)$ un arc quelconque du graphe G ($u \in U, i \in X, j \in X$) :
 - les sommets i et j sont les extrémités de u ; i est l'origine ou l'extrémité initiale de l'arc et j est l'extrémité terminale. i et j sont *adjacents*.
 - le sommet j est un successeur de i et i est un prédécesseur de j .
- Si $u = (i, i)$, u est une *boucle* ; c'est le cas pour l'arc issu du sommet a dans la figure A.1.
- Un *p-graphe* est un graphe tel que le nombre maximum d'arcs entre deux sommets i et j pris dans cet ordre est p .
- Si les propriétés du graphe G ne dépendent pas du sens des arcs, on dit que G est un graphe *non orienté*, on considère alors les couples non ordonnés (i, j) , qu'on appelle *arêtes*.
- Un graphe est *simple* s'il est sans boucle et s'il n'y a jamais plus d'une arête entre deux sommets donnés.
- Le degré d'un sommet est le nombre d'arcs (ou d'arêtes) qui ont ce sommet pour extrémité (initiale ou terminale ; une boucle est comptée deux fois).
- Un graphe est *complet* si pour toute paire de sommets (i, j) , $i \neq j$, il existe au moins une arête (i, j) .
- Un *sous-graphe* de G engendré par un sous-ensemble Y de X est le graphe dont les sommets sont les éléments de Y et dont les arcs sont les arcs de G ayant leurs extrémités dans Y .
- Un graphe *partiel* de G est un graphe (X, V) dont les sommets sont les éléments de X et les arcs sont les éléments de V , où V est un sous-ensemble de U .
- Un graphe est *planaire* s'il est possible de le représenter dans un plan sans que deux arcs ne se croisent.
- Un graphe est *biparti* si l'ensemble de ses sommets peut être partitionné en deux sous-ensembles X_1 et X_2 tels que deux sommets d'un même sous-ensemble ne soient jamais adjacents.

1.2 Chaîne, connexité

- Une *chaîne* de longueur q est une suite de q arcs $C = \{u_1, u_2, \dots, u_q\}$ telle que chaque arc u_r ait une extrémité commune avec u_{r-1} et l'autre extrémité commune avec u_{r+1} .
- Un *cycle* est une chaîne dont les extrémités coïncident et n'utilisant pas deux fois le même arc.
- Un *chemin* de longueur q est une suite de q arcs $C = \{u_1, u_2, \dots, u_q\}$ telle que l'extrémité terminale de u_r soit l'extrémité initiale de u_{r+1} ; autrement dit, un chemin est une chaîne dont les arcs sont tous orientés dans le même sens.
- Un chemin est *simple* si les arcs qui le composent sont distincts.

- Un chemin est *élémentaire* si les origines des arcs sont toutes distinctes.
- Un *circuit* est un chemin dont l'origine et l'extrémité coïncident.
- Un graphe est *connexe* si pour tout couple de sommets (i, j) , il existe une chaîne joignant i à j .

La relation \mathcal{R} définie sur X par :

$$i \mathcal{R} j \Leftrightarrow \begin{cases} i = j & \text{ou} \\ \text{il existe une chaîne joignant } i \text{ à } j \end{cases} \text{ est une relation d'équivalence. Les}$$

sous-graphes G_1, G_2, \dots, G_p engendrés par les classes d'équivalence X_1, X_2, \dots, X_p sont appelés les composantes connexes du graphe G .

- Un *arbre* est un graphe non orienté simple, connexe et sans cycle.
- Un graphe $G = (X, T)$ est une *arborescence* de racine r appartenant à X si :
 - G est un arbre,
 - pour tout sommet j de X , il existe dans G un chemin allant de r à j .

1.3 Graphe valué

- Un graphe $G = (X, U)$ est *valué* si à toute arête e est associé un réel $w(e)$, appelé poids ou longueur de e . On appelle poids de G le réel $w(G) = \sum_{e \in U} w(e)$.
- Soit $G = (X, U)$ un graphe non orienté ; un ensemble C d'arêtes, $C \subset U$, est un *couplage* si deux arêtes quelconques de C n'ont aucune extrémité en commun.
- Le poids d'un couplage $C \subset U$, est la somme des poids des arêtes qui le constituent $w(C) = \sum_{e \in C} w(e)$.

Le problème du couplage de poids maximum sur le graphe $G = (X, U)$ est la recherche d'un couplage $C \subset U$ tel que $w(C)$ soit maximum sur l'ensemble de tous les couplages de G .

2 Différentes approches pour la construction du diagramme de Voronoï

2.1 Approches algorithmiques et distance euclidienne

Le développement de méthodes algorithmiques pour construire le diagramme de Voronoï d'un ensemble de germes constitue un problème important en géométrie algorithmique [AURE91], [BERT94], [EDEL88], [PREP88], [TOUS85].

Il existe essentiellement deux grandes familles d'approches, l'une récursive (intéressante du point de vue de la complexité), et l'autre incrémentale (intéressante de par ses possibilités de mise à jour par insertion ou suppression de germes). Ces deux méthodes génèrent, pour la distance euclidienne, une représentation analogique du diagramme qui est ensuite discrétisée en une image.

Dans le cadre d'une représentation discrète, on peut leur préférer une méthode par propagation en utilisant les travaux effectués sur les distances discrètes et cartes de distance [CHAS91].

2.1.1 Approche Divide and Conquer

Cet algorithme, proposé en 1975 [SHAM75] [LEE 80], est de type récursif. Il a été repris de manière détaillée dans [PREP88]. Le diagramme de Voronoï d'un ensemble de n germes est construit avec une complexité $O(n \cdot \log_2 n)$, dans le pire des cas.

Soit \mathcal{S} l'ensemble des germes vérifiant les deux contraintes suivantes :

- il n'existe pas quatre germes cocirculaires ;
- il n'existe pas trois germes alignés.

Dans le premier cas un sommet de Voronoï aurait plus de trois voisins, et dans le second cas l'intersection des médiatrices ne serait pas définie.

L'algorithme est récursif et se décompose en trois étapes (figure A.2) :

- Séparer \mathcal{S} en deux sous-ensembles $\mathcal{S1}$ et $\mathcal{S2}$, de même taille à quelques éléments près, tels que $\mathcal{S1}$ et $\mathcal{S2}$ sont séparés par une ligne polygonale s formée des arêtes communes à des couples de polygones dont l'un est issu de $\mathcal{S1}$ et l'autre est issu de $\mathcal{S2}$. Tous les points de $\mathcal{S2}$ sont d'un même côté (à droite) de cette ligne et ceux de $\mathcal{S1}$ sont situés de l'autre côté (à gauche) de cette ligne.
- Construire $\text{Vor}(\mathcal{S1})$ et $\text{Vor}(\mathcal{S2})$ récursivement suivant le même principe ; la fin de la récursivité est donnée par $\mathcal{S1}$ ou $\mathcal{S2}$ réduit à un point (le diagramme de Voronoï d'un ensemble réduit à deux points étant la médiatrice de ces deux points).
- Fusionner $\text{Vor}(\mathcal{S1})$ et $\text{Vor}(\mathcal{S2})$ pour former $\text{Vor}(\mathcal{S})$.

La fusion est la phase la plus délicate, ce qui est courant avec ce type d'algorithmes récursifs. Des détails de cet algorithme sont fournis dans [PREP88] et [CHAS91].

2.1.2 Méthode incrémentale

La méthode du *Divide and Conquer* n'est pas adaptée pour des applications où l'on désire modifier la position ou le nombre de germes. En effet, elle nécessite de recalculer entièrement le diagramme à chaque modification.

Pour résoudre ce problème, une méthode incrémentale a été développée [GREE78]. Cette méthode a, par la suite, débouché sur une réalisation algorithmique plus rapide en pratique [OHYA84].

Avec la méthode incrémentale, le diagramme de Voronoï est construit étape par étape, en ajoutant un germe et un seul à chaque étape :

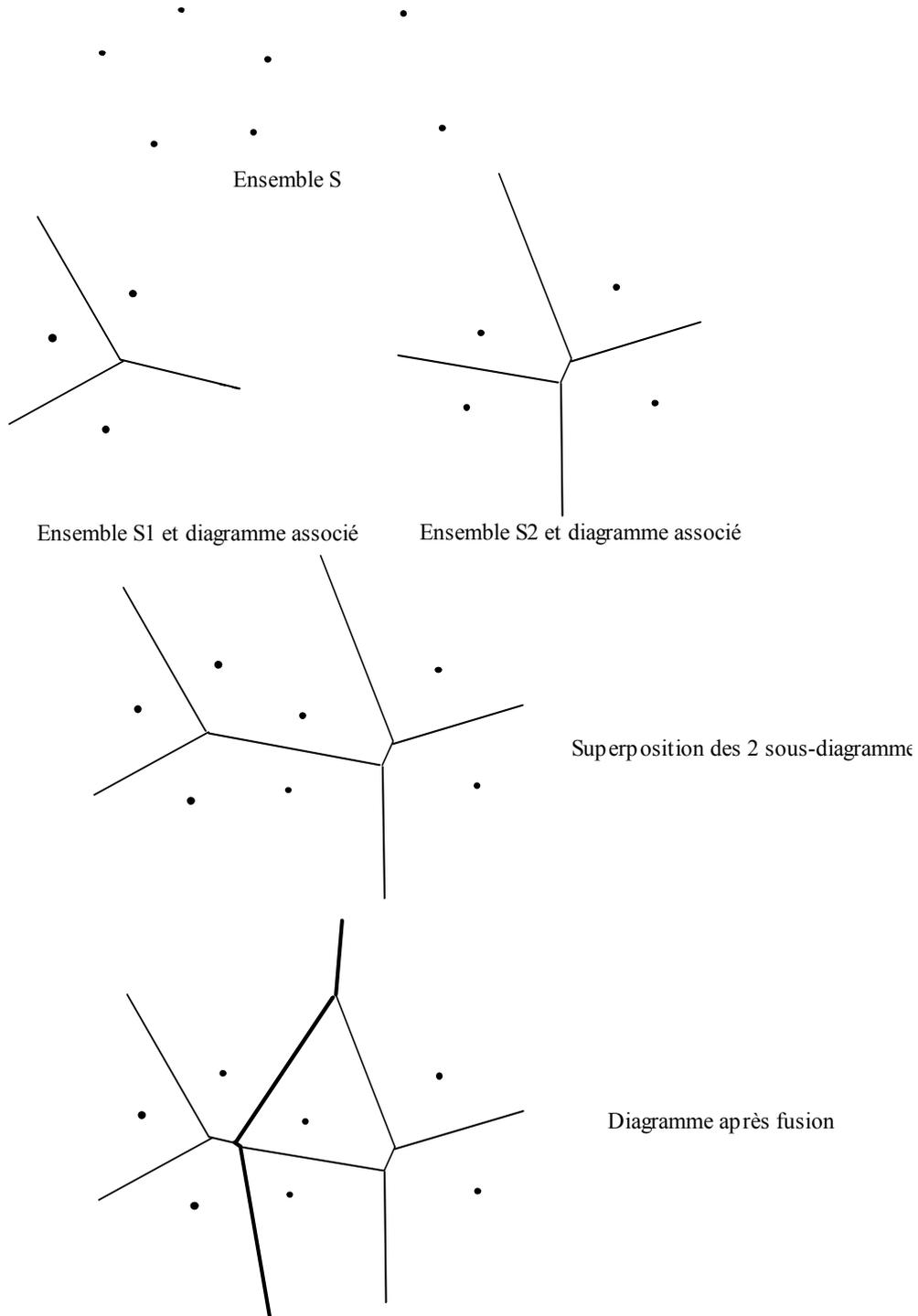


Figure A.2 : Illustration de l'algorithme Divide and Conquer dans le cas d'un ensemble S de 7 points.

Soit n germes distincts P_1, P_2, \dots, P_n et V_m le diagramme de Voronoï associé aux m premiers germes P_1, P_2, \dots, P_m . En partant d'un diagramme trivial, par exemple V_2 , la méthode incrémentale consiste à construire V_m à partir de V_{m-1} ($m \leq n$). L'adjonction d'un nouveau germe P_m au diagramme courant V_{m-1} induit une modification du diagramme qui s'effectue en deux étapes (cf. figure A.3) :

- recherche, parmi les germes de V_{m-1} , du plus proche voisin de P_m , noté $P_{N(m)}$.
- modification locale du diagramme de Voronoï : en commençant avec la médiatrice du segment joignant P_m à $P_{N(m)}$, on trouve l'arête de Voronoï associée à V_{m-1} intersectant cette médiatrice, le plus à droite possible de la direction donnée par $P_m P_{N(m)}$.

Soit m' le point voisin de $P_{N(m)}$ pour cette arête. On considère alors la médiatrice des germes P_m et $P_{m'}$, et on recommence le processus jusqu'à obtenir toutes les arêtes de Voronoï associées au polygone de Voronoï $V(P_m)$.

(Cette étape de modification nécessite que la condition de non-cocircularité soit vérifiée).

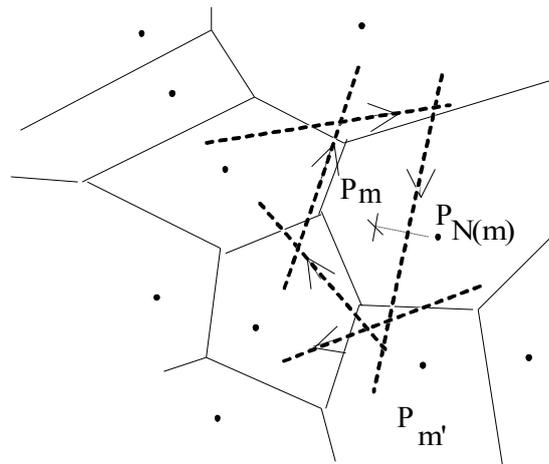


Figure A.3 : Illustration de la méthode incrémentale pour réaliser l'insertion du germe P_m (en flèches pointillées, les nouvelles arêtes)

2.2 Approches algorithmiques et distance discrète

Dans le cadre des travaux appliqués à l'image discrète, le diagramme de Voronoï peut également être vu comme une partition de l'ensemble discret Z^2 .

Pour cela, on utilise les distances discrètes dont les plus connues sont les distances d_4 (*city block*) et d_8 (*chessboard*) (définies dans le chapitre III § 2.1.2). La génération du partitionnement s'effectue par un algorithme de propagation, utilisant les transformations de distance pour lesquelles des schémas parallèle et séquentiel peuvent être utilisés [CHAS91].

On associe une image binaire à l'image des germes, et sur cette image binaire on applique la transformation de distance ; les arêtes de Voronoï sont alors composées des points situés à une même distance de deux germes distincts.

Les distances du chanfrein pondérant les déplacements élémentaires directs (horizontaux ou verticaux) ou indirects (diagonaux) peuvent également être exploitées afin d'avoir une meilleure approximation par rapport à la distance euclidienne [THIE92].

L'intérêt essentiel de l'utilisation des méthodes séquentielles est, outre leur simplicité, un temps de calcul indépendant du nombre de germes, du fait qu'elles ne nécessitent que deux passages.

Le choix des distances utilisées est important et comme on le remarque sur la figure A.2, ceci se répercute sur la forme des polygones de Voronoï : ceux-ci ne sont pas nécessairement convexes, excepté pour la distance euclidienne.

Pour la distance d_4 , il se forme des zones de points à égale distance de deux germes, et les résultats sont très éloignés de ceux de la distance euclidienne.

Pour la distance d_8 les résultats présentent également de nombreuses zones de points à égale distance de deux germes.

Pour les distances du chanfrein, les résultats s'améliorent : il apparaît moins de taches et les frontières « ondulées » moins.

Les distances discrètes génèrent directement le diagramme sous forme d'image, ce qui est généralement le résultat escompté.

Cependant, à leur désavantage, on perd l'information d'adjacence entre polygones voisins, contenue dans la structure algorithmique issue de la construction en continu (*Divide and Conquer* ou méthode incrémentale). Une phase de traitement supplémentaire est alors nécessaire pour rétablir le graphe d'adjacence.

ANNEXE B

Champs markoviens et segmentation

L'utilisation de techniques markoviennes en traitement d'image, et plus particulièrement en segmentation, part du principe suivant : il existe au moins trois domaines implicitement ou explicitement abordés lors de la segmentation et qui utilisent des modélisations stochastiques. Plus précisément :

- La modélisation de textures lorsqu'elles sont présentes dans l'image concernée. En effet, il est clair qu'une texture, même si elle présente parfois une régularité assez grande, n'est jamais parfaitement régulière et présente un caractère « aléatoire », au sens commun du terme, que chacun reconnaît.
- Les déformations observées entre une image « idéale », telle que, par exemple, un chiffre écrit à l'encre noire sur du papier blanc qui devrait donc donner, après numérisation, une image parfaitement binaire, et l'image numérisée font intervenir des modèles physiques plus ou moins complexes qui se traduisent naturellement par des modèles de déformations aléatoires.
- Les critères que l'on est amené à choisir pour décrire la segmentation que l'on souhaite calculer sont généralement complexes, et bien qu'ils doivent être optimisés, ils présentent rarement une forme qui permette leur optimisation directe. L'intérêt de l'utilisation d'algorithmes stochastiques a été prouvé dans de nombreux domaines, et les techniques de relaxation qui ont été employées depuis fort longtemps avec succès ne font que confirmer ce point puisqu'elles peuvent être facilement traduites en termes d'algorithmes stochastiques, bénéficiant ainsi de résultats théoriques de convergence.

Il est donc naturel de plonger le problème de la segmentation dans un cadre stochastique permettant à la fois de modéliser les textures si elles sont présentes, de modéliser les déformations subies lors de l'acquisition, et de disposer d'algorithmes d'optimisation efficaces et dont le comportement est connu théoriquement. Le cadre stochastique que nous présentons ici est celui des champs markoviens qui constituent, en fait, une famille très vaste de modèles stochastiques. De plus, ils sont, de par leur définition, intrinsèquement liés à la notion de système de voisinages qui constituent une notion de base en imagerie.

1 Définitions

1.1 Champs markoviens

Soit une image définie sur une maille dont les sommets constituent un ensemble \mathcal{S} de sites s :

$$\mathcal{S} = \{s=(i, j) ; 1 \leq i, j \leq N\} \quad (\text{B-1})$$

A chaque site s est associé une variable aléatoire A_s , dont les valeurs a_s appartiennent à un ensemble Ω . Par exemple, $\Omega = \{0, \dots, 255\}$ représente les valeurs de niveaux de gris d'une image monochrome et $\Omega = \{0, \dots, 255\}^q$ représente les valeurs susceptibles d'être associées à un pixel d'une image multispectrale à q canaux.

L'image est alors considérée comme un vecteur aléatoire $A = (A_s, s \in \mathcal{S})$ dont le vecteur $\mathbf{a} = (a_s, s \in \mathcal{S})$ est une configuration (ou réalisation). Une configuration est un élément de l'espace produit $\chi = \Omega^{N \times N}$.

$P(\mathbf{A} = \mathbf{a}) = P(\mathbf{a})$ est la probabilité de la configuration \mathbf{a} , c'est la probabilité jointe $P(A_s = a_s, s \in \mathcal{S})$. On notera aussi $P(A_s = a_s) = P(a_s)$ la loi marginale de A_s .

Un système de voisinages $V = (v_s, s \in \mathcal{S})$ est formé de parties v_s de \mathcal{S} , vérifiant les propriétés suivantes:

$$\begin{cases} s \notin v_s \\ s \in v_t \Rightarrow t \in v_s \end{cases}$$

L'ensemble v_s est appelé le voisinage de s ; t est dit voisin de s si $t \in v_s$. Cette relation est symétrique de par la définition du système de voisinages. Un champ aléatoire \mathbf{A} est un *champ markovien* associé à ce système de voisinages si et seulement si :

$$\begin{cases} P(\mathbf{a}) > 0 \\ P(a_s | a_r, r \in \mathcal{S} - \{s\}) = P(a_s | a_r, r \in v_s) \end{cases} \quad (\text{B-2})$$

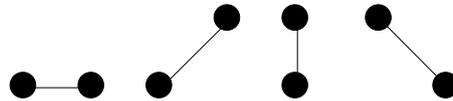
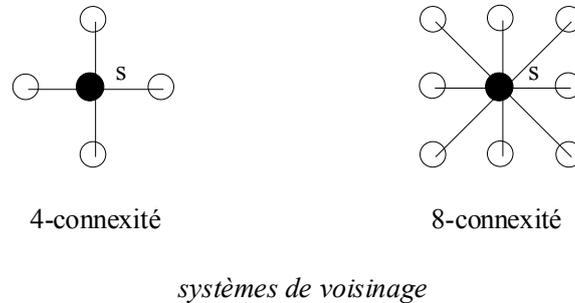
ce qui signifie que la connaissance d'un voisinage du pixel s est suffisante pour calculer la loi marginale en chaque pixel s .

Un sous-ensemble c de \mathcal{S} est appelé une *clique* relative au système de voisinages V si :

- c est un singleton,
- ou
- deux pixels quelconques de c sont voisins (au sens de V).

Remarquons que, en considérant la maille associée à V comme un graphe, une clique est un sous-graphe complet de ce graphe. On notera \mathcal{C} l'ensemble des cliques de \mathcal{S} . L'ordre d'une clique correspond au nombre de ses éléments (ou sommets). Quand cela sera nécessaire, on précisera l'ordre en notant c_n une clique d'ordre n . \mathcal{C}_n sera l'ensemble des cliques d'ordre n sur \mathcal{S} . On notera le plus souvent $c = \langle s, t \rangle$ une clique d'ordre 2 (à titre d'exemple, les cliques d'ordre 2 correspondant au système de voisinages des huit plus proches voisins sont représentées dans la figure 1).

Différents systèmes de voisinages et des exemples de cliques sont représentés figure 1. Le voisinage d'ordre 1 d'un pixel contient ses voisins en 4-connexité et celui d'ordre 2 contient ses voisins en 8-connexité (cf. figure 1).



cliques d'ordre 2 dans le système en 8-connexité

Figure B.1 : Voisinages et cliques correspondantes

1.2 Mesure de Gibbs et champs markoviens

Soient un système de voisinages $V = \{v_s, s \in S, v_s \subset S\}$ et C l'ensemble des cliques définies sur V , le théorème de Hammersley-Clifford [BESA74] démontre qu'un champ aléatoire \mathbf{A} est un champ markovien associé au système de voisinages V si et seulement si sa distribution de probabilité $P(\mathbf{A} = \mathbf{a})$ est une mesure de Gibbs définie par:

$$\forall \mathbf{a} \in \mathcal{X}, P(\mathbf{a}) = \frac{\exp - U(\mathbf{a})}{Z} \quad (\text{B-3})$$

où U est une fonction appelée énergie, définie par :

$$U(\mathbf{a}) = \sum_{c \in C} V_c(\mathbf{a}_{/c}) \quad (\text{B-4})$$

Les fonctions V_c sont appelées des fonctions potentiel, et ne dépendent que des éléments de \mathbf{a} qui appartiennent à la clique c , la notation $\mathbf{a}_{/c}$ désignant la restriction de \mathbf{a} à c . Par exemple, dans le cas le plus simple de modèle d'Ising, $V_c(\mathbf{x}) = x_s x_t$, où c est la clique formée des deux sites s et t , voisins verticaux ou horizontaux, tels ceux qui sont représentés dans la figure 1.

Z est généralement appelée *fonction de partition*, c'est en fait la constante de normalisation de la probabilité P , elle est définie par :

$$Z = \sum_{\mathbf{a} \in \mathcal{S}} \exp(-U(\mathbf{a})) \quad (\text{B-5})$$

1.3 Expression des probabilités conditionnelles

L'interprétation d'un champ markovien défini par une énergie U du type précédent est fondamentalement basée sur les probabilités conditionnelles :

$$P(A_s = v | \mathbf{a}_r, r \neq s) = \frac{P(A_s = v; \mathbf{a}_r, r \in \mathcal{S})}{\sum_{w \in \mathcal{S}} P(A_s = w; \mathbf{a}_r, r \in \mathcal{S})} \quad (\text{B-6})$$

$$= \frac{\exp\left(-\sum_{\mathbf{c}/s \in \mathcal{C}} V_{\mathbf{c}}(\mathbf{a}_{/\mathbf{c}, a_s=v})\right)}{\sum_{w \in \mathcal{S}} \exp\left(-\sum_{\mathbf{c}/s \in \mathcal{C}} V_{\mathbf{c}}(\mathbf{a}_{/\mathbf{c}, a_s=w})\right)} = P(A_s = v | \mathbf{a}_r, r \in \mathcal{V}_s) \quad (\text{B-7})$$

où $\mathbf{a}_{/\mathbf{c}, a_s=v}$ représente une restriction de \mathbf{a} à \mathcal{C} dans laquelle $a_s = v$.

Remarquons tout d'abord que les sommes définissant ces probabilités conditionnelles dans la formule (B-7) contiennent un très petit nombre de termes (ce nombre est directement lié à la taille des voisinages) et donc que leur calcul est relativement rapide. Ces probabilités conditionnelles sont communément associées à des énergies conditionnelles qui sont déduites du numérateur de l'équation précédente. On note :

$$U(\mathbf{a}_s | \mathbf{a}_r, r \in \mathcal{V}_s) = \sum_{\mathbf{c}/s \in \mathcal{C}} V_{\mathbf{c}}(\mathbf{a}_{/\mathbf{c}}) \quad (\text{B-8})$$

Nous allons reprendre ce calcul de probabilités conditionnelles dans un cas simple de modèle d'Ising, afin d'illustrer ce qui précède. Soit $U(\mathbf{a}) = \sum_{\langle s,t \rangle \in \mathcal{C}} a_s a_t$,

les cliques utilisées étant les cliques horizontales et verticales associées à la 4-connexité, on obtient alors:

$$P(a_s = v | \mathbf{a}_r, r \in \mathcal{S} - \{s\}) = \frac{\exp\left(-\sum_{\langle i,j \rangle / i \neq s, j \neq s} a_i a_j - v \sum_{t \in \mathcal{V}_s} a_t\right)}{\sum_w \exp\left(-\sum_{\langle i,j \rangle / i \neq s, j \neq s} a_i a_j - w \sum_{t \in \mathcal{V}_s} a_t\right)} \quad (\text{B-9})$$

On remarque immédiatement que les premiers termes du numérateur et du dénominateur sont les mêmes et donc se simplifient, ce qui donne:

$$P(a_s = v | \mathbf{a}_r, r \in \mathcal{S} - \{s\})$$

$$= P(\mathbf{a}_s = \mathbf{v} \mid \mathbf{a}_r, \mathbf{r} \in \mathbf{v}_s) = \frac{\exp\left(-v \sum_{t \in \mathbf{V}_s} a_t\right)}{\sum_w \exp\left(-w \sum_{t \in \mathbf{V}_s} a_t\right)} \quad (\text{B-10})$$

Cette dernière formule correspond à celle énoncée précédemment dans le cas général.

2 Approche markovienne de la segmentation

Dans cette approche, la segmentation est considérée comme l'estimation d'un processus label Λ à partir de la réalisation d'un processus pixel \mathbf{A} . Plusieurs techniques sont utilisables : on peut ne modéliser que $P(\cdot \mid \mathbf{A} = \mathbf{a})$, ou utiliser une approche bayésienne qui consiste à construire $P(\mathbf{A} = \mathbf{a} \mid \cdot)$, puis à fixer une loi *a priori* sur Λ , notée $P(\cdot)$, à partir de laquelle on calcule $P(\cdot \mid \mathbf{A} = \mathbf{a})$ par la formule de Bayes :

$$P(\cdot \mid \mathbf{A} = \mathbf{a}) = \frac{P(\mathbf{A} = \mathbf{a}, \cdot)}{P(\mathbf{A} = \mathbf{a})} = \frac{P(\mathbf{A} = \mathbf{a} \mid \cdot)P(\cdot)}{P(\mathbf{A} = \mathbf{a})} \quad (\text{B-11})$$

Pour simplifier les notations et quand il n'y a pas d'ambiguïté, nous noterons : $P(\lambda)$ pour $P(\Lambda = \lambda)$, $P(\mathbf{a})$ pour $P(\mathbf{A} = \mathbf{a})$, etc.

Dans l'expression B-11, $P(\mathbf{a})$ est constante car c'est une réalisation du processus observé, $P(\mathbf{a} \mid \cdot)$ est la probabilité de réalisation d'une configuration donnée conditionnellement à l'étiquetage et $P(\cdot)$ représente la loi de probabilité associée au champ Λ , encore appelée loi *a priori* ; la loi déduite, $P(\cdot \mid \mathbf{a})$, étant, elle, la loi *a posteriori*.

La justification la plus simple de l'utilisation de techniques bayésiennes se trouve dans les problèmes de segmentation d'images non texturées dans lesquelles il est souvent possible de préciser les déformations, dues aux capteurs par exemple, subies par une image idéale λ lors de l'acquisition conduisant à \mathbf{a} et de traduire ces informations en un modèle, souvent simple, $P(\mathbf{a} \mid \cdot)$. Par ailleurs, les objets observés sont généralement réguliers, et conduisent à une modélisation *a priori* du type d'objets, donc des régions recherchées, qui est explicitée dans $P(\cdot)$. On emploie souvent pour loi *a priori* en segmentation un modèle d'Ising ou de Potts [PARI88] isotrope en 4 ou 8-connexité.

Le cadre qui nous intéresse ici est celui des lois markoviennes, on obtient ainsi une expression en termes d'énergie du modèle considéré :

$$P(\mathbf{a}) = \frac{\exp(-U_1(\mathbf{a}))}{Z_1} \quad \text{et} \quad P() = \frac{\exp(-U_2())}{Z_2} \quad (\text{B-12})$$

puis, d'après (B.9) :

$$P(\mathbf{a}) = \frac{\exp(-U_1(\mathbf{a}) - U_2())}{Z_3} \quad (\text{B-13})$$

où Z_1 , Z_2 et Z_3 sont les constantes de normalisation appropriées.

La segmentation consiste alors à estimer la configuration λ . Plusieurs estimateurs sont utilisables, et nous présentons trois de ces estimateurs dans le paragraphe 3. Le plus courant est le MAP, c'est à dire l'estimateur du maximum *a posteriori*, qui maximise $P(\mathbf{a})$ ce qui, en termes d'énergie, revient à maximiser en λ ,

$$\exp(-U_1(\mathbf{a} |)) \exp(-U_2()) \quad (\text{B-14})$$

donc à minimiser $U_1(\mathbf{a} |) + U_2()$ en utilisant l'un des algorithmes d'optimisation décrits au paragraphe 6.

En général, les lois $P(\mathbf{a} |)$ et $P()$ utilisées sont paramétrées. Il faut donc avant de réaliser la segmentation, estimer ces paramètres. Dans certains cas, les paramètres figurant dans $P(\mathbf{a} |)$ sont estimés sur des échantillons présélectionnés, et on parle alors de segmentation supervisée. Dans d'autres cas, les paramètres sont estimés ou fixés sans utiliser de connaissances précises de ce qui se trouve dans l'image étudiée, et on parle alors de segmentation non-supervisée. Différentes techniques d'estimation de ces paramètres peuvent être utilisées ; nous présenterons l'une de ces techniques au paragraphe 4.

3 Estimation de la configuration λ

Il existe plusieurs méthodes d'estimation de λ , la plus communément utilisée est la maximisation de la probabilité *a posteriori* (MAP), qui revient à minimiser l'énergie $U(\mathbf{a} |)$. Deux autres méthodes sont généralement définies, le MPM (*Maximum Posterior Marginal*) et le TPM (*Threshold Posterior Mean*). Chacune de ces méthodes peut être aisément décrite en utilisant une fonction de perte: soit $L(\lambda, \hat{\lambda})$, le coût de l'erreur faite en estimant λ par $\hat{\lambda}$.

Le MAP correspond à $L(\lambda, \mathbf{b}) = 1$ si $\lambda \neq \mathbf{b}$ et 0 sinon, ce qui revient, ainsi que nous l'avons déjà dit, à choisir un estimateur $\hat{\lambda}$ qui maximise $P(\lambda | \mathbf{a})$. Cette maximisation est souvent réalisée par un algorithme de recuit simulé.

Le MPM correspond à $L(\lambda, \mathbf{b}) = \text{card} \{s, \lambda_s \neq b_s\}$ ce qui se traduit par une maximisation des probabilités conditionnelles $P(\lambda_s | \mathbf{a})$. Remarquons que ces probabilités sont conditionnées par l'observation \mathbf{a} uniquement et non par le

voisinage dans λ de s . Ceci implique le calcul d'une loi marginale donc une sommation sur l'ensemble des configurations possibles de λ . Cette sommation est en fait approximée par des calculs de type champs moyens [PARI88] [GEIG91].

Le TPM est associé à :

$$L(\lambda, b) = \|\lambda - b\|^2 = \sum_s (\lambda_s - b_s)^2 \quad (\text{B-15})$$

et il se traduit par le calcul de la moyenne de λ sur l'ensemble des configurations. Là encore, le calcul ne peut être fait exactement et il est approximé [PARI88] [GEIG91].

Les paragraphes 5 et 6 donnent une description rapide (et souvent simplifiée) des algorithmes classiques liés aux champs markoviens ; on trouvera plus de détails dans les publications citées en référence.

4 Estimation des paramètres

Pour présenter le principe général de l'estimation, supposons que la modélisation employée utilise une fonction d'énergie $U(\mathbf{a})$ dépendant du paramètre θ , que nous noterons alors $U(\mathbf{a}, \theta)$. Face à une configuration \mathbf{a} observée, le problème est alors d'estimer la valeur $\hat{\theta}$ du paramètre qui « correspond le mieux » à cette réalisation. Ceci ne peut être défini que si l'on choisit un critère permettant de quantifier l'adéquation entre θ_0 , vraie valeur du paramètre, et $\hat{\theta}$ sachant \mathbf{a} . Ceci explique qu'il puisse exister plusieurs estimateurs en fonction du critère choisi. Considérons un exemple extrêmement simple : supposons que nous souhaitons estimer la moyenne μ d'une loi normale P_μ , dont la variance est supposée connue, à partir d'une seule observation. L'estimation qui semble la plus raisonnable est de choisir pour estimateur de μ l'observation dont on dispose, on obtient ainsi un estimateur dont on imagine facilement qu'il ne peut être très précis. Cet estimateur est l'estimateur du maximum de vraisemblance qui consiste à rechercher la valeur de μ qui rend $P_\mu(x)$ maximal, et il est rendu de plus en plus précis lorsque l'on multiplie le nombre d'observations.

De la même façon, l'estimation des paramètres d'un champ markovien à partir d'une seule réalisation nécessite une hypothèse d'ergodicité. Ainsi on suppose que les caractéristiques locales extraites à différents endroits d'une réalisation du champ sont équivalentes à celles extraites sur plusieurs réalisations, ce qui permet de disposer à partir d'une seule image observée d'une grande quantité d'observations. Les méthodes d'estimation se placent, pour la plupart, dans le cadre stationnaire.

Ainsi, si l'on considère l'estimateur du maximum de vraisemblance, on fixe une suite croissante \mathbf{D}_n de sous-ensembles convexes de \mathcal{S} et on recherche la valeur de θ qui maximise la vraisemblance définie à partir de la distribution de Gibbs restreinte à \mathbf{D}_n :

$$P_{\theta,n}(a) = \frac{1}{Z_n(\theta)} \exp(-U_n(a, \theta)) \quad (\text{B-16})$$

Par ailleurs, il est souhaitable qu'un estimateur s'approche de plus en plus de la vraie valeur de θ quand la taille de D_n tend vers l'infini, c'est-à-dire quand n tend vers l'infini. On dit alors que l'estimateur est consistant.

Un grand nombre d'estimateurs peuvent être utilisés ; citons l'estimateur du maximum de vraisemblance [YOUN89], l'estimateur du maximum de pseudo-vraisemblance [GEMA87], [GUYO93], l'estimateur EM [CELE88], [DEMP77], l'estimation de la validation croisée [GUYO93] et celui des « boîtes qualitatives » [AZEN87], [CHAL88], cette liste n'étant pas exhaustive. Parmi les techniques de segmentation qui ont été présentées au chapitre XI, l'une d'entre elles utilise l'estimateur du maximum de pseudo-vraisemblance, et nous allons le présenter brièvement ici.

L'estimateur du maximum de pseudo-vraisemblance [BESA86], [GEMA87] est obtenu en maximisant la fonction :

$$Q_n(\theta) = \prod_{s \in \text{intérieur de } D_n} P_{\theta,n}(a_s | a_r, r \in s) \quad (\text{B-17})$$

où a est la réalisation observée du champ de loi $P_{\theta,n}$.

Cette expression est plus simple à exploiter, en particulier quand l'énergie $U_n(a, \theta)$ est une fonction linéaire du paramètre. La fonction $\ln(Q_n(\theta))$ est alors une fonction convexe et on peut utiliser une méthode de descente de gradient pour obtenir $\hat{\theta}_n$. Moyennant certaines hypothèses dont celle de stationnarité, on montre [GEMA87] que l'estimateur ainsi obtenu est consistant, c'est-à-dire que :

$$\forall \varepsilon > 0, \lim_{n \rightarrow \infty} P(|\hat{\theta}_n - \theta_0| < \varepsilon) = 1 \quad (\text{B-18})$$

5 Algorithmes de simulation

Les deux algorithmes présentés ici sont des algorithmes de simulation, c'est à dire qu'ils permettent de construire des configurations suivant la loi choisie. Ainsi, par exemple, après avoir estimé les paramètres d'un modèle de texture, on peut simuler (ou synthétiser) un échantillon de la loi que l'on vient de choisir.

La loi à échantillonner est notée :

$$P_T(a) = \frac{\exp\left(-\frac{U(a)}{T}\right)}{Z(T)} \quad (\text{B-19})$$

où T est un paramètre que nous appellerons la température et dont le rôle sera explicité plus tard.

Ces deux algorithmes sont itératifs, les exposants (n) désigneront le nombre de sites mis à jour. Ainsi, les valeurs affectées au voisinage d'un site ont été remises à jour lorsque l'on examine ce site. Un balayage est alors constitué d'autant de mises à jour qu'il y a de sites dans la grille concernée.

Le principe de ces algorithmes, ainsi d'ailleurs que de l'algorithme du recuit simulé présenté dans la section suivante, est basé sur la construction d'une chaîne de Markov qui converge théoriquement en loi vers la loi recherchée. Nous présentons des versions simplifiées de ces algorithmes, et recommandons les références citées pour plus de détails.

5.1 Echantillonneur de Gibbs

L'algorithme se décompose de la manière suivante [GEMA84], [VANL87] :

- Tirer une configuration initiale $\mathbf{a}^{(0)}$, état initial de la chaîne de Markov.
- Balayer l'ensemble des sites \mathcal{S} (ligne par ligne par exemple, ou dans n'importe quel ordre). En chaque site s , calculer l'ensemble des probabilités conditionnelles $P(A_s = a \mid \mathbf{a}_r^{(n)}, \mathbf{r} \in \mathcal{V}_s^{(n)})$. Ensuite, effectuer un tirage aléatoire suivant la loi précédente pour obtenir $a_s^{(n+1)}$.
- Poursuivre les balayages pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

Il est prouvé que si le nombre de balayages tend vers l'infini, le processus ainsi construit converge en loi vers la loi de Gibbs P_T . Dans la pratique, la difficulté essentielle est de choisir le nombre de balayages, ou un critère d'arrêt. En règle générale, il est nécessaire d'utiliser un nombre de balayages plus grand lorsque la complexité de l'énergie augmente, ou, pour une énergie donnée, lorsque la température T diminue.

5.2 Algorithme de Métropolis

Cet algorithme a été proposé en 1953 [METR53]. Il se décompose comme suit [VANL87] :

- Tirer une configuration initiale $\mathbf{a}^{(0)}$.
- Balayer l'ensemble des sites \mathcal{S} . Chaque site s visité à l'étape n est caractérisé par son état $a_s^{(n)}$ et par la configuration de son voisinage $\mathbf{v}_s^{(n)}$. Son nouvel état $a_s^{(n+1)}$ est obtenu de la manière suivante :
 - On effectue un tirage aléatoire suivant une loi fixée *a priori*, uniforme sur Ω par exemple. Soit ξ l'état ainsi obtenu.
 - On calcule la variation d'énergie locale associée:

$$\Delta U = U(\xi \mid \mathbf{a}_r^{(n)}, \mathbf{r} \in \mathcal{V}_s^{(n)}) - U(\mathbf{a}_s^{(n)} \mid \mathbf{a}_r^{(n)}, \mathbf{r} \in \mathcal{V}_s^{(n)})$$

et on pose :

$$\mathbf{a}_s^{(n+1)} = \begin{cases} \xi & \text{si } \Delta U > 0 \\ \mathbf{a}_s^{(n)} & \text{si } \Delta U < 0, \xi \end{cases} \begin{cases} \text{avec la probabilité } \exp(-\Delta U/T) \\ \text{avec la probabilité } 1 - \exp(-\Delta U/T) \end{cases}$$

- Poursuivre les balayages pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

Comme précédemment, il est montré que si le nombre de balayages tend vers l'infini, le processus ainsi construit converge en loi vers la distribution de Gibbs P_T .

6 Algorithmes d'optimisation

Envisageons le cas de la segmentation où le problème est de trouver une configuration λ d'étiquettes correspondant à une configuration \mathbf{a} de l'image observée. Ayant défini une distribution de probabilité *a posteriori* $P(\lambda|\mathbf{a})$ et la fonction d'énergie associée, $U(\lambda|\mathbf{a})$, plusieurs algorithmes d'optimisation, c'est-à-dire d'estimation de λ , ont été proposés. Nous noterons :

$$P_T(\lambda|\mathbf{a}) = \frac{\exp\left(-\frac{U(\lambda|\mathbf{a})}{T}\right)}{Z(T)} = \frac{\exp\left(-\frac{U(\lambda)}{T}\right)}{Z(T)} \quad (\text{B-20})$$

afin de simplifier les notations.

6.1 Algorithme ICM (*Iterated Conditional Modes*)

Il s'agit d'un simulateur de Gibbs à température nulle [BESA74], [BESA86]. Les différentes étapes sont les suivantes :

- Choisir une configuration initiale λ_0 aussi proche que possible de la configuration optimale.
- Balayer l'ensemble des sites \mathcal{S} . Chaque site s visité à l'étape n est caractérisé par son état $\lambda_s^{(n)}$ et par la configuration de son voisinage $\mathbf{v}_s^{(n)}$. Son nouvel état $\lambda_s^{(n+1)}$ est la valeur qui correspond à l'énergie conditionnelle $U(\lambda_s^{(n+1)} | \lambda_r^{(n)}, r \in \mathbf{v}_s^{(n)})$ minimale.
- Poursuivre les balayages pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

Cet algorithme est parfaitement déterministe dans la mesure où il n'y a aucun tirage aléatoire, le seul paramètre qui influe sur le résultat final, outre le nombre de balayages, est l'ordre dans lequel on balaye l'image. D'autre part, il y a cette fois un critère d'arrêt simple qui consiste à compter le nombre de modifications d'état lors d'un balayage et à s'arrêter si ce nombre est nul ou suffisamment petit, ce qui se produit d'ailleurs en un nombre généralement faible de balayages.

Cet algorithme converge vers un minimum local de l'énergie *a posteriori*, cependant, le résultat obtenu dépend fortement de l'initialisation. Notons que cet algorithme ne comporte pas la caractéristique essentielle du recuit simulé qui est de visiter plusieurs puits d'énergie avant de se stabiliser dans l'un d'eux. Cet algorithme ne réalise qu'une descente dans un puits d'énergie déterminé par la configuration initiale et le mode de balayage de l'image. En ce sens, il ne peut être adapté qu'à des énergies pour lesquelles on peut disposer d'une configuration initiale convenable. Cependant ce genre d'énergies peut exister pour certains problèmes, et la sélection de telles énergies est alors une bonne solution au problème des temps de calculs. En effet, le nombre d'itérations nécessaires pour un ICM est en général très nettement inférieur à celui nécessaire pour un recuit simulé.

6.2 Recuit stochastique simulé

C'est un simulateur de Gibbs combiné avec une décroissance de température [KIRK83], [GEMA84], [VANL87]. Cet algorithme converge théoriquement vers un minimum global de la fonction d'énergie indépendamment de l'état initial utilisé. son schéma général est le suivant :

- Tirer une configuration initiale $\lambda^{(0)}$.
- Fixer une température initiale T_0 et poser $T = T_0$.
- Balayer l'ensemble des sites \mathcal{S} (ligne par ligne par exemple, ou dans n'importe quel ordre). Chaque site s visité à l'étape n est caractérisé par son état $\lambda_s^{(n)}$ et par les configurations $v_s^{(n)}$ de son voisinage $v_s^{(n)}$. Son nouvel état $\lambda_s^{(n+1)}$ est obtenu de la manière suivante:
 - Calculer les probabilités conditionnelles $P_T(\lambda_r^{(n)}, r \in v_s^{(n)})$
 - Effectuer un tirage aléatoire de $\lambda_s^{(n+1)}$ suivant cette loi.
- Faire décroître la température : $T = T_k$, où k est le numéro de l'itération.
- Poursuivre les balayages pour un nombre déterminé d'itérations ou jusqu'à la réalisation d'un critère d'arrêt.

Cette fois, en plus du nombre total d'itérations, il y a deux paramètres importants : la température initiale et le rythme de décroissance de la suite (T_k) . Pourvu que T_0 soit assez grande et que la décroissance de la température soit assez lente (en $T_0/\ln k$), on peut prouver que lorsque le nombre d'itérations tend vers l'infini, le processus ainsi construit se concentre sur les minimums globaux de la fonction d'énergie [GEMA84], [VANL87]. Dans la pratique, la température initiale T_0 est choisie interactivement, et la décroissance est souvent choisie *ad hoc* en fonction du nombre d'itérations dont on dispose (par exemple $T_{k+1} = cT_k$ où c est une constante positive inférieure à 1).

Remarquons que lorsque la température est élevée, l'algorithme peut accepter des accroissements importants de l'énergie, ce qui permet en particulier de sortir de puits d'énergie assez profonds. Inversement, lorsque la température décroît ces sauts en énergie deviennent de plus en plus rares et le système se stabilise au fond d'un puits.

ANNEXE C

Eléments de Morphologie Mathématique

La morphologie mathématique ne relève pas des domaines de l'analyse et de l'algèbre linéaire ou de la géométrie différentielle. Elle repose en revanche sur la théorie des ensembles et emprunte largement à la topologie, aux probabilités et à la théorie des graphes. Ainsi, elle a même suscité des développements sur les treillis booléens [MATH83], contribué à définir une topologie originale, la topologie en tout ou rien [MATH75] et permis une étude approfondie des ensembles fermés aléatoires [MATH69].

1 Morphologie mathématique appliquée aux images binaires

Le concept de forme, que l'on peut considérer comme étant à l'origine de la morphologie mathématique, assimilé à la notion de structure au sens mathématique du terme, a été défini par la connaissance d'une famille de relations considérées comme possibles et la donnée d'une famille d'ensembles géométriques arbitraires mais explicites, appelés *éléments structurants*.

D'un point de vue déterministe, caractériser la forme d'un objet, c'est savoir si, pour un élément structurant donné, chaque relation possible se trouve ou non vérifiée. Or si l'on s'en tient à ce principe général, on se trouve confronté à d'infinies possibilités.

La morphologie mathématique est donc connue pour ses relations ensemblistes « $S \subset X$ » ; « $S \cap X \neq \emptyset$ » où X est l'ensemble à analyser et S un élément structurant que l'on choisit en fonction des besoins de l'analyse.

Ces relations sont en effet, à la base des opérateurs morphologiques élémentaires que sont l'érosion et la dilatation.

Sur une image binaire, X pourra être l'ensemble des pixels blancs ou tout sous-ensemble de ce dernier, et S sous-ensemble quelconque de l'ensemble des sites.

On notera :

$$X_t = \{x + t, x \in X\}, \text{ le translaté de } X \text{ par } t \quad (\text{C-1})$$

$$\check{S} = \{-s, s \in S\}, \text{ le symétrique de } S \text{ par rapport à l'origine.}$$

1.1 Erosion et dilatation morphologiques

La soustraction de Minkowski de l'ensemble X par l'ensemble Y est définie par :

$$X \ominus Y = \bigcap_{y \in Y} X_y \quad (\text{C-2})$$

On appelle *érosion morphologique* par l'élément structurant S , la soustraction de Minkowski par $\overset{\vee}{S}$. On démontre que l'érodé de X par S est l'ensemble des sites x pour lesquels S , translaté par x , est inclus dans X (cf. figure C.1) :

$$X \ominus \overset{\vee}{S} = \{x, S_x \subset X\} \quad (\text{C-3})$$

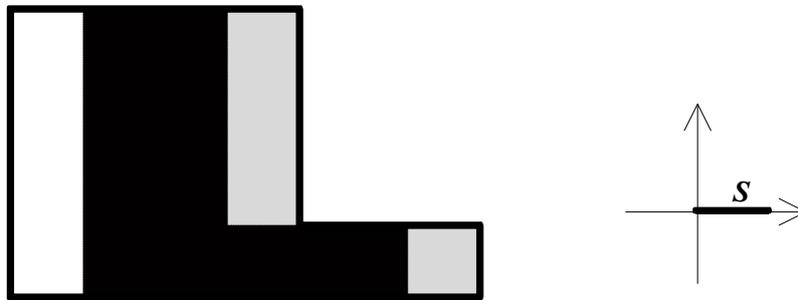


Figure C.1 : Erosion et dilatation par un élément structurant S non symétrique :

en gris clair X , hachuré $X \ominus \overset{\vee}{S}$, contours épais $X \oplus \overset{\vee}{S}$

La figure C.2 présente une érosion effectuée avec une maille hexagonale simulée sur un maillage carré (cf. chapitre III § 2.1.2) ; la « taille » de l'élément structurant correspond alors à la distance maximale de ses pixels à l'origine : taille 1 signifie 6 voisins, taille 2 signifie 18 voisins, etc.



image originale



érosion par un élément structurant pseudo-hexagonal de taille 1

Figure C.2 : Erosion morphologique

L'addition de Minkowski de l'ensemble X par l'ensemble Y est définie par :

$$X \oplus Y = \{x + y \mid x \in X, y \in Y\} \quad (\text{C-4})$$

On appelle *dilatation morphologique* par l'élément structurant S , l'addition de Minkowski par \check{S} . On démontre que le dilaté de X par S est l'ensemble des pixels x pour lesquels S , translaté par x , « intersecte » X (cf. figure C.1) :

$$X \oplus \check{S} = \{x, S_x \cap X \neq \emptyset\} \quad (\text{C-5})$$

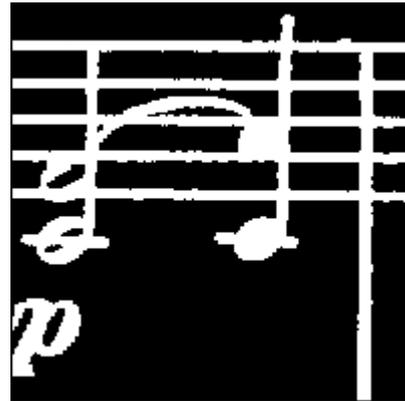
On notera la propriété d'inclusion suivante entre érodé et dilaté :

$$X \ominus \check{S} \subset X \subset X \oplus \check{S} \quad (\text{C-6})$$

Pour les développements sur les opérateurs morphologiques, leurs propriétés (croissance, anti-extensivité, idempotence...), toutes liées au fondement ensembliste, le lecteur pourra se reporter à [SERR82], [COST89].



image originale



dilatation par un élément structurant pseudo-hexagonal de taille 1

Figure C.3 : Dilatation morphologique

1.2 Ouverture et fermeture morphologiques

A partir de ces deux transformations de base, on peut élaborer deux transformations qui enchaînent une dilatation et une érosion opérées avec des éléments structurants symétriques par rapport à l'origine.

L'*ouverture morphologique* par l'élément structurant S est définie par :

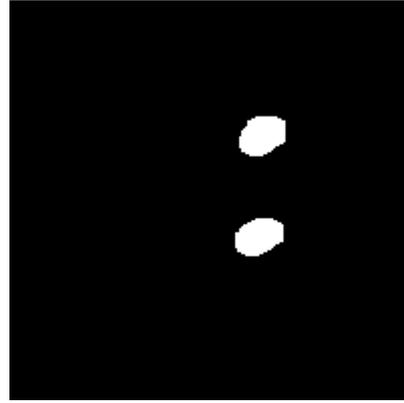
$$O_S(X) = (X \ominus \check{S}) \oplus S \quad (\text{C-7})$$

et la *fermeture morphologique* par S par :

$$F_S(X) = (X \oplus \check{S}) \ominus S \quad (\text{C-8})$$



fermeture par un élément structurant
pseudo-hexagonal de taille 2



ouverture par un élément structurant
pseudo-hexagonal de taille 5

Figure C.4 : Fermeture et ouverture morphologiques sur l'image originale de la figure C.2

La fermeture a pour effet de combler les trous, les chenaux étroits et de connecter les objets proches.

L'ouverture supprime les caps, les petits objets. Sur la figure C.4, toutes les petites structures ont disparu lors de l'érosion, les autres sont préservées, mais leurs contours ont été « nettoyés ».

Le fermé et l'ouvert d'un ensemble X sont liés par la relation d'inclusion suivante :

$$O_S(X) \subset X \subset F_S(X) \quad (C-9)$$

Ces deux transformations sont en outre idempotentes, c'est-à-dire :

$$O_S(O_S(X)) = O_S(X) \text{ et } F_S(F_S(X)) = F_S(X) \quad (C-10)$$

1.3 Amincissement et épaissement

Considérons des éléments structurants S formés de deux parties disjointes, notées S^0 et S^1 .

On note S_x^0 (resp. S_x^1) le translaté de S^0 (resp. S^1) par x .

On note $X \otimes S$ l'ensemble des pixels x pour lesquels S translaté par x vérifie :

- $\forall x_1 \in S_x^1, x_1 \in X$
- $\forall x_0 \in S_x^0, x_0 \notin X$

La transformation $X \otimes S$ est dénommée *transformation en tout ou rien*.

L'*amincissement* est la transformation, notée \ominus , qui consiste à supprimer d'un ensemble X l'ensemble $X \otimes S$:

$$X \ominus S = X \setminus X \otimes S \quad (C-11)$$

où « \setminus » représente la différence ensembliste.

Tout pixel ayant une configuration donnée (définie par l'élément structurant) est éliminé de X .

L'épaississement est la transformation, notée \odot , qui consiste à ajouter à un ensemble X l'ensemble $X \odot S$:

$$X \odot S = X \cup X \odot S \quad (\text{C-12})$$

Tout pixel ayant une configuration donnée est ajouté à X .

Il existe un très grand nombre d'éléments structurants, les plus utilisés sont répertoriés dans l'alphabet de Golay [GOLA69] [COST89].

Voici ci-dessous les éléments structurants L et E ; les pixels appartenant à S^0 sont notés 0, ceux appartenant à S^1 sont notés 1, les * sont des pixels qui ne sont pas pris en compte.

$$\begin{array}{ccc} & 1 & 1 \\ \text{L :} & * & 1 & * \\ & 0 & 0 \end{array} \qquad \begin{array}{ccc} & 1 & 1 & 1 \\ & * & 1 & * \\ & 0 & 0 & 0 \end{array}$$

$$\begin{array}{ccc} & * & * \\ \text{E :} & 0 & 1 & 0 \\ & 0 & 0 \end{array} \qquad \begin{array}{ccc} & * & * & * \\ & 0 & 1 & 0 \\ & 0 & 0 & 0 \end{array}$$

maille hexagonale

maille carrée 8-connexité

Certains de ces éléments conduisent à des transformations homotopiques, c'est-à-dire qui préservent le nombre de composantes connexes et le nombre de « trous » de chaque composante connexe. L'élément structurant L présenté ci-dessus est homotopique, alors que E ne l'est pas.

Les éléments structurants n'étant en général pas isotropes, si on veut opérer une transformation isotrope, on itère des amincissements ou des épaississements avec les 5 éléments structurants déduits de ceux de l'alphabet de Golay par rotation de $\pi/6$ en maille hexagonale (ou les 7 éléments déduits par rotation de $\pi/4$ en maille carrée).

Par exemple, l'amincissement itéré avec l'élément structurant L, dans toutes les directions jusqu'à stabilisation conduit au *squelette* homotopique de l'image (figure C.6) ; le squelette est sensible au bruit (irrégularités des formes), il est souvent suivi d'un *ébarbulage*, qui consiste à supprimer n pixels à chaque extrémité du squelette. Cette opération s'effectue par un amincissement du squelette, itéré dans toutes les directions n fois, avec l'élément structurant E.

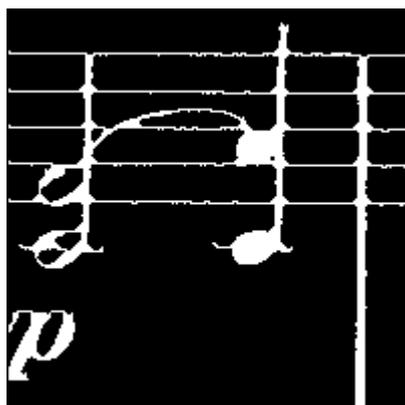


image originale

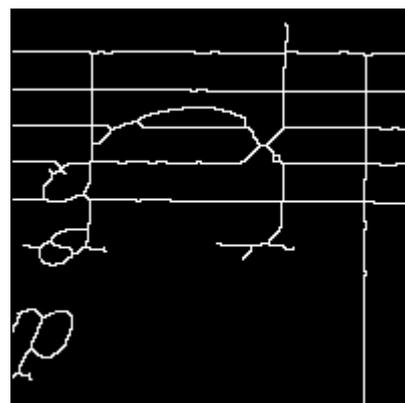


épaisissement avec L (maille carrée) itéré deux fois dans toutes les directions

Figure C.5 : Epaisseur morphologique



*amincissement avec l'élément L
(maille carrée)*



Squelette

*Figure C.6 : Amincissement morphologique homotopique
opéré sur l'image originale de la figure C.5*

2 Morphologie mathématique appliquée aux images en niveaux de gris

On se place dans le cadre des fonctions continues et on note Φ l'espace des fonctions continues de \mathbb{R}^n dans $\bar{\mathbb{R}}$.

La notion de sous-graphe permet d'établir une correspondance entre approche ensembliste et approche fonctionnelle.

Soit $f \in \Phi$; on appelle *sous-graphe* de la fonction f , le sous-ensemble de $\mathbb{R}^n \times \bar{\mathbb{R}}$ défini par : $SG(f) = \{ (x,t) , x \in \mathbb{R}^n, t \in \bar{\mathbb{R}}, t \leq f(x) \}$ (figure C.7)

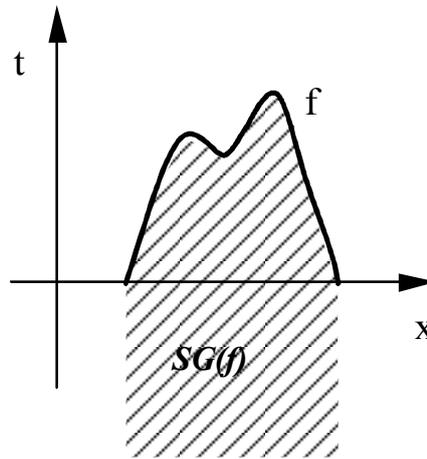


Figure C.7 : Exemple de sous-graphe (zone hachurée), pour $n=1$

2.1 Erosion morphologique

La *soustraction de Minkowski* de la fonction f de Φ par la fonction g de Φ est l'application de Φ dans Φ , notée $f \ominus g$ définie par :

$$\forall x \in \mathbb{R}^n, (f \ominus g)(x) = \inf \{ f(y) - g(x-y), y \in \mathbb{R}^n \} \quad (C-13)$$

(avec la convention $-\infty + \infty = -\infty$).

On appelle *érosion morphologique* par l'élément structurant g de Φ , la soustraction de Minkowski par $\underset{\vee}{g}$, où $\underset{\vee}{g}$ est définie par :

$$\forall x \in \mathbb{R}^n, \underset{\vee}{g}(x) = g(-x) \quad (C-14)$$

L'expression de l'érodé de f par $g \in \Phi$ est donnée par :

$$\forall x \in \mathbb{R}^n, (f \ominus^{\vee} g)(x) = \inf \{ f(y) - g(y-x), y \in \mathbb{R}^n \} \tag{C-15}$$

En terme de sous-graphe, l'érodé de f par g peut s'interpréter comme l'ensemble des positions du sous-graphe de g , pour lesquelles ce dernier est inclus dans le sous-graphe de f (cf. figure C.8), ce qui s'exprime mathématiquement par :

$$SG(f \ominus^{\vee} g) = \{ (x,t) \in \mathbb{R}^n \times \bar{\mathbb{R}}, (SG(g))_{(x,t)} \subset SG(f) \} \tag{C-16}$$

où l'indice indique une translation comme dans la définition C-1.

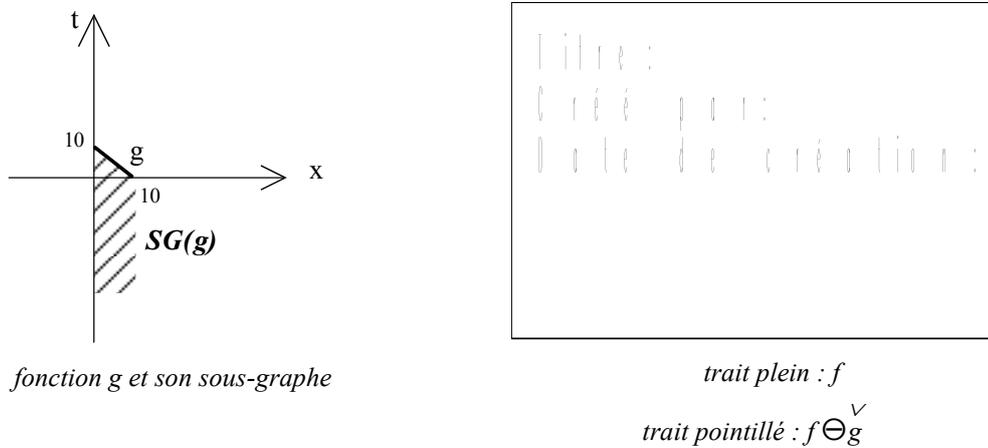


Figure C.8 : Erosion de f par g

Considérons comme élément structurant une fonction, notée g_S , associée à un compact¹¹ S de \mathbb{R}^n et définie par :

$$g_S(x) = \begin{cases} 0 & \text{si } x \in S \\ -\infty & \text{sinon} \end{cases} \tag{C-17}$$

Sous forme fonctionnelle, l'érodé de f par g_S s'exprime par :

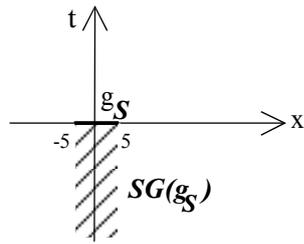
$$\forall x \in \mathbb{R}^n, (f \ominus^{\vee} g_S)(x) = \inf \{ f(y) - g_S(y-x), y \in \mathbb{R}^n \} \tag{C-18}$$

ou encore

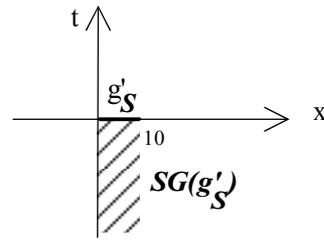
$$\forall x \in \mathbb{R}^n, (f \ominus^{\vee} g_S)(x) = \inf \{ f(y), y \in S_x \} \tag{C-19}$$

L'érosion d'une fonction par un élément structurant compact S plan (cf. figure C.9) consiste donc à effectuer l'« inf » sur le voisinage défini par S .

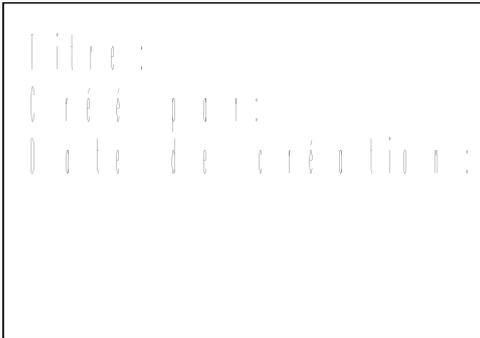
¹¹ un compact de \mathbb{R}^n est un sous-ensemble fermé borné de \mathbb{R}^n



fonction g_S associée à un compact centré



fonction g'_S associée à un compact non centré



trait plein : f
 trait pointillé : $f \ominus g_S$



trait plein : f
 trait pointillé : $f \ominus g'_S$

Figure C.9 : Erosions par des éléments structurants plans

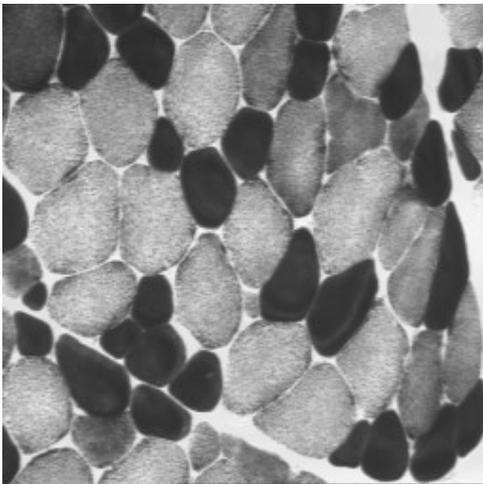
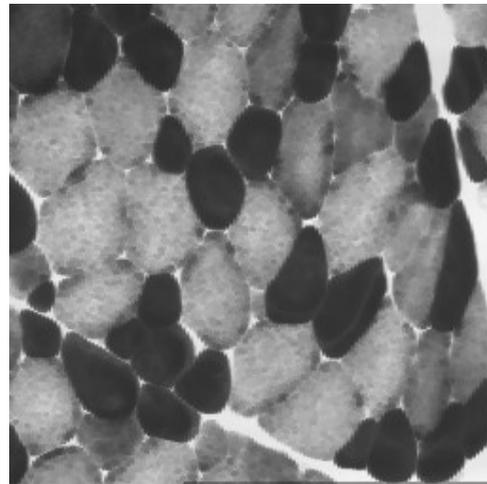


image originale



érosion (pseudo-hexagonale taille 2)

Figure C.10 : Erosion par un élément structurant plan centré

2.2 Dilatation morphologique

L'addition de Minkowski de deux fonctions f et g de Φ , est l'application notée $f \oplus g$ de Φ dans Φ , définie par :

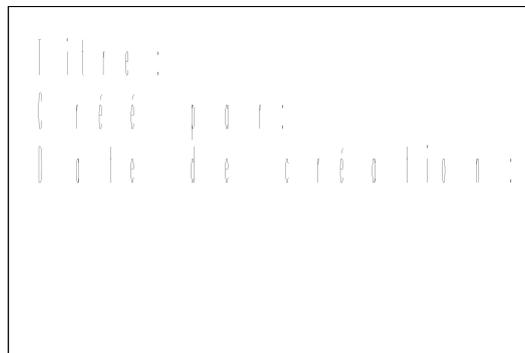
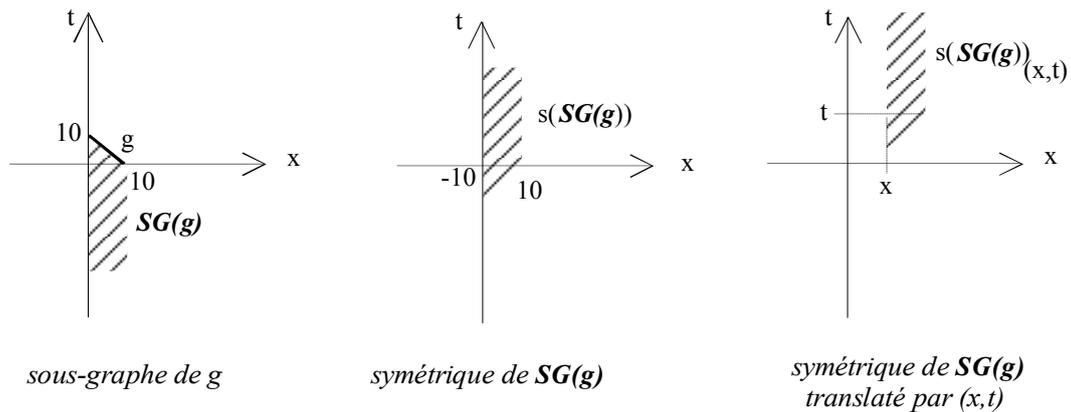
$$\forall x \in \mathbb{R}^n, (f \oplus g)(x) = \sup \{ f(y) + g(x-y), y \in \mathbb{R}^n \} \tag{C-20}$$

(avec la convention $+\infty - \infty = +\infty$)

La dilatation morphologique par l'élément structurant g , est égale à l'addition de Minkowski par $\overset{\vee}{g}$. La dilatation morphologique de f par $g \in \Phi$ est donc définie par :

$$\forall x \in \mathbb{R}^n, (f \oplus \overset{\vee}{g})(x) = \sup \{ f(y) + g(y-x), y \in \mathbb{R}^n \} \tag{C-21}$$

L'interprétation ensembliste (fig. C.2) du dilaté de f par g est l'ensemble des positions $(x,t) \in \mathbb{R}^n \times \bar{\mathbb{R}}$ pour lesquelles le symétrique par rapport à $\mathbb{R}^n \times \{0\}$ du sous-graphe de g , noté $s(SG(g))$, translaté par (x,t) , intersecte le sous-graphe de f .



trait plein : f
trait pointillé : $f \oplus \overset{\vee}{g}$

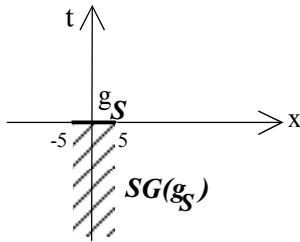
Figure C.11 : Dilatation de f par g

Si on particularise l'élément g en considérant l'élément structurant plan g_S , associé au compact S de \mathbb{R}^n , le dilaté de f par g_S s'écrit :

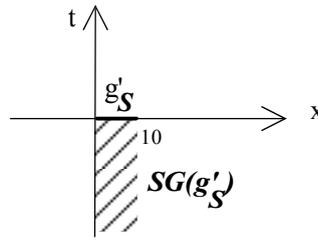
$$\forall x \in \mathbb{R}^n, \quad (f \oplus \underset{\vee}{g}_S)(x) = \sup \{ f(y) + g_S(y-x), y \in \mathbb{R}^n \}$$

ou encore :

$$\forall x \in \mathbb{R}^n, \quad (f \oplus \underset{\vee}{g}_S)(x) = \sup \{ f(y), y \in S_x \}$$



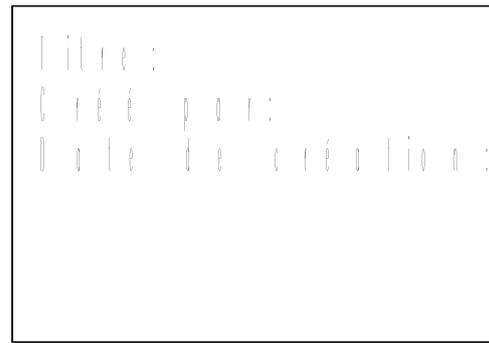
fonction g_S associée à un compact centré



fonction g'_S associée à un compact non centré



trait plein : f
trait pointillé : $f \oplus \underset{\vee}{g}_S$



trait plein : f
trait pointillé : $f \oplus \underset{\vee}{g}'_S$

Figure C.12 : Dilatations par des éléments structurants plans

Le dilaté et l'érodé d'une fonction sont liés par la relation suivante :

$$\forall x \in \mathbb{R}^n, \quad (f \ominus \underset{\vee}{g})(x) \leq f(x) \leq (f \oplus \underset{\vee}{g})(x) \tag{C-22}$$

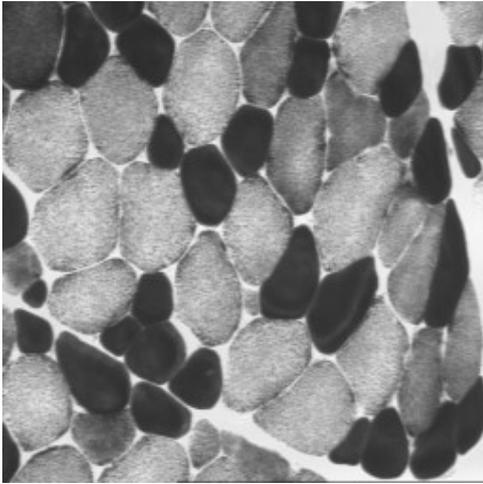
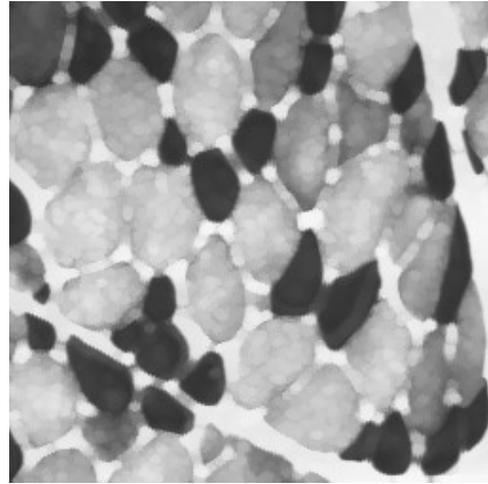


image originale



dilatation (pseudo-hexagonale taille 2)

Figure C.13 : Dilatation par un élément structurant plan centré

2.3 Ouverture, fermeture morphologiques

A partir des transformations de base que sont l'érosion et la dilatation morphologiques, on peut créer de nouvelles transformations : l'ouverture et la fermeture morphologiques.

Soit $g \in \Phi$. On appelle *ouverture morphologique* par l'élément structurant g , l'application de Φ dans Φ notée O_g , définie par :

$$\forall f \in \Phi, O_g(f) = (f \ominus \overset{\vee}{g}) \oplus g \quad (\text{C-23})$$

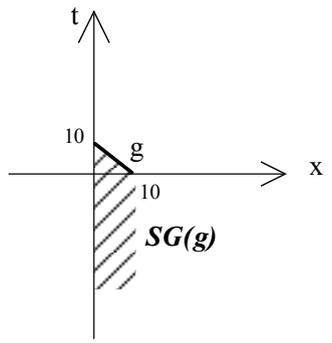
On appelle *fermeture morphologique* par l'élément structurant g , l'application de Φ dans Φ notée F_g , définie par :

$$\forall f \in \Phi, F_g(f) = (f \oplus \overset{\vee}{g}) \ominus g \quad (\text{C-24})$$

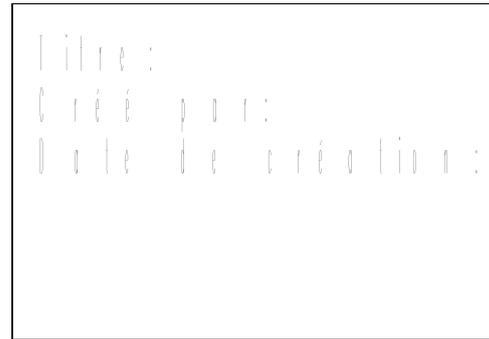
Les interprétations géométriques de l'ouverture et de la fermeture morphologiques sont données par la proposition suivante :

Soient $f \in \Phi$ et $g \in \Phi$. On a :

$$SG(O_g(f)) = \bigcup_{(x,t) \in R^n \times \bar{R}} \{SG(g)_{(x,t)}, SG(g)_{(x,t)} \subset SG(f)\} \quad (\text{C-25})$$

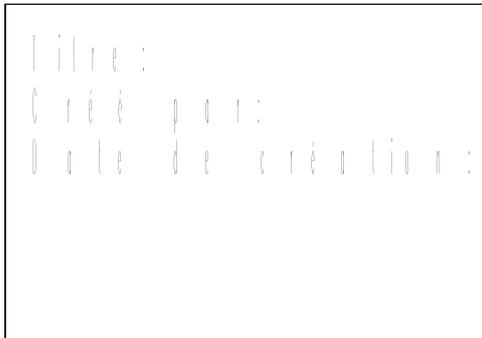


fonction g et son sous-graphe



trait plein : f
trait pointillé : $O_g(f)$

Figure C.14 : Ouverture morphologique de f par g



trait plein : f
trait pointillé : ouverture par g_S



trait plein : f
trait pointillé : fermeture par g_S

Figure C.15 : Ouverture et fermeture morphologiques par un compact centré (représenté figure C.9)

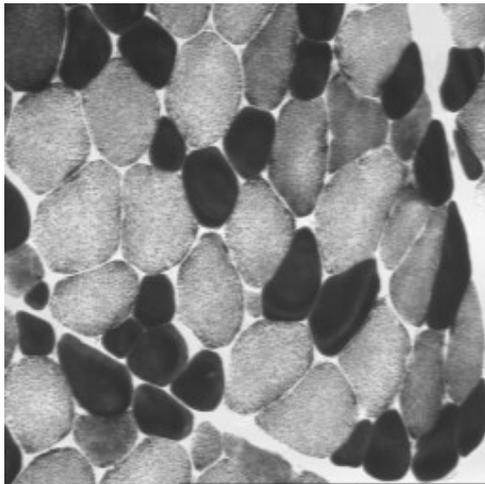
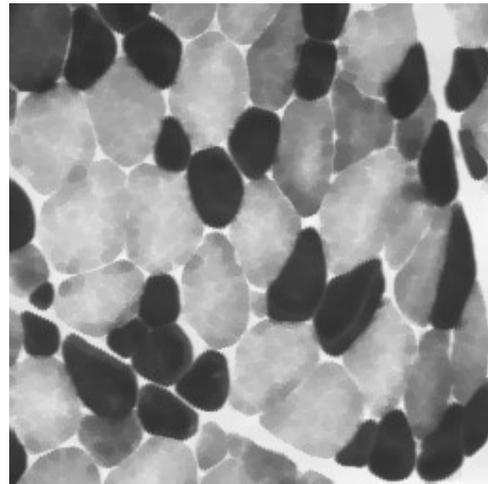
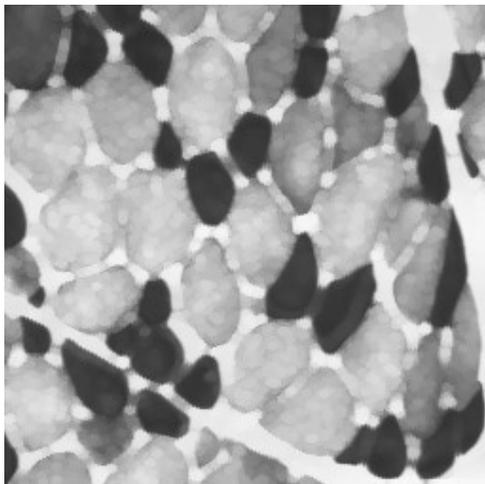
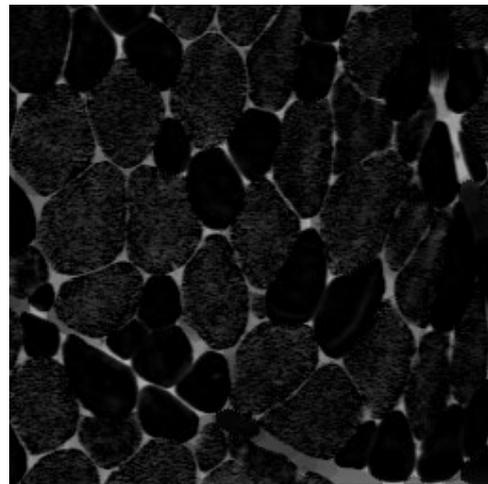
L'ouverture et la fermeture morphologiques sont des fonctions idempotentes :

$$O_g(O_g(f)) = O_g(f) \text{ et } F_g(F_g(f)) = F_g(f) \tag{C-26}$$

et qui vérifient la relation :

$$\forall x \in \mathbb{R}^n, O_g(f) \leq f \leq F_g(f) \tag{C-27}$$

Le sous-graphe de l'ouvert $O_g(f)$ est le domaine balayé par le sous-graphe de l'élément structurant g lorsqu'il reste inclus dans le sous-graphe de la fonction f .

*image originale**fermeture(pseudo-hexagonale taille 2)**ouverture (pseudo-hexagonale taille 2)**chapeau haut-de-forme
(masque hexagonal taille 5)**Figure C.16 : Fermeture, ouverture par des éléments structurants plans centrés*

Chapeau haut-de-forme

Une application intéressante de la fermeture est la transformation pittoresquement dénommée chapeau haut-de-forme et qui résulte de la soustraction de l'image initiale à l'image obtenue par fermeture :

$$\forall f \in \Phi, F_g(f) - f \quad (\text{C-28})$$

On ne conserve donc de l'image initiale que les pics ou les crêtes qui « entrent » (au moins dans l'une des directions) dans le chapeau, supposé de hauteur infinie, et dont la base est l'élément structurant. Sur la figure C.15, on ne conserverait que les sommets des deux pics. L'intérêt de cette transformation est d'extraire les zones claires de l'image, en fonction de critères de forme (définis par

l'élément structurant) et du niveau de gris relatif des pixels par rapport à leur voisinage.

2.4 Amincissement et épaissement

Considérons des éléments structurants \mathcal{S} contenant deux parties disjointes, notées \mathcal{S}^0 et \mathcal{S}^1 .

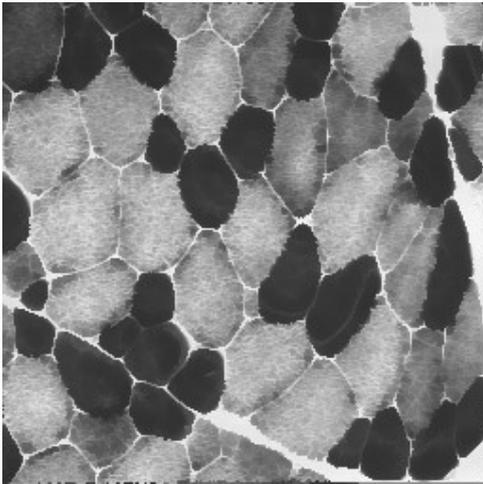
On note \mathcal{S}_x^0 (resp. \mathcal{S}_x^1) le translaté de \mathcal{S}^0 (resp. \mathcal{S}^1) par x .

On appelle *amincissement* de f par $g_{\mathcal{S}}$, la transformation définie par :

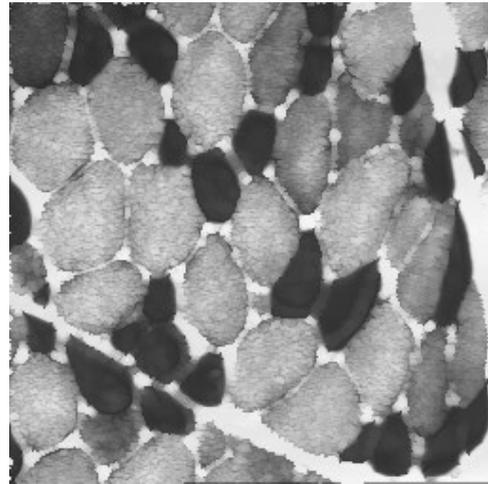
$$f \ominus g_{\mathcal{S}} = \begin{cases} \sup\{f(y), y \in \mathcal{S}_x^0\} & \text{si } \sup\{f(y), y \in \mathcal{S}_x^0\} < f(x) \leq \inf\{f(y), y \in \mathcal{S}_x^1\} \\ f(x) & \text{sinon} \end{cases}$$

On appelle *épaissement* de f par $g_{\mathcal{S}}$, la transformation définie par :

$$f \odot g_{\mathcal{S}} = \begin{cases} \inf\{f(y), y \in \mathcal{S}_x^1\} & \text{si } \sup\{f(y), y \in \mathcal{S}_x^0\} \leq f(x) < \inf\{f(y), y \in \mathcal{S}_x^1\} \\ f(x) & \text{sinon} \end{cases}$$



*amincissement maille carrée 2 itérations
dans chacune des 8 directions*



*épaissement maille carrée 2 itérations
dans chacune des 8 directions*

Figure C.17 : Amincissement et épaissement par l'élément structurant L sur l'image initiale de la figure C.16

ANNEXE D

Rappels de Traitement du signal

1 Transformée de Fourier

1.1 Définition

Soit un signal continu $A(x)$, sa transformée de Fourier ou spectre $\hat{A}(u)$ est définie par :

$$\text{TF } A(u) = \hat{A}(u) = \int_{-\infty}^{+\infty} A(x) \exp(-2j\pi ux) dx \quad (\text{D-1})$$

Si $A(x)$ est définie dans $L^1(\mathbb{R})$: l'espace des fonctions absolument intégrables, on a les propriétés suivantes :

- $\text{TF } A(u)$ est une fonction continue et bornée sur \mathbb{R} ,
- TF est un opérateur linéaire et continu de $L^1(\mathbb{R})$ dans $L^\infty(\mathbb{R})$,

$$\lim_{u \rightarrow \infty} \left| \hat{f}(u) \right| = 0$$

La transformée inverse est donnée par :

$$\text{TF}^{-1} A(x) = \int_{-\infty}^{+\infty} \hat{A}(u) \exp(2j\pi ux) du \quad (\text{D-2})$$

1.2 Translation

L'opérateur translation, noté \mathbf{r}_a est tel que : $\mathbf{r}_a A(x) = A(x-a)$. Nous avons alors la relation suivante :

$$\text{TF}\{\mathbf{r}_a A(x)\}(u) = \int_{-\infty}^{+\infty} A(x-a) \exp(2j\pi ux) dx = \exp(-2j\pi ua) \hat{A}(u) \quad (\text{D-3})$$

1.3 Dérivation

Pour un signal $A(x)$ dérivable, nous avons :

$$\frac{dA}{dx}(x) \xrightarrow{\text{TF}} 2j\pi u \hat{A}(u) \quad (\text{D-4})$$

1.4 Théorème de convolution

La transformée de Fourier du produit de convolution :

$$B(x) = A * h(x) = \int_{-\infty}^{+\infty} A(t)h(x-t)dt \quad (\text{D-5})$$

où h représente, par exemple, une réponse impulsionnelle de filtre, est donnée par :

$$\hat{B}(u) = \text{TF}\{A * h(x)\}(u) = \int_{-\infty}^{+\infty} \exp(-2j\pi ux) \left(\int_{-\infty}^{+\infty} A(t)h(x-t)dt \right) dx \quad (\text{D-6})$$

$$\begin{aligned} \hat{B}(u) &= \int_{-\infty}^{+\infty} A(t) \left(\int_{-\infty}^{+\infty} \exp(-2j\pi ux) h(x-t) dx \right) dt \\ &= \int_{-\infty}^{+\infty} A(t) \exp(-2j\pi ut) \hat{h}(u) dt \end{aligned} \quad (\text{D-7})$$

$$\hat{B}(u) = \hat{A}(u) \cdot \hat{h}(u) \quad (\text{D-8})$$

1.5 Transformée de Fourier d'une distribution

Soit $S(\mathbb{R})$ l'espace des fonctions f de \mathbb{R} dans C telles que :

- f est indéfiniment dérivable,
- f et toutes ses dérivées sont à décroissance rapide

Pour obtenir l'expression de la transformée de Fourier d'une distribution, il faut définir l'espace $S'(\mathbb{R})$ des distributions tempérées qui est l'espace vectoriel des formes linéaires T définies et continues sur $S(\mathbb{R})$.

On démontre par ailleurs que toute fonction f à croissance lente est une distribution tempérée.

Pour T appartenant à l'espace $S'(\mathbb{R})$, on a alors :

$$\begin{aligned} T_{\hat{f}}(\varphi) &= \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} f(x) \exp(-2j\pi ux) dx \right) \varphi(u) du = \\ &= \int_{-\infty}^{+\infty} f(x) \left(\int_{-\infty}^{+\infty} \exp(-2j\pi ux) \varphi(u) du \right) dx = T_f(\hat{\varphi}) \end{aligned} \quad (\text{D-9})$$

On note souvent $\hat{T}_f(\varphi)$ la transformée de Fourier d'une distribution,
 $\hat{T}_f(\varphi) = T_f(\varphi)$

1.5.1 Transformée de Fourier de la distribution de Dirac

$$\hat{\delta}(\varphi) = \delta(\hat{\varphi}) = \hat{\varphi}(0) = \int_{-\infty}^{+\infty} \varphi(u) du = T_1(\varphi)$$

$$\text{d'où : } \delta \xleftarrow{\text{TF}} 1 \quad (\text{D-10})$$

$$\text{Egalement : } \hat{\delta}_a(\varphi) = \int_{-\infty}^{+\infty} \hat{\delta}_a \varphi(x) dx$$

$$\hat{\delta}_a(\varphi) = \delta_a(\hat{\varphi}) = \int_{-\infty}^{+\infty} \exp(-2j\pi u a) \varphi(u) du \quad (\text{D-11})$$

$$\text{d'où : } \delta_a \xleftarrow{\text{TF}} \exp(-2j\pi u a) \quad (\text{D-12})$$

1.5.2 Transformée de Fourier d'un signal sinusoïdal

Soit un signal sinusoïdal $f(x) = \exp(2j\pi u_0 x)$ qui peut être considéré comme une distribution tempérée : $f \in L^\infty(\mathbb{R})$. Prenons $\varphi \in \mathcal{S}(\mathbb{R})$ espace vectoriel de fonctions à décroissance rapide. Nous avons :

$$T_f(\varphi) = T_f(\hat{\varphi}) = \int_{-\infty}^{+\infty} \exp(2j\pi u_0 x) \hat{\varphi}(x) dx = \text{TF}^{-1} \hat{\varphi}(u_0) = \varphi(u_0) \quad (\text{D-13})$$

$$\text{donc : } \exp(2j\pi u_0 x) \xleftarrow{\text{TF}} \delta_{u_0} \quad (\text{D-14})$$

$$\text{et en particulier : } 1 \xleftarrow{\text{TF}} \delta(u) \quad (\text{D-15})$$

1.6 Développement en série de Fourier de la fonction peigne

Le calcul s'effectue à partir de la fonction « dent de scie » $f(x)$ périodique de période τ qui vaut $\frac{x}{\tau}$ sur $[0, \tau[$.

La dérivée au sens des distributions de $f(x)$ vaut :

$$f'(x) = \frac{1}{\tau} - \sum_{n=-\infty}^{+\infty} \delta_{n\tau}(\varphi) \quad (\text{D-16})$$

Le développement en série de Fourier de $f(x)$ a pour expression :

$$f(x) = \frac{1}{2} + \frac{j}{2\pi} \sum_{n \neq 0} \frac{1}{n} \exp\left(2j\pi n \frac{x}{\tau}\right) \quad (\text{D-17})$$

En dérivant l'expression ci-dessus, on obtient une autre forme de $f'(x)$:

$$f'(x) = -\frac{1}{\tau} \sum_{n \neq 0} \exp\left(2j\pi n \frac{x}{\tau}\right) \quad (\text{D-18})$$

A partir des deux relations (D-16) (D-18), on obtient l'expression de la distribution peigne Π_τ :

$$-\frac{1}{\tau} \sum_{n \neq 0} \exp\left(2j\pi n \frac{x}{\tau}\right) = \frac{1}{\tau} - \sum_{n=-\infty}^{+\infty} \delta_{n\tau}(\varphi)$$

soit :

$$\Pi_\tau = \sum_{n=-\infty}^{+\infty} \delta_{n\tau} = \frac{1}{\tau} \sum_{n=-\infty}^{+\infty} \exp\left(2j\pi n \frac{x}{\tau}\right) \quad (\text{D-19})$$

La série de droite est divergente au sens des fonctions mais convergente au sens des distributions vers le peigne de Dirac dont elle constitue le développement en série de Fourier.

En exploitant la relation (D-12) on calcule la transformée de Fourier du peigne :

$$\Pi_\tau = \sum_{n=-\infty}^{+\infty} \delta_{n\tau} \xrightarrow{\text{TF}} \sum_{n=-\infty}^{+\infty} \hat{\delta}_{n\tau} = \sum_{n=-\infty}^{+\infty} \exp(-2j\pi n \tau) \quad (\text{D-20})$$

Par ailleurs, avec (D-19) et (D-14), on peut aussi écrire que :

$$\hat{\Pi}_\tau = \frac{1}{\tau} \sum_{n=-\infty}^{+\infty} \delta_{\frac{n}{\tau}} = \frac{1}{\tau} \Pi_{\frac{1}{\tau}} \quad (\text{D-21})$$

C'est à dire que la transformée de Fourier d'un peigne de Dirac de période τ est, au coefficient $1/\tau$ près, un peigne de Dirac de période $1/\tau$.

2 Echantillonnage

2.1 Formule de Poisson

La formule de Poisson établit le lien entre série de Fourier et transformée de Fourier :

$$\sum_{n=-\infty}^{+\infty} f(x - n\tau) = \frac{1}{\tau} \sum_{n=-\infty}^{+\infty} \hat{f}\left(\frac{n}{\tau}\right) \exp\left(-2j\pi n \frac{x}{\tau}\right)$$

et la forme duale :

$$\sum_{n=-\infty}^{+\infty} \hat{g}\left(u - \frac{n}{\tau}\right) = \tau \sum_{n=-\infty}^{+\infty} g(n\tau) \exp(-2j\pi n \tau u) \quad (\text{D-22})$$

Elle est valable pour toute distribution f appartenant à l'ensemble E' des distributions à support compact. La formule duale est vraie pour toute fonction g telle que $\hat{g} \in E'$.

2.2 Spectre d'une fonction échantillonnée

Une manière d'exprimer les conditions d'applicabilité de la formule de Poisson à un signal $A(x)$ revient à dire qu'il doit être à bande limitée, c'est à dire que :

$$\text{Supp } \hat{A} \subset -u_c, u_c$$

L'expression d'un signal échantillonné est de la forme :

$$A\Pi_\tau = \sum_{n=-\infty}^{+\infty} A(n\tau)\delta_{n\tau} \quad (\text{D-23})$$

Il s'ensuit que :

$$\text{TF}\{A\Pi_\tau\} = \hat{A} * \hat{\Pi}_\tau = \frac{1}{\tau} \cdot \hat{A} * \Pi_{\frac{1}{\tau}} \quad (\text{D-24})$$

En appliquant la formule de Poisson, on obtient :

$$\tau \text{TF}\{A\Pi_\tau\} = \sum_{n=-\infty}^{+\infty} \hat{A}\left(u - \frac{n}{\tau}\right) = \tau \sum_{n=-\infty}^{+\infty} A(n\tau) \exp(-2j\pi n \tau u) \quad (\text{D-25})$$

Les relations précédentes montrent donc que :

- le spectre du signal correspondant à l'échantillonnage toutes les τ secondes d'un signal $A(x)$ à bande limitée est périodique de période $1/\tau$,
- le spectre est la somme de tous les translatés d'un multiple de $1/\tau$ du spectre de A .
- suivant la valeur de τ , deux cas se présentent :

$$\tau > \frac{1}{2u_c} \text{ il y a recouvrement des spectres périodiques,}$$

$$\tau < \frac{1}{2u_c} \text{ les spectres périodiques ne se recouvrent pas. Le spectre du}$$

signal échantillonné est une reproduction périodique de \hat{A}

2.3 Théorème de l'échantillonnage, formule de Shannon

En analysant la formule de Poisson vraie dans S' :

$$\sum_{n=-\infty}^{+\infty} \hat{A}\left(u - \frac{n}{\tau}\right) = \tau \sum_{n=-\infty}^{+\infty} A(n\tau) \exp(-2j\pi n \tau u) \quad (\text{D-26})$$

on peut dire que reconstruire $A(x)$ à partir des valeurs des échantillons consiste à isoler le lobe central du spectre périodique pour reconstruire A qui s'exprimera en fonctions des valeurs $A(n\tau)$.

Le terme de gauche de l'équation ci-dessus admet un développement en série de Fourier égal à :

$$B(u) = \sum_{n=-\infty}^{+\infty} c_n \exp(2j\pi n u) \quad (\text{D-27})$$

Compte tenu de l'unicité du développement et de la relation (D-27), il vient que :

$$c_{-n} = \tau A(n\tau)$$

Si on multiplie la fonction $B(u)$ par la fonction fenêtre $\Psi_{\frac{1}{2\tau}}$ nulle en dehors

de $\left[-\frac{1}{2\tau}, \frac{1}{2\tau}\right]$ nous obtenons :

$$\hat{A}(u) = \tau \sum_{n=-\infty}^{+\infty} A(n\tau) \Psi(u) \exp(-2j\pi n u) \quad (\text{D-28})$$

Or :

$$\text{TF}^{-1}\{\Psi(u) \exp(-2j\pi n u)\}(x) = \text{TF}^{-1}\{\Psi(u)\}(x - n\tau) = \frac{\sin \frac{\pi}{\tau}(x - n\tau)}{\pi(x - n\tau)} \quad (\text{D-29})$$

On obtient alors la formule d'interpolation :

$$A(x) = \sum_{n=-\infty}^{+\infty} A(n\tau) \frac{\sin \frac{\pi}{\tau}(x - n\tau)}{\frac{\pi}{\tau}(x - n\tau)} \quad (\text{D-30})$$

Cette formule permet de reconstruire le signal $A(x)$ à partir de ses échantillons pris à une période $\tau \leq \frac{1}{2u_c}$ où u_c est la bande passante du signal.

3 Processus aléatoires

Une variable aléatoire A est une règle d'affectation d'un nombre $A(\zeta)$ à chaque réalisation ζ d'une expérience donnée. Un processus aléatoire ou stochastique $A(x)$ est une règle d'affectation d'une fonction $A(x, \zeta)$ à chaque réalisation ζ d'une expérience. On utilise de façon classique la notation $A(x)$ pour un processus aléatoire, on ne mentionne pas la dépendance avec ζ .

Un signal $A(x)$ peut être considéré comme la réalisation ζ d'une épreuve, c'est alors une fonction $A(x, \zeta)$ de la variable x . On utilise le terme de signal aléatoire.

Si x est fixé, $A(x, \zeta)$ est une variable aléatoire associée aux réalisations d'une expérience.

Par exemple, la numérisation de n photographies de tissu prises dans les mêmes conditions fournit n images différentes. Chaque image \mathbf{a} est la réalisation ζ

d'un processus stochastique \mathbf{A} . Le niveau de gris a_s d'un pixel est une réalisation de la variable aléatoire $A[s, \zeta]$ notée aussi A_s .

3.1 Statistiques des processus aléatoires

$A(x)$ est un processus stochastique. On définit les statistiques suivantes :

Moyenne ou moment d'ordre 1 :

$$\mu(x) = E\{A(x)\} = \int_{-\infty}^{\infty} a \cdot f(a, x) da \quad (D-31)$$

$$\text{avec} \begin{cases} f(a, x) = \frac{\partial F}{\partial a}(a, x) \\ F(a, x) = P(A(x) \leq a) \end{cases}$$

Autocorrélation :

$$R(x_1, x_2) = E\{A(x_1)A^*(x_2)\} \quad (D-32)$$

$A^*(.)$ représente le complexe conjugué.

Puissance moyenne :

$$R(x, x) = E\{|A(x)|^2\} \quad (D-33)$$

Autocovariance :

$$C(x_1, x_2) = R(x_1, x_2) - \mu(x_1)\mu^*(x_2) \quad (D-34)$$

Variance :

$$C(x, x) = E\{|A(x)|^2\} - |\mu(x)|^2 \quad (D-35)$$

Intercorrélation :

Pour deux processus $A(x)$ et $B(x)$, l'intercorrélation est définie par :

$$R_{AB}(x_1, x_2) = E\{A(x_1)B^*(x_2)\} \quad (D-36)$$

3.2 Relation d'incertitude

On appelle dispersion spatiale d'énergie d'un signal $H(x)$, la grandeur :

$$\sigma_H^2 = \int_{-\infty}^{+\infty} x^2 |H(x)|^2 dx \quad (D-37)$$

De même, on exprime la dispersion fréquentielle d'énergie par :

$$\sigma_{\hat{H}}^2 = \int_{-\infty}^{+\infty} u^2 |\hat{H}(u)|^2 du \quad (D-38)$$

Sachant que l'énergie du signal $H(x)$ est donnée par :

$$E_H = \int_{-\infty}^{+\infty} H(x)H^*(x)dx \quad (D-39)$$

on définit respectivement la largeur spatiale utile et la largeur fréquentielle utile par :

$$\Delta x^2 = \frac{\sigma_H^2}{E_H} \quad \text{et} \quad \Delta u^2 = \frac{\sigma_{\hat{H}}^2}{E_H} \quad (\text{D-40})$$

On démontre qu'on ne peut pas avoir de précision arbitraire à la fois sur la localisation spatiale d'un signal et sur sa largeur spectrale ; cette propriété est traduite par la relation d'incertitude :

$$\sigma_H \sigma_{\hat{H}} \geq \frac{E_H}{4\pi} \quad (\text{D-41})$$

L'impulsion gaussienne est le signal qui réalise le meilleur compromis entre précision fréquentielle et spatiale. Pour une largeur fréquentielle utile Δu , le signal :

$$H(x) = \alpha \exp\left(- (2\pi\Delta u)^2 x^2\right) \quad (\text{D-42})$$

minimise la largeur spatiale.

3.3 Processus stationnaires

Un processus stationnaire $A(x)$ est du premier ordre si :

$$\mu = E\{A(x)\} \text{ est indépendant de } x.$$

Il est stationnaire du deuxième ordre si, en plus :

$$R_{AA}(x_1, x_2) = R_{AA}(\tau) = E\{A(x+\tau)A^*(x)\}, \text{ avec } x_1 - x_2 = \tau$$

$$\text{Il s'ensuit que : } R_{AA}(0) = E\{|A(x)|^2\} \quad (\text{D-43})$$

$$\text{et : } C_{AA}(\tau) = R_{AA}(\tau) - \mu^2 \quad (\text{D-44})$$

La densité spectrale est la transformée de Fourier de la fonction d'autocorrélation :

$$S_{AA}(u) = \hat{R}_{AA}(u) = \int_{-\infty}^{+\infty} R_{AA}(x) \exp(-2j\pi ux) dx \quad (\text{D-45})$$

$$\text{et } R_{AA}(\tau) = \text{TF}^{-1} S_{AA} = \int_{-\infty}^{+\infty} S_{AA}(u) \exp(2j\pi u\tau) du \quad (\text{D-46})$$

4 Filtre

Un filtre linéaire est un système continu et stationnaire qui établit une réponse de convolution entre le signal d'entrée $A(x)$ et le signal de sortie $B(x)$:

$$B(x) = A * h(x) = \int_{-\infty}^{+\infty} A(t) h(x-t) dt \quad (\text{D-47})$$

$h(x)$ est la réponse impulsionnelle du filtre.

4.1 Filtrage d'un signal aléatoire

Soit un filtre linéaire tel que $B(x)=A(x)*h(x)$, on peut écrire que :

$$E\{B(x)\} = E\{A * h(x)\} = E\{A(x)\} * h(x) \quad (D-48)$$

Le calcul du spectre du signal de sortie $B(x)$ quand $A(x)$ est stationnaire s'effectue en exploitant les relations suivantes :

$$R_{AB}(x_1, x_2) = R_{AA} * h^*(x_2) \quad (D-49)$$

$$R_{BB}(x_1, x_2) = R_{AB} * h(x_1) \quad (D-50)$$

On montre alors que :

$$R_{BB}(\tau) = [h^*(-\tau) * [h * R_{AA}(\tau)]] \quad (D-51)$$

Il s'ensuit que :

$$\hat{R}_{BB}(u) = |\hat{h}(u)|^2 \cdot \hat{R}_{AA}(u)$$

soit :

$$S_{BB}(u) = |\hat{h}(u)|^2 \cdot S_{AA}(u) \quad (D-52)$$

$$R_{BB}(\tau) = \int_{-\infty}^{+\infty} S_{BB}(u) \exp(2\pi j u \tau) du \quad (D-53)$$

$$R_{BB}(0) = E\{B(x)^2\} = \int_{-\infty}^{+\infty} S_{BB}(u) du = \int_{-\infty}^{+\infty} |\hat{h}(u)|^2 \cdot S_{AA}(u) du \quad (D-54)$$

4.2 Filtre adapté

Le but est d'optimiser la détection d'un signal déterministe en présence d'un bruit additif stationnaire. Pour cela, il faut considérer un critère qui caractérisera la qualité de la détection. Le filtre sera alors optimal au sens de ce critère. Quand le critère choisi est le rapport signal sur bruit, on parle de filtre adapté.

Soit un filtre de réponse impulsionnelle $h(x)$ auquel on applique un signal d'entrée $B(x)$ qui est la somme d'un signal connu $A(x)$ et d'un bruit $N(x)$. Le signal de sortie $D(x)$ vaut :

$$D(x) = B * h(x)$$

$$B(x) = A(x) + N(x)$$

$$D(x) = A * h(x) + N * h(x)$$

Le rapport signal à bruit (RSB) est défini comme le rapport de la contribution du signal seul à la racine carrée de la puissance du bruit :

$$RSB^2 = \frac{|A * h(x)|^2}{E\{N * h(x)\}^2} \quad (D-55)$$

En posant $I(x) = N(x) * h(x)$, on peut écrire que :

$$S_{II}(u) = |\hat{h}(u)|^2 \cdot S_{NN}(u) \quad (D-56)$$

avec :

$$S_{NN}(u) = \int_{-\infty}^{+\infty} R_{NN}(\tau) \exp(-j2\pi u \tau) d\tau$$

Si $N(x)$ est un bruit blanc stationnaire :

$$R_{NN}(\tau) = N_0^2 \delta(\tau) \text{ et } S_{NN} = N_0^2 \quad (D-57)$$

Il vient que :

$$E\{I(x)^2\} = \int_{-\infty}^{+\infty} |\hat{h}(u)|^2 \cdot S_{NN}(u) du = N_0^2 \int_{-\infty}^{+\infty} |\hat{h}(u)|^2 du \quad (D-58)$$

puis :

$$A * h(x) = \text{TF}^{-1}\{\hat{A}\hat{h}\}(x) = \int_{-\infty}^{+\infty} \hat{A}(u) \hat{h}(u) \exp(2j\pi ux) du \quad (D-59)$$

Le problème est de trouver $\hat{h}(u)$ qui maximise le rapport signal à bruit en une position x_0 .

Pour cela, on applique l'inégalité de Schwartz et à partir de l'expression (D-55) on peut écrire :

$$RSB^2 \leq \frac{\int_{-\infty}^{+\infty} |\hat{A}(u) \exp(2j\pi ux)|^2 du \int_{-\infty}^{+\infty} |\hat{h}(u)|^2 du}{N_0^2 \int_{-\infty}^{+\infty} |\hat{h}(u)|^2 du} \quad (D-60)$$

$$RSB^2 \leq \frac{1}{N_0^2} \int_{-\infty}^{+\infty} |\hat{A}(u)|^2 du \quad (D-61)$$

L'expression de $\hat{h}(u)$ qui donne un RSB maximum en une position x_0 est donnée par :

$$\hat{h}(u) = k(\hat{A}(u) \exp(2j\pi ux_0))^* \quad (D-62)$$

soit :

$$\hat{h}(u) = k(\hat{A}(u) \exp(-2j\pi ux_0)) \quad (D-63)$$

ce qui signifie que :

$$h(x) = kA(x_0 - x) \quad (D-64)$$

La réponse impulsionnelle du filtre adapté en x_0 est donc égale au symétrisé du signal retardé de x_0 . La sortie du filtre adapté a pour expression :

$$D(x) = \int_{-\infty}^{+\infty} B(t)h(x-t)dt = \int_{-\infty}^{+\infty} B(t)B(x_0 - x + t)dt \quad (D-65)$$

Pour faire un filtrage adapté, on effectue donc une corrélation avec le symétrisé du signal décalé de x_0 .

4.3 Filtre dérivateur

Soit un signal continu $A(x)$, le problème est de trouver la réponse impulsionnelle $h(x)$ du filtre dérivateur. Nous avons :

$B(x) = A(x)*h(x)$ et on souhaite que $B(x) = A'(x)$ donc :

$$h(x)*A(x) = A'(x)$$

Au sens des distributions :

$$A'(x) = \delta*A'(x) = \delta'*A(x)$$

Il s'ensuit que :

$$h(x) = \delta' \quad (D-66)$$

Ce filtre est réalisable, mais instable.

Pour un signal échantillonné :

$$A(x)\Pi_\tau = \sum_{n=-\infty}^{+\infty} A(n\tau)\delta_{n\tau} \quad (D-67)$$

La sortie du filtre dérivateur est donnée par :

$$B(x) = A(x).\Pi_\tau * \delta' \quad (D-68)$$

En prenant la transformée de Fourier de chaque membre de l'équation (D-68), il vient :

$$TFB(u) = TF\{A(x)\Pi_\tau\}.TF\delta' \quad (D-69)$$

$$TFB(u) = (\hat{A}(u)*\hat{\Pi}_\tau)2j\pi u TF\delta \quad (D-70)$$

$$TFB(u) = \frac{1}{\tau} \left(\hat{A}(u)*\Pi_{\frac{1}{\tau}} \right) .2j\pi u \quad (D-71)$$

$$TFB(u) = \frac{1}{\tau} .2j\pi u . \sum_{n=-\infty}^{\infty} \hat{A}\left(u - \frac{n}{\tau}\right) \quad (D-72)$$

L'échantillonnage n'est valable que pour un signal à bande limitée, ceci peut être garanti en appliquant un filtre passe bande dans l'espace de Fourier. Ce filtre peut être choisi pour satisfaire au mieux le principe d'incertitude. Dans ce cas, la transformée de Fourier du signal de sortie s'écrit :

$$\text{TFB}(x) = \frac{1}{\tau} \cdot 2j\pi u \cdot \sum_{n=-\infty}^{\infty} \hat{A}\left(u - \frac{n}{\tau}\right) \cdot \hat{g}\left(u - \frac{n}{\tau}\right) \quad (\text{D-73})$$

$\hat{g}(u)$ est choisi de telle sorte que les spectres $\hat{A}\left(u - \frac{n}{\tau}\right) \cdot \hat{g}\left(u - \frac{n}{\tau}\right)$ ne se recouvrent pas. Pour dériver un signal, on peut, par exemple, le convoluer, avant échantillonnage, avec un filtre dont la réponse impulsionnelle est la dérivée d'une gaussienne.

ANNEXE E

Implantation récursive des opérateurs de Deriche et Shen

1 Rappels sur la transformée en Z

1.1 Définition

Un signal échantillonné à la période τ s'exprime sous la forme :

$$A[n\tau] = A(x) \cdot \Pi_\tau = \sum_{n=-\infty}^{+\infty} A(n\tau) \delta_{n\tau} \quad (\text{E-1})$$

où Π_τ est la distribution peigne.

Son spectre est une distribution périodique :

$$TF\{A \cdot \Pi_\tau\} = \sum_{n=-\infty}^{+\infty} A(n\tau) \exp(-2j\pi n \tau u) \quad (\text{E-2})$$

Le changement de variable $z = \exp(-2j\pi \tau u)$ transforme ce spectre en la fonction appelée transformée en Z :

$$A(z) = \sum_{n=-\infty}^{+\infty} A[n] z^{-n} \quad (\text{E-3})$$

qui est un développement sous forme d'une série de Laurent de la variable complexe z . La série converge dans le cercle de convergence défini par : $r < |z| < R$.

1.2 Quelques propriétés de la transformée en Z

La transformée en Z $A[i] \xleftrightarrow{Z} A(z)$ est linéaire.

Si r_a est un opérateur de retard :

$$r_a A[i] \xleftrightarrow{Z} z^{-a} A(z) \quad (\text{E-4})$$

$$i.A[i] \xleftrightarrow{Z} -z \frac{dA}{dz}(z) \quad (\text{E-5})$$

Soit un filtre discret de réponse impulsionnelle $h[i]$. La transformée en z de $h[i]$ s'appelle *fonction de transfert*. Pour que le filtre soit stable, il faut et il suffit

que la couronne de convergence de $h(z)$ contienne le cercle unité. Si le filtre est réalisable et causal, il est stable si les pôles de $h(z)$ sont à l'intérieur du cercle unité.

Pour un filtre linéaire tel que $C[i] = A * h[i]$, on a :

$$A * h[i] \xrightarrow{Z} A(z)h(z) \quad (\text{E-6})$$

$$\text{donc : } h(z) = \frac{C(z)}{A(z)}$$

Si $h(z)$ s'exprime sous la forme d'une fonction rationnelle :

$$h(z) = \frac{\sum_{u=0}^p c_u z^{-u}}{\sum_{v=0}^q a_v z^{-v}} \quad (\text{E-7})$$

on peut écrire que :

$$C(z) \sum_{u=0}^q c_u z^{-u} = A(z) \sum_{v=0}^p a_v z^{-v} \quad (\text{E-8})$$

et obtenir l'équation aux différences suivante :

$$C_i = \sum_{v=0}^p a_v A_{i-v} - \sum_{u=1}^q c_u C_{i-u} \quad (\text{E-9})$$

2 Implantation des opérateurs de Deriche et Shen

Ces filtres possèdent une réponse impulsionnelle infinie dont l'expression analytique permet une implantation récursive exacte. Pour les opérateurs de Deriche, les implantations sont de type « parallèle » alors qu'elles sont de type « cascade » ou « parallèle » pour les opérateurs de Shen.

2.1 Implantation des opérateurs monodimensionnels de Deriche

2.1.1 Dérivation

Une mise en oeuvre récursive du filtre optimal de dérivation :

$$h(x) = c \cdot x e^{-\alpha|x|} \quad (\text{E-10})$$

peut être obtenue en employant la transformée en Z . Soit $h[i]$ la séquence discrète résultant de l'échantillonnage de $h(x)$ et $h(z)$ sa transformée en Z :

$$h(z) = \sum_{i=-\infty}^{+\infty} h[i] z^{-i} \quad (\text{E-11})$$

La séquence $z[i]$ est mise sous la forme d'une somme d'une séquence causale $h_- i$ et d'une séquence non causale $h_+ i$:

$$h i = h_- i + h_+ i \quad (\text{E-12})$$

avec :

$$h_-[i] = \begin{cases} c i e^{-\alpha i} & \text{pour } i \geq 0 \\ 0 & \text{pour } i < 0 \end{cases} \quad (\text{E-13})$$

$$h_+[i] = \begin{cases} 0 & \text{pour } i \leq 0 \\ c i e^{\alpha i} & \text{pour } i < 0 \end{cases} \quad (\text{E-14})$$

La transformée $h(z)$ a pour expression :

$$h(z) = h_-(z^{-1}) + h_+(z) \quad (\text{E-15})$$

$$\text{avec } h_-(z^{-1}) = \frac{c e^{-\alpha} z^{-1}}{1 - 2 e^{-\alpha} z^{-1} + e^{-2\alpha} z^{-2}} \quad (\text{E-16})$$

$$\text{et } h_+(z) = \frac{-c e^{-\alpha} z}{1 - 2 e^{-\alpha} z + e^{-2\alpha} z^2} \quad (\text{E-17})$$

$h_-(z^{-1})$ (resp. $h_+(z)$) converge pour $|e^{-\alpha} z^{-1}| < 1$ (resp. $|e^{-\alpha} z| < 1$). Comme toutes les singularités de $h_+(z)$ (resp. $h_-(z^{-1})$) sont à l'intérieur (resp. extérieur) du cercle unité pour α réel positif, ces 2 transformées en Z correspondent à 2 fonctions de transfert de filtres récursifs stables et du second ordre. Le premier $h_-(z^{-1})$ opérant de gauche à droite et le second $h_+(z)$ de droite à gauche.

Le coefficient c est choisi pour obtenir un maximum d'amplitude de 1 en réponse à un échelon unitaire, ce qui donne :

$$B i = \sum_{m=-\infty}^{+\infty} U i - m h m = 1 \text{ pour } i = 1$$

$$B[0] = -c \sum_{m=0}^{+\infty} m e^{-\alpha m} = 1 \Rightarrow c = -\frac{(1 - e^{-\alpha})^2}{e^{-\alpha}} \quad (\text{E-18})$$

On peut remarquer que cette valeur de la constante de « normalisation » est différente de celle calculée pour un filtre continu.

On obtient alors les équations aux différences suivantes :

$$B_1 i = c e^{-\alpha} A i - 1 + 2 e^{-\alpha} B_1 i - 1 - e^{-2\alpha} B_1 i - 2$$

pour $i = 1, \dots, M$ (E-19)

$$B_2[i] = -ce^{-\alpha} A[i+1] + 2e^{-\alpha} B_2[i+1] - e^{-2\alpha} B_2[i+2] \quad (E-20)$$

pour $i = M, \dots, 1$

où M est la longueur de la séquence d'entrée $A[i]$. La séquence de sortie est alors donnée par :

$$B[i] = B_1[i] + B_2[i] \quad (E-21)$$

L'intérêt de cette mise en oeuvre récursive est le faible nombre d'opérations : 5 multiplications et 5 additions pour l'opérateur dérivée première. Le nombre d'opérations est indépendant de la valeur de la résolution α utilisée pour détecter les contours. En effet, la forme du filtre liée à α peut varier mais le nombre d'opérations par point reste identique. A titre de comparaison, une implantation RIF de l'opérateur $h[n]$ à l'aide de $2N+1$ coefficients non nuls requiert $2N+1$ opérations par point de sortie. Un gain de $\frac{2N+1}{5}$ en résulte en faveur de la mise en oeuvre récursive qui présente en plus l'avantage de ne pas introduire de déformation par troncature de la réponse impulsionnelle. Par exemple une implantation de cet opérateur de détection sous forme RIF pour $\alpha = 0.5$ et $\alpha = 0.25$ nécessite respectivement de l'ordre de 50 et 100 opérations par point.

2.1.2 Lissage

En adoptant la même démarche que dans le paragraphe précédent, le filtre de lissage :

$$f(x) = b(\alpha|x| + 1)e^{-\alpha|x|} \quad (E-22)$$

peut être implanté de manière récursive.

La transformée en Z du filtre numérique correspondant décomposé en partie causale et anti-causale s'écrit :

$$f(Z) = f_-(z^{-1}) + f_+(z) \quad (E-23)$$

$$\text{avec } f_-(z^{-1}) = b \frac{1 + e^{-\alpha}(\alpha - 1)z^{-1}}{1 - 2e^{-\alpha}z^{-1} + e^{-2\alpha}z^{-2}} \quad (E-24)$$

$$\text{et } f_+(z) = b \frac{e^{-\alpha}(\alpha + 1)z - e^{-2\alpha}z^2}{1 - 2e^{-\alpha}z + e^{-2\alpha}z^2} \quad (E-25)$$

Les domaines de convergence sont les mêmes que précédemment.

Les équations aux différences suivantes correspondantes ont pour expression :

$$B_1[i] = bA[i] + be^{-\alpha}(\alpha - 1)A[i - 1] + 2e^{-\alpha}B_1[i - 1] - e^{-2\alpha}B_1[i - 2] \quad (E-26)$$

pour $i = 1, \dots, M$

$$B_2[i] = be^{-\alpha}(\alpha + 1)A[i + 1] - be^{-2\alpha}A[i + 2] + 2e^{-\alpha}B_2[i + 1] - e^{-2\alpha}B_2[i + 2] \quad (E-27)$$

pour $i = M, \dots, 1$

$$B_i = B_1 i + B_2 i \quad (E-28)$$

Pour réaliser un lissage il faut donc 8 multiplications et 7 additions, indépendamment du facteur α .

La constante b est calculée pour obtenir une réponse d'amplitude 1 à une entrée constante unitaire :

$$b = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}} \quad (E-29)$$

Cette valeur est différente de celle trouvée en continu : $b = \frac{\alpha}{4}$

2.2 Implantation de l'opérateur gradient bidimensionnel de Deriche

En transposant les relations :

$$h_x(x, y) = -b\alpha^2 x e^{-\alpha|x|} \cdot b(\alpha|y| + 1)e^{-\alpha|y|} \quad (E-30)$$

$$h_y(x, y) = -b\alpha^2 y e^{-\alpha|y|} \cdot b(\alpha|x| + 1)e^{-\alpha|x|} \quad (E-31)$$

L'image $B_j[i, j]$ de la dérivée directionnelle suivant la direction x est calculée en appliquant la relation :

$$B_i i, j = A * f_j * h_i \quad (E-32)$$

De même :

$$B_j i, j = A * f_i * h_j \quad (E-33)$$

Le calcul se fait donc en appliquant les filtres monodimensionnels précédents.

La mise en oeuvre d'une dérivée directionnelle demande seulement 13 multiplications et 12 additions par point indépendamment de la taille du filtre spécifié par le paramètre α .

2.3 Implantation des opérateurs monodimensionnels de Shen-Castan

2.3.1 Opérateur de lissage

Nous partons des fonctions continues (5-95) (5-96) du chapitre V :

$$f_1(x) = \begin{cases} \beta e^{-\beta x} & \text{pour } x \geq 0 \\ 0 & \text{pour } x < 0 \end{cases} \quad (E-34)$$

$$f_2(x) = \begin{cases} 0 & \text{pour } x > 0 \\ \beta e^{\beta x} & \text{pour } x \leq 0 \end{cases} \quad (E-35)$$

Pour un signal d'entrée $A[i]$, le signal de sortie du filtre de lissage a pour expression :

$$B_i = A * f_1 * f_2 \quad (E-36)$$

avec les fonctions discrètes et leur transformée en Z correspondantes :

$$f_1[i] = \begin{cases} b e^{-\beta i} & \text{pour } i \geq 0 \\ 0 & \text{pour } i < 0 \end{cases} \quad f_1(z^{-1}) = \frac{b}{1 - e^{-\beta} z^{-1}} \quad (E-37)$$

$$f_2[i] = \begin{cases} 0 & \text{pour } i > 0 \\ b e^{\beta i} & \text{pour } i \leq 0 \end{cases} \quad f_2(Z) = \frac{b}{1 - e^{-\beta} z} \quad (E-38)$$

Les équations aux différences ont la forme suivante :

$$B_1 i = b A_i + e^{-\beta} B_1 i-1 \quad (E-39)$$

pour $i = 1, \dots, M$

$$B_2 i = b B_1 i + e^{-\beta} B_2 i+1 \quad (E-40)$$

pour $i = M, \dots, 1$

b est une constante de « normalisation » différente de celle calculée en continu. Elle est obtenue en fixant une réponse unitaire pour un signal d'entrée unitaire. On trouve alors :

$$b = 1 - e^{-\beta} \quad (E-41)$$

Les équations aux différences s'écrivent alors :

$$B_1[i] = e^{-\alpha} (B_1[i-1] - A[i]) + A[i] \quad (E-42)$$

pour $i = 1, \dots, M$

$$B_2[i] = e^{-\alpha} (B_2[i+1] - B_1[i]) + B_1[i] \quad (E-43)$$

pour $i = M, \dots, 1$

Avec cette nouvelle forme, il apparaît que ce traitement requiert 2 multiplications et 4 additions.

2.3.2 Opérateur de dérivation

On adopte la même démarche que précédemment en partant de la formulation continue :

$$f'(x) = \frac{\beta}{2} (f_2(x) - f_1(x))$$

pour obtenir une forme discrète donnée par :

$$B_i = A * f_2 - A * f_1 \quad (E-44)$$

avec :

$$f_1[i] = \begin{cases} c e^{-\beta i} & \text{pour } i \geq 0 \\ 0 & \text{pour } i < 0 \end{cases}$$

$$f_2[i] = \begin{cases} 0 & \text{pour } i > 0 \\ c e^{\beta i} & \text{pour } i \leq 0 \end{cases}$$

Les filtres $f_i[i]$ sont les mêmes que pour les filtres de lissage. La constante de normalisation qui est calculée pour obtenir une réponse d'amplitude 1 à l'origine est également la même :

$$c = 1 - e^{-\beta} \quad (\text{E-45})$$

Etant donné l'expression du dérivateur, son implantation est directe sous une forme « parallèle » :

$$B_1[i] = e^{-\alpha} (B_1[i-1] - A[i]) + A[i] \quad (\text{E-46})$$

pour $i = 1, \dots, M$

$$B_2[i] = e^{-\alpha} (B_2[i+1] - A[i]) + A[i] \quad (\text{E-47})$$

pour $i = M, \dots, 1$

$$B_i = B_2[i] - B_1[i] \quad (\text{E-48})$$

Elle demande 2 multiplications et 5 additions.

2.4 Implantation des opérateurs bidimensionnels de Shen-Castan

Pour une image $A[i, j]$, du fait de la séparabilité de l'opérateur de lissage, l'implantation bidimensionnelle est faite en exploitant les résultats de l'implantation monodimensionnelle.

2.4.1 Lissage

A partir de la relation donnant l'expression dans le domaine continu de l'image lissée B :

$$B(x, y) = A * f(x) * f(y) \quad (\text{E-49})$$

On peut écrire que :

$$B_{i, j} = A * f_{i, j} = A * f_{1, i} * f_{2, i} * f_{1, j} * f_{2, j} \quad (\text{E-50})$$

Ce lissage comprend 4 multiplications et 8 additions.

2.4.2 Opérateurs de dérivation première

Pour le calcul des dérivées première et seconde, on utilise la même démarche que ci-dessus en étendant au domaine discret les formules (5-109) et (5-110) rappelée ci-après :

$$C_x(x, y) = [(A * f(y)) * f_2(x) - (A * f(y)) * f_1(x)]$$

$$C_y(x, y) = [(A * f(x)) * f_2(y) - (A * f(x)) * f_1(y)]$$

Donc :

$$B_i \ i, j = A * f_2 \ i - A * f_1 \ i * f_2 \ j * f_1 \ j \quad (E-51)$$

$$B_j \ i, j = A * f_2 \ j - A * f_1 \ j * f_2 \ i * f_1 \ i \quad (E-52)$$

L'opérateur dérivée première directionnelle demande 4 multiplications et 10 additions.

ANNEXE F

Discrétisation des critères de Canny

Les filtres utilisés en détection de contours sont généralement développés dans le domaine spatial continu puis transposés par échantillonnage dans leur champ d'application : le monde discret. La période d'échantillonnage est fixe (distance entre deux pixels) et la largeur des filtres le plus souvent inférieure à une dizaine de pixels, il en résulte un échantillonnage très épars des réponses impulsionnelles continues ce qui induit de grandes différences entre les propriétés du filtre discret et de son homologue continu. Nous définissons ici la version discrétisée des critères de Canny tout en préservant leur signification physique. Cette étude met en évidence l'existence de deux types de dérivateurs suivant que leur réponse impulsionnelle dans le domaine continu exhibe ou non une discontinuité au centre.

Les critères discrets permettent :

- de comparer les filtres entre eux, ce qui est impossible à partir des critères continus,
- de quantifier les dégradations introduites, par exemple, par une troncature de réponse impulsionnelle (synthèse RIF) ou par une approximation des coefficients du filtre.

D'autre part, le critère de bonne détection peut être étendu aux filtres du type lisseur. Le critère de bonne localisation peut être appliqué au filtre de type dérivée seconde et le critère de faible multiplicité des réponses est applicable aux deux types. Le paragraphe 1.8.2 du chapitre V montre une application de ces critères à la comparaison des filtres dérivée première de Shen, de Deriche, DOB (*Difference Of Boxes*) et DOG (dérivée de gaussienne).

La principale difficulté de transposition des critères au monde discret est l'utilisation dans leur définition (*cf.* chapitre V § 1.6.1) des dérivées première et seconde de la réponse impulsionnelle. Toute méthode basée sur une approximation de ces dérivées serait imparfaite. La bonne approche consiste à travailler directement dans le domaine échantillonné et à recalculer entièrement les critères.

Nous noterons h_d la réponse impulsionnelle du filtre discret et h_c celle du filtre continu associé quand il existe.

Le signal d'entrée A servant à définir les critères est un échelon discret U d'amplitude U_0 noyé dans un bruit blanc gaussien discret N de variance N_0^2 .

$$\forall i, A[i] = U_0 U[i] + N[i] \text{ avec } \begin{cases} \forall i < 0 U[i] = 0 \\ \forall i \geq 0 U[i] = 1 \end{cases} \quad (\text{F-1})$$

1 Critère de bonne détection

Il correspond au rapport du RSB (rapport signal sur bruit) de sortie sur le RSB d'entrée. Le signal est défini à l'entrée comme l'amplitude de l'échelon et à la sortie comme le maximum de la réponse à $U[i]$.

L'expression discrète de ce critère ne pose pas de difficulté puisque ne faisant pas intervenir les dérivées de h_c . On note $Cd1$, ce critère pour les filtres du type dérivée première et $Cl1$ pour les filtres de lissage. Pour $Cl1$, l'entrée est un signal constant d'amplitude U_0 .

$$Cd1 = \frac{\sum_{k=-\infty}^0 h_d[i]}{\left[\sum_{k=-\infty}^{+\infty} h_d^2[i] \right]^{\frac{1}{2}}} \quad Cl1 = \frac{\sum_{k=-\infty}^{+\infty} h_d[i]}{\left[\sum_{k=-\infty}^{+\infty} h_d^2[i] \right]^{\frac{1}{2}}} \quad (\text{F-2})$$

2 Probabilité de localisation exacte du maximum

Ce calcul intermédiaire est utile pour permettre d'établir les deux autres critères de Canny.

On note C la réponse du filtre à l'entrée A . Dans le cas d'un filtre ayant un zéro au centre de sa réponse impulsionnelle (filtres RII du chapitre V ou filtres RIF à nombre impair de coefficients), en absence de bruit, on obtient deux maximums situés en $i = -1$ et $i = 0$. Par ailleurs, les filtres RIF à nombre pair de coefficients, ou RII sans zéro central produisent un seul maximum situé en $i = 0$ [DEMI 94].

En présence de bruit, il est clair que la probabilité d'obtenir un maximum en $i = 0$ sera identique à celle de l'avoir en $i = -1$ et que ces deux événements sont incompatibles. Dans un premier temps, nous étudions la probabilité d'une détection en $i = 0$. La condition nécessaire s'écrit alors :

$$C_0 - C_{-1} > 0 \quad \text{et} \quad C_0 - C_1 > 0 \quad (\text{F-3})$$

Soit :

$$C_0 - C_{-1} = c_1 + X_1 \quad \text{et} \quad C_0 - C_1 = c_2 + X_2 \quad (\text{F-4})$$

avec :

$$\begin{cases} X_1 = \sum_{i=-\infty}^{+\infty} (h_d[i] - h_d[i-1]) \cdot N[-i] \\ X_2 = \sum_{i=-\infty}^{+\infty} (h_d[i] - h_d[i+1]) \cdot N[-i] \end{cases} \quad (\text{F-5})$$

$$\text{et : } \begin{cases} c_1 = U_0 \cdot h_d[0] = 0 \\ c_2 = U_0 \cdot h_d[-1] = -U_0 \cdot h_d[1] \end{cases} \quad (\text{F-6})$$

$N[i]$ est un bruit blanc gaussien, alors X_1 et X_2 sont deux variables aléatoires gaussiennes d'écart type σ , corrélées par le filtrage. On note ρ le coefficient de corrélation.

En notant les suites :

$$h'_d[i] = h_d[i] - h_d[i-1] \quad \text{et} \quad h''_d[i] = h'_d[i] - h'_d[i-1]$$

on obtient :

$$\sigma^2 = N_0^2 \sum_{i=-\infty}^{+\infty} h_d'^2[i] \quad \text{et} \quad \rho = \frac{\sum_{i=-\infty}^{+\infty} h_d''^2[i]}{2 \sum_{i=-\infty}^{+\infty} h_d'^2[i]} - 1 \quad (\text{F-7})$$

On peut exprimer la densité de probabilité $P(x_1, x_2)$.

$$P(x_1, x_2) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \cdot \exp\left(-\frac{x_1^2}{2\sigma^2}\right) \cdot \exp\left(-\frac{(x_2 - \rho x_1)^2}{2\sigma^2(1-\rho^2)}\right) \quad (\text{F-8})$$

La probabilité P_{\max} d'avoir un maximum localisé en $i = 0$ est donc :

$$P_{\max} = P(X_1 > -c_1, X_2 > -c_2) = \int_{-c_1}^{+\infty} \int_{-c_2}^{+\infty} P(x_1, x_2) \cdot dx_1 \cdot dx_2 \quad (\text{F-9})$$

On introduit la fonction de répartition normalisée $L(a, b, \rho)$ définie par :

$$L(a, b, \rho) = \int_a^{+\infty} \int_b^{+\infty} \frac{1}{2\pi\sqrt{1-\rho^2}} \cdot \exp\left(-\frac{x_1^2}{2}\right) \cdot \exp\left(-\frac{(x_2 - \rho x_1)^2}{2(1-\rho^2)}\right) \cdot dx_1 \cdot dx_2 \quad (\text{F-10})$$

Alors, pour les filtres centrés, la probabilité d'obtenir un maximum localisé en $i = -1$ ou $i = 0$ vaut :

$$P = 2 \cdot L\left(\frac{Ah_d[1]}{\sigma}, 0, \rho\right) \quad (\text{F-11})$$

σ étant proportionnel à l'écart type du bruit N_0 , P dépend du rapport signal à bruit d'entrée, mais de façon non linéaire.

3 Critère de faible multiplicité des réponses (Cd3)

De l'étude du paragraphe précédent, on déduit aisément le troisième critère de Canny. En effet, en absence de signal ($U_0 = 0$), La probabilité P_b d'obtenir un maximum en 0 représente la probabilité d'avoir un maximum dû au bruit en un endroit quelconque puisque le processus est stationnaire.

On obtient :

$$P_b = L(0,0,\rho) = \frac{\text{acos}(-\rho)}{2\pi} \quad (\text{F-12})$$

Par application des lois classiques sur les probabilités conditionnelles en remarquant qu'on ne peut avoir deux maximums en deux pixels immédiatement voisins, il découle que la distance moyenne entre deux maximums dus au bruit vaut :

$$\text{Cd3} = \frac{1}{P_b} = \frac{2\pi}{\text{acos}(-\rho)} \quad (\text{F-13})$$

Puisque dans l'étude aboutissant à ce résultat aucune hypothèse n'est faite sur la forme du filtre linéaire (cas où $U_0 = 0$), il en résulte que ce critère est applicable à tous les filtres et en particulier aux filtres lisseurs.

P_b mesure le rapport du nombre moyen de « bosses » du signal de sortie au nombre de pixels et Cd3 la distance moyenne entre les « bosses ».

Comportement asymptotique de Cd3

Lorsque l'échantillonnage est suffisamment fin, c'est-à-dire quand la largeur de la réponse impulsionnelle du filtre est grande, on peut voir si le critère Cd3 tend vers l'expression trouvée dans le cadre continu :

$$Cc3 = 2\pi \left(\frac{\int_{-\infty}^{+\infty} h_c'^2(x) dx}{\int_{-\infty}^{+\infty} h_c''^2(x) dx} \right)^{\frac{1}{2}} \quad (\text{F-14})$$

Si h_d correspond à la fonction « support » continue h_c , il est possible d'étudier l'influence du facteur d'échelle ω en introduisant une fonction « support » normalisée, c'est à dire indépendante de ω , g telle que : $g(x) = h_c(\omega x)$. Augmenter ω revient à échantillonner plus finement la fonction g .

On étudie alors le comportement asymptotique de l'expression $R(\omega)$:

$$R(\omega) = \frac{\sum_{i=-\infty}^{+\infty} h_d'^2[i]}{\sum_{i=-\infty}^{+\infty} h_d''^2[i]} = \frac{1}{2(1+\rho)} \quad (\text{F-15})$$

On montre alors que si $g(x)$ est une fonction continue en $x = 0$, alors lorsque ω tend vers l'infini, il en va de même de $R(\omega)$ et de $Cd3$; ρ tend alors vers -1.

Dans ce cas, pour ω suffisamment grand on a $Cd3 \approx Cc3$. C'est, par exemple, le cas du filtre DOG ou du filtre de Deriche. Il est important de noter que, pour le filtre de Deriche, avec un facteur d'échelle important $\alpha = 0.1$, on trouve $Cd3 = 30.3$ et $Cc3 = 28.1$, ce qui correspond à une erreur relative de 7.3% en regard du critère continu!

Par contre, si $g(x)$ est discontinue en $x = 0$ (cas du filtre de Shen) alors lorsque ω tend vers l'infini, $R(\omega)$ tend vers 1 et ρ tend vers -0.5 et donc $Cd3$ reste inférieur à 6. $Cd3$ est donc borné pour ce type de filtre.

4 Critère de bonne localisation

Une extension simple de l'expression donnant la probabilité d'un maximum localisé en $i = 0$ définit la probabilité de détecter le maximum en $i = i_0$ quelconque.

$$P[i_0] = L\left(-\frac{A.h_d[i_0]}{\sigma}, \frac{A.h_d[i_0+1]}{\sigma}, \rho\right) \quad (\text{F-16})$$

La figure F.1 montre l'évolution de cette probabilité en fonction de la distance à la position réelle de la transition de l'échelon pour le filtre de Deriche ($\alpha = 0.5$) et pour différentes valeurs de RSB. L'allure générale de ces caractéristiques est indépendante du type de filtre utilisé. On voit que lorsque le RSB augmente, une zone de probabilité nulle (correspondant à un meilleur RSB local) est créée entre la région de la transition et la région dominée par le bruit d'asymptote P_b .

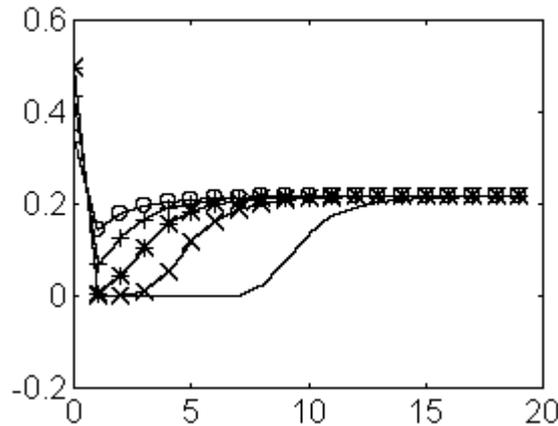


Figure F.1 : Variation de la probabilité qu'un maximum soit localisé à la distance i de la transition pour différents RSB : (o RSB = 1; +RSB = 2; * RSB = 4; x RSB = 10 et . RSB = 100)

On peut alors calculer la largeur N de la zone dans laquelle on est sûr de détecter le contour :

$$N = \arg \min_I \sum_{i=0}^I P(i) \geq 0.5 \quad (\text{F-17})$$

La valeur 0.5 est utilisée parce que l'antisymétrie de la réponse impulsionnelle des filtres dérivée première autorise le calcul sur un seul côté de la réponse. Ainsi, la probabilité de trouver le maximum dû à la transition dans la zone $[-(N+1), N]$ est de 1.

On définit $Q(i)$ pour ne garder que la part de $P(N)$ nécessaire pour compléter la probabilité à 0.5 :

$$\begin{aligned} 0 \leq i < N & \quad Q(i) = P(i) \\ i = N & \quad Q(N) = 0.5 - \sum_{i=0}^{N-1} P(i) \\ i > N & \quad Q(i) = 0 \end{aligned} \quad (\text{F-18})$$

On peut maintenant calculer la variance de la distance à laquelle on détecte la transition telle que Canny l'a défini :

$$dx_0^2 = 2 \sum_{i=0}^{+\infty} i^2 Q(i) \quad (\text{F-19})$$

Ce résultat est valable que la délocalisation soit faible ou pas puisqu'aucune approximation au voisinage de $x = 0$ n'a été utilisée comme dans les travaux originaux de Canny (*cf.* chapitre V § 1.6.1). Par contre, cette expression de la distance n'est pas inversement proportionnelle au RSB, on ne peut alors définir Cd_2 comme l'a fait Canny par une expression du type :

$$\frac{N_0}{A \cdot \sqrt{dx_0^2}} \quad (F-20)$$

Nous convenons donc simplement de prendre : $Cd2 = \frac{1}{\sqrt{dx_0^2}}$

La figure F.2 montre différentes mesures de localisation pour le filtre de Deriche pour différentes valeurs de α , le RSB étant fixé à 3. La courbe pleine correspond au tracé du produit $RSB \times Cd2$, c'est à dire à l'application de notre critère. Les o correspondent à une mesure expérimentale de la délocalisation (10 mesures sur 500 échelons pour chaque valeur de α), les barres d'erreur de mesure valent un écart type. Les * correspondent à l'application du critère continu de Canny, et les + à une discrétisation brutale (on remplace les intégrales par des sommes discrètes) de l'expression continue du critère de localisation. Des mesures à RSB variable ont été menées qui confirment aussi la justesse de cette approche.

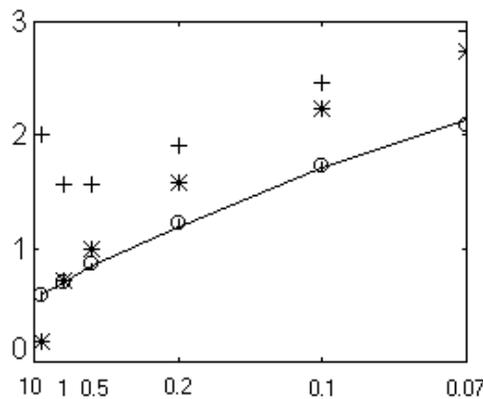


Figure F.2 : mesure de localisation pour le filtre de Deriche pour différentes valeurs de α avec un RSB = 3

On peut expliquer l'échec du critère continu par les remarques suivantes. Lorsque α est grand (Par exemple $\alpha=1$), l'échantillonnage de la réponse continue est éparse et les filtres continu et discret sont très différents; d'autre part quand α est petit, le remplacement des intégrales par des sommes discrètes ainsi que l'approximation des dérivées par des différences finies sont corrects (les points marqués + rejoignent ceux marqués * sur la figure F.2), mais la délocalisation est forte et le développement limité employé par Canny au voisinage de zéro n'est plus valable.

Quelques résultats.

Nous ne donnons pas ici beaucoup de résultats puisque le cas des filtres les plus utilisés en détection est examiné dans le chapitre V au paragraphe 1.8.2.

Mais montrons l'application à quelques filtres simples :

- **filtre identité $y[i] = x[i]$:**

Ce filtre ne fait rien, mais permet de calculer la distance moyenne entre les maximums d'un bruit blanc gaussien. On obtient $\rho = 0.5$ et $Cd3 = 3$.

- **filtre de Sobel :**

le dérivateur $[-1 \ 0 \ 1]$ donne $Cd1 = 1/\sqrt{2}$ et $Cd3 = 3.44$

le lisseur $[1 \ 2 \ 1]$ donne $C11 = 1.63$ et $Cd3 = 4.77$

BIBLIOGRAPHIE

Sigles des revues ou congrès les plus couramment cités

Congrès AFCET-RFIA : Congrès AFCET Reconnaissance des Formes et Intelligence Artificielle.

CGIP : Computer Graphics and Image Processing.

Colloque GRETSI : Colloque du Groupe de Recherche en Traitement du Signal et des Images.

CVGIP : Computer Vision Graphics and Image Processing.

ICASSP : International Conference on Acoustic, Speech and Signal Processing.

ICCV : International Conference on Computer Vision.

ICPR : International Conference on Pattern Recognition.

IEEE Trans. on ASSP : IEEE Transaction on Acoustic, Speech and Signal Processing.

IEEE Trans. on PAMI : IEEE Transaction on Pattern Analysis and Machine Intelligence.

IEEE Trans. on SMC : IEEE Transaction on System, Man and Cybernetic.

INNC : International Neural Networks Conference.

Chapitre II

[BARN88] M. F. Barnsley, R. L. Devaney, B. B. Mandelbrot, H-O. Peitgen, D. Saupe, R. Voss — The science of fractal images, *Springer-Verlag*, Heidelberg, (1988).

[BESL86] P.J. Besl, R.C. Jain — Invariant surface characteristics for 3-D object recognition in range images, *CVGIP* 33, p.33-80, (1986).

[BROD66] P. Brodatz — Textures, *Dover Publications*, New-York, (1966).

[BROW92] G. Brown, G. Michon, J. Peyrière — On the multifractal analysis of measures, *Journ. of Stat. Phys.*, t. 66, p 775-790, (1992).

[CHAS91] J.M Chassery, A. Montanvert — Géométrie discrète en analyse d'images, *Hermès*, Paris, (1991).

[CONN80a] R.W. Connors, C.A. Harlow — Toward a structural textural analyser based on statistical methods, *CGIP* 12, 224-256, (1980).

[CONN80b] R.W. Connors, C.A. Harlow — A theoretical comparaison of texture algorithms, *IEEE Trans. on PAMI*, vol. 2, n°3, 204-222, (1980).

[DAVI79] L.S. Davis, S.A. Johns, J.K. Aggarwal — Texture analysis using generalized co-occurrence matrices, *IEEE Trans. on PAMI*, vol. 1, n°3, 251-259, (1979).

- [DAVI80] L.S. Davis, A. Mitiche — Edge detection in textures, *CGIP* 12, 25-39, (1980).
- [DAVI81] L.S. Davis, M. Clearman, J.K. Aggarwal — An empirical evaluation of generalized cooccurrence matrices, *IEEE Trans. on PAMI*, vol. 3, n°2, 214-221, (1981).
- [GAGA83] A. Gagalowicz — Vers un modèle de textures, Thèse d'Etat, Université Pierre et Marie Curie, Paris VI, (1983).
- [GALL75] M. M. Galloway — Texture analysis using gray level run lengths, *CGIP* 4, 172-179, (1975).
- [GANG86] J.J. Gangepain, C. Roques-Carmes — Fractal approach to two-dimensional and three dimensional surface roughness, *Wear* 109, 119-126, (1986).
- [GONZ92] R.C. Gonzalez, R.E. Woods — Digital image processing, *Addison-Wesley*, (1992).
- [HARA73] R. Haralick, K. Shanmugan, I. Dinstein — Textural features for image classification, *IEEE Trans. on SMC*, vol. 3, n°6, 610-621, (1973).
- [HARA79] R. Haralick — Statistical and structural approaches to textures, *Proceedings of the IEEE*, vol 67, n°5, 786-804, (1979).
- [HENT83] H.G.E. Hentschel, I. Procaccia — The infinite number of generalized dimensions of fractals and strange attractors, *Physica 8D*, 435-444, (1983).
- [JOUR89] M. Jourlin, J.C. Pinoli, R. Zeboudj — Contrast definition and contour detection for logarithmic images, *Journal of Microscopy*, 156, 33-40, (1989).
- [JULE62] B. Julesz — Visuel pattern recognition, *IEEE Trans. on Information Theory*, vol. 8, (1962).
- [JULE84] B. Julesz — A brief outline of the texton theory of human vision, *Trends in Neurosciences*, (1984).
- [KELL89] J. M. Keller, S. Chen, R. M. Crownover — Texture description and segmentation through fractal geometry, *CVGIP*, 45, 150-166, (1989).
- [LECH91] Y. Lechevallier, J. Lévy-Véhel, P. Mignot, Arthur, un système d'analyse de texture, *8^{ème} Congrès AFCET-RFIA*, Lyon-Villeurbanne, (nov.1991).
- [LEVI79] J.Z. Levin — Mathematical models for determining the intersections of quadric surfaces, *CGIP* 11, 73-87, (1979).
- [LEVY90] J. Lévy-Véhel — About lacunarity, some links between fractal and integral geometry, *ICCV*, Osaka, (1990).
- [LEVY94] J. Lévy-Véhel, P. Mignot — Multifractal segmentation of images, *Fractals*, 2(3), 371-377, (1994).
- [MA83] S. Ma — Modélisation et synthèse de texture ; Application à l'infographie, Thèse d'Etat, Université P. et M. Curie, ParisVI, (1986).

- [MAND68] B.B.Mandelbrot, J.W. Van Ness—Fractional Brownian motions, fractional noises and applications, *SIAM Review*, vol. 10, n°4, 422-437, (1968).
- [MAND82] B.B. Mandelbrot— The fractal geometry of nature, *W. H. Freeman and Co*, New York, (1982).
- [MATS83] T. Matsuyama, S. Miura, M. Nagao — Structural analysis of natural textures by Fourier transformation, *CVGIP*, 24, 347-362, (1983).
- [OHAN92] P. P. Ohanian, R. C. Dubes — Performance evaluation for four classes of textural features, *Pattern Recognition*, vol. 25, n°8, 819-833, (1992).
- [PAPO91] A. Papoulis — Probability random variables and stochastic process, *Mc Graw Hill*, New-York, (1991).
- [PEET85] F. G. Peet, T.S. Sahota — Surface curvature as a measure of image texture, *IEEE Trans. on PAMI*, vol. 7, n°6, 734-738, (1985).
- [PENT84] A.P.Pentland — Fractal-based description of natural scenes, *IEEE Trans. on PAMI*, vol. 6, n°6, 661-674, (1984).
- [PENT86] A.P. Pentland — Shading into textures, *Artif. Intell* 29, 147-170, (1986).
- [PHIL88] S. Philipp — Analyse de texture appliquée aux radiographies industrielles, Thèse de l'Université P. et M. Curie, Paris VI, (1988).
- [PHIL89] S. Philipp — Modélisation structurale de la texture. Extraction du grain primaire et de sa règle de placement, *12^{ème} Colloque GRETSI*, Juan-les-Pins, (juin 1989).
- [PHIL93] S. Philipp, J-P. Cocquerez — Fusion de régions sur critères géométriques, *14^{ème} colloque GRETSI*, Juan-les-Pins, (sept. 1993).
- [PHIL94] S. Philipp, M. Smadja — Approximation of granular textures by quadric surfaces, *Pattern Recognition* 27, n°8, 1051-1063, (1994).
- [PRET87] F. Prêteux — Description et interprétation des images par la morphologie mathématique. Application à l'imagerie médicale, Thèse d'Etat, Université P. et M. Curie, Paris VI, (1987).
- [PRET91] F. Prêteux, X. Descombes — Synthèse et analyse de textures par coopération de processus multi-échelles, *8^{ème} Congrès AFCET- RFIA*, Lyon-Villeurbanne, (nov. 1991).
- [RAO90] A.R. RAO — A taxonomy for texture description and identification, Springer series in Perception engineering, *Springer-Verlag*, Heidelberg, (1990).
- [SARK92] N. Sarkar, B.B. Chaudhuri — An efficient approach to estimate fractal dimension of textural images", *Pattern Recognition*, vol. 25, n°9, 1035-1041, (1992).
- [VOLE87] P. Volet — Analyse et synthèse d'images de textures structurées, Thèse de l'Ecole Polytechnique de Lausanne, (1987).

- [VOSS86] R. Voss — Random fractals : Characterization and measurement, in *Scaling Phenomena and disordered systems*, in R. Pynn & A. Skjeltorp eds, *Plenum Press*, New York, 1-11, (1986).
- [YOKO89] N. Yokoya, M.D. Levine — Range image segmentation based on differential geometry : a hybrid approach, *IEEE Trans. on PAMI*, vol. 11, n°6, (1989).
- [ZEBO88] R. Zéboudj — Filtrage, Seuillage automatique, Contraste et Contours : du pré-traitement à l'analyse d'image, Thèse de l'Université de Saint-Etienne, (1988).

Chapitre III

- [AHO89] A. Aho, J. Hopcroft, G. Ullman — Structures de données et algorithmes, *InterEditions*, (1989).
- [ATTA94] D. Attali, A. Montanvert — Semi-continuous skeletons for 2D and 3D shapes, 2nd Int. Workshop on Visual Form, Capri (Italy), *Plenum Press*, New-York, (1994).
- [CHAS91] J. M. Chassery, A. Montanvert — Géométrie discrète en analyse d'images, *Hermès*, Paris, (1991).
- [DAVI77] L. S. Davis — Understanding shape : angles and sides, *IEEE Trans. on Computers*, vol. 26, n°3, 236-242, (1977).
- [FREE61] H. Freeman — On the encoding of arbitrary geometric configurations, *IRE Trans. Electron. Compu.*, 10, 260-268, (1961).
- [JOLI90] J.M. Jolion — Analyse d'images : Le Modèle Pyramidal, *Traitement du Signal*, vol. 7, 5-17, (1990).
- [KLEI89] R. Klein — Concrete and abstract Voronoi diagrams, *Lecture Notes in Computer Science*, *Springer-Verlag*, Heidelberg, (1989).
- [MATH93] C. Mathieu — Segmentation d'images par pyramides souples : Application à l'imagerie médicale multi-dimensionnelle, Thèse de l'INSA, Lyon, (1993)
- [MATH95] C. Mathieu, I. Magnin — On the choice of the first level of graph pyramids, *Journal of Mathematical Imaging and Vision*, (1995).
- [MEAG82] D. Meagher — Geometric Modeling Using Octree Encoding, *CGIP*, vol. 19, 129-147, (1982).
- [MEER89] P. Meer — Stochastic image pyramids, *CVGIP*, vol. 45, 269-294, (1989).
- [MELK91] M. Melkemi — Approches géométriques par modèles de Voronoï en segmentation d'images, Thèse de l'Université J. Fournier, Grenoble I, (1991).
- [PAVL74] T. Pavlidis, S. L. Horowitz — Segmentation of plane curves, *IEEE Trans. on Computers*, vol. 23, n°8, 860-870 (1974).
- [PHIL68] G. M. Phillips — Algorithms for piecewise straight line approximations, *Computer Journal*, 2, 211-212, (1968).

- [PREP88] F.P. Preparata, M.I. Shamos — Computational Geometry, an introduction, Texts and Monographs in Computer Science, *Springer-Verlag*, Heidelberg, (1988).
- [ROSE74] A. Rosenfeld — Adjacency in Digital Pictures, *Information and Control*, vol. 26, 24-33, (1974).
- [ROSE66] A. Rosenfeld, J. L. Pfaltz — Sequential operations in digital picture processing, *Journal of ACM*, vol. 13, n°4, 471-494, (1966).
- [ROSE73] A. Rosenfeld, E. Johnston — Angle detection on digital curves, *IEEE Trans. on Computers*, 875-878, (1973).
- [SAME82] H. Samet — Neighbour finding techniques for images represented by quadtrees, *CGIP*, vol. 18, 298-303, (1982).
- [SAME84] H. Samet — The Quadtree and Related Hierarchical Data Structures, *Computing Surveys*, vol. 16, n°2, 187-260, (1984).
- [SERR82] J. Serra — Image analysis and mathematical morphology, *Academic press*, London, (1982).
- [WALL84] K. Wall, P. E. Danielsson — A fast sequential method for polygonal approximation of digitized curves, *CVGIP*, 28, 220-227, (1984).

Chapitre IV

- [ARCE87] G.R. Arce, P.Mc Loughlin — Theoretical analysis of the max-median filter, *IEEE Trans. on ASSP*, vol. 35, n°1, 60-69, (1987).
- [ATAM80] E. Ataman, V.K. Aatre, K.M. Wong — A fast method for real time median filtering, *IEEE Trans. on ASSP*, vol. 28, n°4, 415-420, (1980).
- [ASTO89] J. Astola, Y. Neuvo — Optimal median type filters for exponential noise distributions, *Signal Processing*, vol. 17, n°2, 95-104, (1989).
- [BELL84] M. Bellanger — Traitement numérique du signal, *Masson*, Paris, (1984).
- [BOLO91] Ph. Bolon, A. Raji — Filtrage médian récursif multiétage. Etude des propriétés statistiques et morphologiques, *12^{ème} Colloque GRETSI*, Juan les Pins, 1005-1008, (1991).
- [BOLO92] Ph. Bolon — Filtrage d'ordre, vraisemblance et optimalité des prétraitements d'image, *Traitement du Signal*, vol. 9, n°3, 225-250, (1992).
- [BOVI83] A.C. Bovik, T.S. Huang, D.C. Munson — A generalization of median filtering using linear combinations of order statistics, *IEEE Trans. on ASSP*, vol. 31, n°6, 1342-1350, (1983).
- [BROW84] D.R.K. Brownrigg — The weighted median filter, *Comm. ACM*, vol. 27, n°8, 807-818, (1984).
- [CAI86] Y.L. Cai, C.S. Chen — An edge preserving smoothing filter based on the robust estimate, *Proc. 8th ICPR*, Paris, 206-208, (1986).

- [CHEH91] K. Chehdi, M. Sabri, M. Corazza — Amélioration d'un algorithme de filtrage du bruit multiplicatif, *Traitement du Signal*, vol. 8, n°1, 63-74, (1991).
- [COCQ85] J.P. Cocquerez, J. Devars — Détection de contours dans les images aériennes : nouveaux opérateurs, *Traitement du Signal*, vol. 2, n°1, 45-64, (1985).
- [COST89] M. Coster, J-L. Chermant — Précis d'analyse d'images, *Editions CNRS*, Paris, (1989).
- [DAVI81] H.A. David — Order statistics, *Wiley Interscience*, (1981).
- [FONT93] P. Fontanot, G. Ramponi — A polynomial filter for the preprocessing of mail address images, *Proc. of IEEE Workshop on Nonlinear Digital Signal Processing*, Tampere, 2161-2166, (1993).
- [GALL81] N.C. Gallagher, G.L. Wise — A theoretical analysis of the properties of median filters, *IEEE Trans. on ASSP*, vol. 29, n°6, 1136-1141, (1981).
- [GONZ87] R.C. Gonzalez, P. Wintz — Digital image processing, *Addison Wesley*, 2nd edition, (1987).
- [HEIN87] P. Heinonen, Y. Neuvo — FIR median hybrid filters, *IEEE Trans. on ASSP*, vol. 35 n° 6, 832-838, (1987).
- [HILL90] A. Hillion, J.M. Boucher — A unified approach to nonlinear processing of multiplicative noise with applications to radar images, *Signal Processing V Theories and Applications*, L.Torres et al. Eds, *Elsevier*, 2027-2029, (1990).
- [HUAN79] T.S. Huang, G.J. Yang, G.Y. Tang — A fast 2D median filtering algorithm, *IEEE Trans. on ASSP*, vol. 27, n°1, 13-18, (1979).
- [HUBE64] P.J. Huber — Robust estimation of a location parameter, *Annals of Mathematical Statistics*, vol. 35, 73-101, (1964).
- [HUMM77] R. Hummel — Image enhancement by histogram transformation, *CGIP*, vol. 6, 184-195, (1977).
- [ITO88] K. Ito, Y. Ichioka, T. Minami — Nearest neighbor median filter, *Applied Optics*, vol. 27, n°16, 3445-3450, (1988).
- [JAIN89] A.K. Jain — Fundamentals of digital image processing, *Prentice-Hall*, (1989).
- [KRAM75] H.P. Kramer, J.B. Bruckner — Iterations of non-linear transformations for enhancement on digital images, *Pattern Recognition*, n°7, 53-58, (1975).
- [KIM86] V. Kim, L. Yaroslavskii — Rank algorithms for picture processing, *CVGIP*, vol. 35, 234-258, (1986).
- [KUNT93] M. Kunt, G.Granlund, M.Kocher — Traitement numérique des images, *Presses Polytechniques et Universitaires Romandes*, (1993).
- [LEE83] J.S. Lee — Digital image smoothing and the sigma filter, *CVGIP*, vol. 24, 255-269, (1983).

- [LEE85] Y.H. Lee, S.A. Kassam — Generalized median filtering and related nonlinear filtering techniques, *IEEE Trans. on ASSP*, vol. 33, n°3, 672-683, (1985).
- [LEE87] Y.H. Lee, A.T. Fam — An edge gradient enhancing adaptive order statistic filter, *IEEE Trans. on ASSP*, vol. 35, n°5, 680-695, (1987).
- [LIN88] H.M. Lin, A.N. Willson — Median filters with adaptive length, *IEEE Trans. on ASSP*, vol. 36, n°6, 675-690, (1988).
- [LLOY52] E.H. Lloyd — Least square estimation of location and scale parameters using order statistics, *Biometrika*, vol. 39, 88-95, (1952).
- [MARR80] D. Marr, E. Hildreth — Theory of edge detection, *Proc. Roy. Soc. London*, B-207, 187-214, (1980).
- [MATH83] G. Matheron — Filters and lattices, *Rapport interne CGMM n°851*, Ecole des Mines de Paris, (1983).
- [NAGA79] M. Nagao, M. Matsuyama — Edge preserving smoothing, *CVGIP*, vol. 9, 394-407, (1979).
- [NIEM88] A. Niemenen, Y. Neuvo — Comments on Theoretical analysis of max/median filter, *IEEE Trans. on ASSP*, vol. 36, n°5, (1988).
- [OCHO87] E. Ochoa, J.P. Allebach, D.W. Sweeney — Optical median filtering using threshold decomposition, *Applied optics*, vol. 26, n°2, (1987).
- [PITA90] I. Pitas, A.N. Venetsanopoulos — Non linear digital filters. Principles and applications, *Kluwer Academic Press*, London, (1990).
- [POMA84] C.A. Pomalaza-Raez, C.D. McGillem — An adaptive non linear edge preserving filter, *IEEE Trans. on ASSP*, vol. 32 n°3, 571-576, (1984).
- [PRET87] F. Prêteux — Description et interprétation des images par la morphologie mathématique. Application à l'imagerie médicale, Thèse d'Etat, Université P. et M. Curie, Paris VI, (1987).
- [RAMP88] G. Ramponi — Quadratic filters for image enhancement, *Signal Processing IV*, Theories and Applications, *Lacoume et al. Eds*, 239-242, *Elsevier*, (1988).
- [REST88] A. Restrepo, A.C. Bovik — Adaptive trimmed mean filters for image restoration, *IEEE Trans. on ASSP*, vol. 36, n°8, 1326-1337, (1988).
- [RICH90] D.S. Richards — VLSI median filters, *IEEE Trans. on ASSP*, vol. 38, n°1, 145-155, (1990).
- [ROSE82] A. Rosenfeld, A.C. Kak — Digital picture processing, vol. 1, *Academic Press*, London, (1982).
- [SALE92] Ph. Salembier — Adaptive rank order based filters, *Signal Processing*, vol. 27, 1-25, (1992).
- [SCHU94] M.A. Schulze, J.A. Pearce — A morphology-based filter structure for edge-enhancing smoothing, *Proc. IEEE Int. Conf. on Image Proces.*, Austin USA, 530-534, (1994).

- [SERR82] J. Serra — Image analysis and Mathematical Morphology, *Academic Press*, London, (1982).
- [SERR86] J. Serra — Eléments de théorie pour l'optique morphologique, Thèse d'Etat, Université P. et M. Curie, Paris VI, (1986).
- [SERR88] J. Serra — Image analysis and Mathematical Morphology : Theoretical advances, *Academic Press*, London, (1988).
- [SHEN85] J. Shen, S. Castan — Un nouvel algorithme de detection de contours, *5^{ème} Congrès AFCET-RFIA*, Grenoble, 201-213, (1985).
- [SUN94] T. Sun, M. Gabbouj, Y. Neuvo, — Adaptive L-filters with applications in signal and image processing, *Signal Processing*, vol. 38, n°3, 331-344, (1994).
- [VILA93] J.L. Vila, Ph. Bolon — Filtrage d'ordre adaptatif pour le prétraitement d'images naturelles, *14^{ème} Colloque GRETSI*, Juan-les Pins, 563-566, (sept. 1993).
- [WANG90] X. Wang, D. Wang — On the max/median filter, *IEEE Trans. on ASSP*, vol. 38, n°8, 1473-1475, (1990).
- [WEND86] P. Wendt, E.J. Coyle, N.C. Gallagher — Stack filters, *IEEE Trans. on ASSP*, vol. 34, n°4, 898-911, (1986).
- [YAN86] P.F. Yan, C.H. Chen — An algorithm for filtering multiplicative noise in wide range, *Traitement du Signal*, vol. 3, n°2, 91-96, (1986).
- [ZAMP90] P. Zamperoni — Some adaptive rank order filters for image enhancement, *Pattern Recognition Letters*, vol. 11, 81-86, (1990).
- [ZAMP92] P. Zamperoni — Adaptive rank-order filters for image processing based on local anisotropy measures, *Digital Signal Processing*, vol. 2, 174-182, (1992).

Chapitre V

- [CANN86] J.F. Canny — A Computational Approach to Edge Detection, *IEEE Trans. on PAMI*, vol. 8, n°6, 679-698, (1986).
- [CAST89] S. Castan, J. Zhao, J. Shen — Une Famille de Détecteurs de Contours Basée sur le Filtre Exponentiel Optimal, *7^{ème} Congrès AFCET-RFIA*, Paris, (1989).
- [COCQ85] J.P. Cocquerez et J. Devars — Détection du contours dans les images aériennes : nouveaux opérateurs, *Traitement du Signal*, vol. 2, n°1, (1985).
- [DERI87] R. Deriche — Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector, *International Journal of Computer Vision*, vol. 1, n°2, 167-187, (1987).
- [DERI90] R. Deriche — Fast algorithms for low-level vision, *IEEE Trans. on PAMI*, vol. 12, n°1, 78-87, (1990).
- [HARA84] R.M. Haralick — Digital Step Edges from Zero-Crossings of Second Directional Derivative, *IEEE Trans. on PAMI*, vol. 6, n°1, 58-68, (1984).

- [HUEC71] M.F. Hueckel — An operator which locates edges in digitized pictures, *J. Ass. Comput. Mach.*, vol. 18, n°1, 113-125, (1971).
- [HUER86] A. Huertas and G. Medioni — Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-gaussian Masks, *IEEE Trans. on PAMI*, vol. 8, n°5, 651-661, (1986).
- [KIRS71] R. Kirsch — Computer Determination of the Constituent Structure of Biological Images, *Computer Biomedical Research*, 4, 315-328, (1971).
- [MARR80] D. Marr, E. Hildreth — Theory of Edge Detection, *Proceedings of Royal Society of London*, vol. 207, 187-217, (1980).
- [NAGA79] M. Nagao, T. Matsuyama — Edge Preserving Smoothing, *CVGIP*, 9, 394-407, (1979).
- [POGG85] T. Poggio — Early Vision : From Computational Structure to Algorithms and Parallel Hardware, *CVGIP*, 31, 135-155, (1985).
- [PREW70] J.M.S. Prewitt — Object Enhancement and Extraction, *Picture Processing and Psychopictorics*, B.S.Likin and A. Rosenfeld, *Academic Press*, London, 75-149, (1970).
- [RICE54] S.O. Rice — Mathematical Analysis of Random noise, *Selected papers on noise and stochastic processes*, Dover, N.Y., (1954).
- [SERR88] J. Serra — Image analysis and mathematical morphology : theoretical advances, vol. 2, *Academic press*, London, (1988).
- [SHEN92] J. Shen, S. Castan — An Optimal Linear Operator for Step Edge Detection, *CVGIP*, vol. 54, 112-133, (1992).

Chapitre VI

- [BALL81] D. H. Ballard — Generalizing the HOUGH transform to detect arbitrary shapes, *Pattern Recognition*, 13, 2, 111-122, (1981).
- [BONN91] C. Bonnet, D. Dresp — Psychophysique de l'extraction de contours en vision humaine, *8^{ème} Congrès AFCET-RFIA*, Lyon-Villeurbanne, (nov. 1991).
- [CHAS91] J. M. Chassery, A. Montanvert — Géométrie discrète en analyse d'images, *Hermès*, Paris, (1991).
- [COHE83] M.A.Cohen, S. Grossberg — Absolute stability of global pattern formation and parallel memory storage by competitive neural network, *IEEE. Trans. on SMC*, vol. 13, 5, 815-826, (1983).
- [DEMI91] J.F. Quesne, D. Demigny, J. Devars, J.P. Cocquerez — Architecture temps réel pour la fermeture des contours et l'étiquetage des régions, *8^{ème} Congrès AFCET-RFIA*, Lyon-Villeurbanne, (nov. 1991).
- [DERI88] R. Deriche, J.P. Cocquerez, G. Almozni — An efficient method to build early image description, *9th ICPR*, Rome, (nov. 1988).

- [GAUS91] P. Gaussier, J.P. Cocquerez — Utilisation des réseaux de neurones pour la reconnaissance de scènes complexes : simulation d'un système comprenant plusieurs aires corticales, *Traitement du Signal*, vol. 8, n°6, 441-466, (1991).
- [GROS82] S. Grossberg — Study of mind and brain, *Reidel*, Boston, (1982).
- [GROS85a] S. Grossberg, E. Mingolla — Neural dynamics of form perception : boundary completion, illusory figures and neon color spreading, *Psychological Review*, vol. 92, n°2, 173-211, (1985).
- [GROS85b] S. Grossberg, E. Mingolla — Neural dynamics of perceptual grouping : textures, boundary and emergent segmentation, *Perception and Psychophysics*, vol. 38, n°2, 141-171, (1985).
- [GROS87] S. Grossberg — Cortical dynamics of three dimensional form, color and brightness perception : monocular theory, *Perception and Psychophysics*, vol. 41, n°2, 87-116, (1987).
- [GROS88] S. Grossberg — Non-linear networks : principles, mechanism and architectures, *Neural Network*, 1, 17-61, (1988).
- [GROS91] S. Grossberg, D. Somers — Synchronized oscillations during cooperative feature linking in a cortical model of visual perception, *Neural Network*, 4, 453-466, (1991).
- [LEHA90] S. Lehar, T. Howells, I. Smotroff — Application of Grossberg and Mingolla Neural vision model to satellite weather imagery, *INNC90*, 805-808, (1990).
- [NILS71] N.J. Nilsson — Problem-solving methods in artificial intelligence, McGraw-Hill, (1971).
- [OSAW90] K. Osawa — Simulation studies on optical illusion based on a position dependant spread function, *Pattern Recognition*, 23, 12, 1361-1366, (1990).
- [PAVL74] T. Pavlidis, S. L. Horowitz — Segmentation of plane curves, *IEEE Trans. on Computers*, 23(8), 860-870 (1974).
- [QUES91] D. Demigny, J.F. Quesne, J. Devars, Boundary closing with asynchronous cellular automata. *Computer Architecture for Machine Perception*, Paris, (déc. 1991).
- [RAME70] U. Ramer — An iterative procedure for the polygonal approximation of plane curves, *CGIP*, 12, 244-256, (1970).
- [RUME85] D. E. Rumelhart, D. Zipser — Feature discovery by competitive learning, *Cognitive Sciences*, 9, 75-112, (1985).
- [SEIB89] M. Seibert, A. Waxman — Spreading activation layers, visual saccades and invariant representation for neural pattern recognition systems, *Neural Networks*, 2, 9-27, (1989).
- [SKRZ90] J. Skrzypek — Lightness constancy : connectionist architecture for controlling sensitivity, *IEEE Trans. on SMC*, vol. 20, n°5, (1990).
- [SLAN80] J. Slansky, V. Gonzales — Fast polygonal approximation of digital curves, *Pattern Recognition*, 12, 327-331, (1980).

- [VOND84] R. Von der Heydt, E. Peterhans, G. Baumgartner — Illusory contours and cortical neuron responses, *Science*, 224, 1260-1262, (1984).

Chapitre VII

- [AZEN87] R. Azencott — Image analysis and Markov fields, *Proc. of ICIAM*, Paris, (1987).
- [AZEN92] R. Azencott — Simulated annealing : parallelization technics, *Wiley*, (1992).
- [BILB88] G. Bilbro, R. Mann, T. Miller, W. Snyder, D. Van den Bout et M. White — Optimization by mean field annealing, *Advances in Neural Information Processing Systems*, *Touretzky Ed.*, vol. 1, 91-98, (1988).
- [BLAK87] A. Blake et A. Zisserman — Visual reconstruction, *MIT Press*, Cambridge-MA, (1987).
- [CAMP89] C. Campbell, D. Sherington et K.Y.M. Wong — Statistical mechanics and neural Network Computing Architecture, *MIT Press*, Cambridge-MA, I Aleksander Ed, 239-257, 1989.
- [GEIG90] D. Geiger et F. Girosi — Parallel and deterministic algorithms for MRFs : surface reconstruction and integration, *Proc. ECCV*, Antibes, (avr. 1990).
- [GEIG91] D. Geiger et F. Girosi — Parallel and deterministic algorithms for MRFs : surface reconstruction, *IEEE Trans. on PAMI*, vol. 13, 401-412, (1991)
- [GEMA84] S. Geman et D. Geman — Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. on PAMI*, vol. 6, 721-741, (1984).
- [GEMA87a] S. Geman, D. Geman et C. Graffigne — Locating texture and object boundaries, *Pattern Recognition Theory and Application*, Ed P.A. Devijver et J. Kittler, Heidelberg, *Springer-Verlag*, Heidelberg, (1987).
- [GEMA87b] S. Geman et C. Graffigne — Markov random fields and image models and their application to computer vision, *Proc. ICM86*, Ed A.M. Gleason, Math. Soc. -Providence, (1987).
- [GEMA90] S. Geman, D. Geman, C. Graffigne et P. Dong — Boundary detection by constrained optimization, *IEEE Trans. on PAMI*, vol. 12, 609-628, (1990).
- [HERA89] P. Herault et J. Niez — Neural networks and graph K-partitioning, *Complex Systems*, vol. 3, 531-575, (1989).
- [KIND80] R. Kindermann et J.L. Snell — Markov random fields and their applications, *Amer. Math. Soc.*, vol. 1, 1-142, (1980).
- [MARR85] J. Marroquin, S. Mitter et T. Poggio — Probabilistic solution of ill-posed problems in computational vision, *Proc. Image Understanding Workshop*, Scient. App. Int. Corp., (août 1985).
- [MARR87] J. Marroquin — Deterministic Bayesian estimation of Markovian random fields with applications to computational vision, *Proc. ICCV*, London, (juin 1987).

- [MEMI91] E. Memin, F. Charot et F. Heitz — Parallel algorithms and architectures for multiscale Markov random fields base image analysis, *Proc. Workshop on computer architecture for machine perception*, 309-320, Paris, (déc. 1991).
- [SIMC89a] T. Simchony, R. Chellappa et Z. Lichtenstein — Pyramid implementation of optimal step conjugate search algorithm for some low level vision problems, *IEEE Trans. on SMC*, vol. 19, 1408-1426, (1989).
- [SIMC89b] T. Simchony, R. Chellappa et Z. Lichtenstein — The Graduated Non Convexity algorithm for image estimation using Compound Gauss-Markov Fields models, *Proc. ICASSP*, Glasgow, (mai 1989).
- [YUIL90] A.L. Yuille, D. Geiger et H. Bulthoff — Stereo integration, mean field theory and psychophysics, *Proc. ECCV*, Antibes, (avril 1990).
- [ZERU90] J. Zerubia et R. Chellappa — Mean field approximation using Compound Gauss-Markov Random Field for edge detection and image restoration, *Proc. ICASSP*, Albuquerque, (avril 1990).
- [ZERU91] J. Zerubia et F. Ployette — Détection de contours et restauration d'image par des algorithmes déterministes de relaxation. Mise en oeuvre sur la machine à connexions CM2, *Traitement du Signal*, vol. 8, n°3, 165-179, (1991).
- [ZERU93] J. Zerubia et R. Chellappa — Mean field annealing using compound Gauss-Markov random fields for edge detection and image estimation, *IEEE Trans. on Neural Network*, vol. 4, 703-709, (1993).

Chapitre VIII

- [BASC93] B. Bascle, R. Deriche — Stereo matching, reconstruction and refinement of 3D curves using deformable contours, In *Proc. of the IEEE Int. Conf. on Computer Vision*, 421-430, (1993).
- [BERG91] M.O. Berger — Les contours actifs : modélisation, comportement, convergence, Thèse INPL, Nancy, (1991).
- [BOSS94] PL. Bossart — Détection de contours réguliers dans des images bruitées et texturées : association contours actifs et approche multiéchelle, Thèse INPG, Grenoble, (1994).
- [BLAK93] A. Blake, R. Curwen, A. Zisserman — Affine-invariant contourtracking with automatic control of spatiotemporal scale, In *Proc. of the IEEE ICCV*, 66-75, (1993).
- [CHAS95] JM. Chassery, Y. Elomary — Coopération contours actifs et multi-résolution en segmentation d'images, *15^{ème} colloque GRETSI*, Juan-les-Pins, (sept. 1995).
- [CIPO92] R. Cipolla, A. Blake — Surface shape from the deformation of apparent contours, *Int. Journal of Computer Vision*, vol. 9, 2, 83-112, (1992).
- [COHE93] L.D. Cohen, I. Cohen — Finite element methods for active contourmodels and ballons for 2D and 3D images, *IEEE Trans. on PAMI*, vol. 15, 11, 1131-1147, (1993).
- [COHE90] L.D. Cohen — Etude des modèles de contours actifs et autres techniques de traitement d'images, Thésis de l'Université d'Orsay, Paris XI, (1990).

- [COOT92a] T.F. Cootes, C.J. Taylor — Active shape models : smart snakes, In *Proc. of the British Machine Vision Conference*, Leeds, UK, 266-275, (1992).
- [COOT92b] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham — Training models of shape from sets of examples, In *Proc. of the British Machine Vision Conference*, Leeds, UK, 9-18, (1992).
- [ELOM94] Y. Elomary — Modèles déformables et multi-résolution pour la détection de contours, Thèse de l'Université de Grenoble, (1994).
- [GREN92] U. Grenander, Y. Chow, D.M. Keenan, Hands — A pattern theoretical study of biological shapes, *Springer-Verlag*, Heidelberg, New-York, (1991).
- [HARR90] J.Harris, C.Koch, J.Luo, J.Wyatt — Analog hardware for detecting discontinuities in early vision, *Int Journal of Computer Vision*, vol. 4, 211-223, (1990).
- [HEIT92] F. Heiter, L.Rosenthaler — Simulation of Neural Contour Mechanismes : from Simple to End-stopped Cell, *Vision research*, vol. 32, 5, 963-981, (1992).
- [KASS87] M. Kass, A. Witkin — Analyzing oriented patterns, *Computer Vision, Graphics and Image Processing*, 37, 362-385, (1987).
- [KASS88] M. Kass, A. Witkin, D. Terzopoulos — Snakes : Active contour models, *International Journal of Computer Vision*, 1, 312-331, (1988).
- [LEAV93] V.F. Leavers — Which Hough transform?, *CVGIP-Image understanding*, vol. 58, 2, 250-264, (1993).
- [LEYM93] F. Leymarie, M.D. Levine — Tracking deformable objects in the plane using an active contour model, *IEEE Trans. on PAMI*, vol. 15, n°6, 617-634, (1993).
- [LIND89] T. Lindeberg — Scale-space for discrete signals, *IEEE Trans. on PAMI*, vol. 12, n°3, 234-254, (1989).
- [MAIT85] H. Maître — Un panorama sur la transformation de Hough, *Traitement du Signal*, vol. 2, n°4, 305-317, (1985).
- [MOHA89] R. Mohan, R. Nevatia — Segmentation and description based on perceptual organisation, In *Proc. of the IEEE Conf on Computer Vision and Pattern Recognition*, 333-334, (1989).
- [MORE94] JM. Morel — Variational methods in image segmentation. Serie in progress in nonlinear differential equations, *Birkhause*, (1994).
- [RIPL90] B.D. Ripley, A.I. Sutherland — Finding spiral structures in images of galaxies, *Phil. Trans. Royal Society London*, A332, 477-485, (1990).
- [SOLI90] F. Solina, R. Bajcsy — Recovery of parametric models from range images : the case for superquadrics, *IEEE Trans. on PAMI*, vol. 12, n°2, 131-147, (1990).
- [TERZ92] D. Terzopoulos, R. Szeliski — Tracking with Kalman snakes, In *A. Blake and A. Yuille eds*, Active vision, 3-20, MIT Press, (1992).

- [UEDA92] N. Ueda, K. Mase — Tracking moving contours using energy-minimizing contour models, in *Proc. of the European Conference on Computer Vision*, Bari, Italy, 453-457, (1992).
- [YUIL92] A.L. Yuille, P.W. Hallinan, D.S. Cohen — Feature extraction from faces using deformable templates, *Int. Journal of Computer Vision*, vol. 8, n°2, 99-111, (1992).

Chapitre IX

- [BLAK87] A. Blake, A. Zisserman — Visual reconstruction, MIT Press, Cambridge-MA, (1987).
- [COCQ85] J-P. Cocquerez, J. Devars — Détection de contours dans les images aériennes : nouveaux opérateurs, *Traitement du Signal*, vol. 2, n°1, (1985).
- [DEMI93a] D. Demigny, T. Kamlé, L. Kessal — Implémentation non récursive du filtre de Deriche : performance et complexité, *14^{ème} colloque GRETSI*, Juan-les Pins, (sept.1993).
- [DEMI93b] D. Demigny, J. Devars, L. Kessal, J-F. Quesne — Implémentations temps réel du filtre de lissage d'images de Nagao, *Traitement du Signal*, vol.10, n°4, (janv 1993).
- [FRAM75] J.R. Fram, E.S. Deutsch — On the Quantitative Evaluation of Edge Detection Schemes and their Comparison with Human Performance, *IEEE Trans. on Comp.*, vol. 24, 616-628, (1975).
- [FRAM78] J.R. Fram, E.S. Deutsch — A Quantitative Study of the orientation bias of some Edge Detection Schemes, *IEEE Trans. on Comp.*, vol. 27, 205-213, (1978).

Chapitre X

- [ADE83a] F. Ade — Characterization of textures by « eigenfilters », *Signal Processing*, 5, 451-457, (1983).
- [ADE83b] F. Ade — Application of principal components analysis to the inspection of industrial goods, *Proc. SPIE Conf.*, Genève, (1983).
- [AURA90] J. Auray, G. Duru, A. Zighed — Analyse des données multidimensionnelles, *Ed. Alexandre Lacassagne*, (1990).
- [BART79] P.H. Bartels — Numerical evaluation of cytologic data : Description of Profiles, *Analytical and Quantitative Cytology*, vol. 1, n°1, 20-28, (1979).
- [BHAN82] B. Bhanu, O.D. Faugeras — Segmentation of images having unimodal distributions, *IEEE Trans. on PAMI*, vol. 4, 408-419, (1982).
- [BHAN87] B. Bhanu, B.A. Parvin — Segmentation of natural scenes, *Pattern Recognition*, vol. 20, n°5, 487-496, (1987).
- [BHAT67] C.G. Bhattacharya — A simple method of resolution of a distribution into gaussian components, *Biometrics*, vol. 23, 115-135, (1967).
- [CAIL76] F. Caillez, J. Pages — Introduction à l'analyse des données, *Soc. de Math. appliquées et de sciences humaines*, (1976).

- [CELE82] G. Celeux, Y. Lechevallier — Méthodes de segmentation non paramétriques, *R.S.A.*, vol. 30, n°4, 39-53, (1982).
- [CELE89] G. Celeux, E. Diday, G. Govaert, Y. Lechevallier, H. Ralambondrainy — Classification automatique des données, *Dunod Informatique*, Paris, (1989).
- [CHEH89] A. Chéhikian — Binarisation d'image : deux solutions à ce problème, *Traitement du Signal*, vol. 6, n°1, 59-68, (1989).
- [CHOW72] C.K. Chow, T. Kaneko — Automatic boundary detection of the left ventricle from cineangiograms, *Comput.Biomed.Res.*, vol. 5, 388-410, (1972).
- [COLE79] G. B. Coleman, H. C. Andrews — Image segmentation by clustering, *Proc. IEEE*, vol. 67, n°5, 773-785, (1979).
- [DIDA82] E. Diday, J. Lemaire, J. Pouget, F. Testu — Eléments d'analyse de données, *Dunod*, Paris, (1982).
- [DUBU90] B. Dubuisson — Diagnostic et reconnaissance des formes, *Hermès*, Paris, (1990).
- [DUDA73] R.O. Duda, E. Hart — Pattern Classification and scene analysis, *ed J. Wiley & sons*, (1973).
- [FISH58] W.D. Fisher — On grouping for maximum homogeneity, *JASA*, vol. 53, 789-798, (1958).
- [FU81] K.S.Fu, J.K.Mui — A survey on image segmentation, *Pattern Recognition*, vol. 13, 3-16, (1981).
- [GAGA88] A. Gagalowicz, C. Graffigne — Blind Texture Segmentation, *IEEE Proc. IAPR*, Rome, (1988).
- [HARA85] R.M. Haralick, L.D. Shapiro — Image segmentation techniques, *CVGIP*, vol. 29, 100-132, (1985).
- [HERT88] L.Hertz, R.W.Schafer — Multilevel thresholding using edge matching, *CVGIP*, vol. 44, 279-295, (1988).
- [INRI90] Analyse discriminante sur variables continues, Editeur G. Celeux, Coll. Didactique *INRIA*, (1990).
- [LAWS80] K. I. Laws — Rapid texture identification, *Proc. SPIE*, vol. 238, Image Processing for Missila Guidance, (1980).
- [LEBA82] L. Lebart, A. Morineau, J-P. Fénelon — Traitement des données statistiques, *Dunod*, Paris, (1982).
- [LECH91] Y. Lechevallier, J. Lévy-Véhel, P. Mignot — Arthur, un système d'analyse de texture *8^{ème} Congrès AFCET-RFIA*, Lyon-Villeurbanne, (nov. 1991).
- [LEVY92] J. Lévy-Véhel, P. Mignot, J. Berroir — Multifractals, Texture, and Image Analysis, *Proc. CVPR* (1992).

- [MARD88] K.V. Mardia, T.J. Hainsworth — A spatial thresholding method for image segmentation, *IEEE Trans. on PAMI*, vol. 10, n°6, 919-927, (1988).
- [NAKA79] Y.Nakagawa, A.Rosenfeld — Some experiments on variable thresholding, *Pattern Recognition*, vol. 11, 191-204, (1979).
- [OTSU79] N. Otsu — A threshold selection method from gray-level histograms, *IEEE Trans. on SMC*, vol. 9, n°1, 62-66, (1979).
- [PAPP92] T. N. Pappas — An adaptative clustering algorithm for image segmentation, *IEEE Trans. on Signal Processing*, vol. 40, n°4, (1992).
- [PREP88] F. Preparata, M.I. Shamos — Computational Geometry, an introduction, *Springer-Verlag*, Heidelberg, (1988).
- [PUN80] T. Pun — A new method for gray-level picture thresholding using the entropy of the histogram, *Signal Processing*, 2, 223-237, (1980).
- [ROSE76] A.Rosenfeld, R. Hummel, S.W. Zucker — Scene labeling by relaxation operation, *IEEE Trans. on SMC*, vol. 6, n°6, 420-430, (1976).
- [SAHO78] P.K. Sahoo, S. Soltani, AKC Wong, YC Chen — A survey of thresholding techniques, *CVGIP*, vol. 41, 233-260, (1988).
- [UNSE89] M. Unser, M. Eden — Multiresolution feature extraction and selection for texture segmentation, *IEEE Trans. on PAMI*, vol. 11, n°7, (1989).
- [WANG90] L. Wang, DC. He — Texture classification using texture spectrum, *Pattern Recognition*, vol. 23, n°8, (1990).
- [WESK74] J.S.Weszka, R.N.Nagel, A.Rosenfeld — A threshold selection technique, *IEEE Trans. on Computers*, vol. 23, 1322-1326, (1974).
- [WESK78] J.S.Weszka — A survey of threshold selection techniques, *CGIP*, 7, 259-265, (1978).

Chapitre XI

- [AZEN87] R. Azencott — Image analysis and Markov fields, *Proc. of ICIAM*, Paris, (1987).
- [AZEN92a] R.Azencott, C. Graffigne, C. Labourdette — Textured image segmentation and edge detection, Lecture Notes in Statistics, stochastic models, statistical methods and algorithms in image analysis, Eds. P. Barone, A. Frigessi and M. Piccioni, *Springer-Verlag*, Heidelberg, (1992).
- [AZEN92b] R.Azencott, C. Graffigne — Non supervised segmentation using multilevel Markov random fields, *Proc. ICPR*, The Hague, (1992).
- [BAUM72] L. Baum, T. Petrie, N. Soules, N. Weiss — A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Ann. Math. Statist.*, vol. 41, 164-171, (1972).
- [BEND78] E.A. Bender — The asymptotic number of labelled graphs with given degree sequences, *J. of Comb. Theory*, A24, 296-307, (1978).

- [BESA74] J. Besag — Spatial interaction and the statistical analysis of lattice systems, *J. of Royal Statistical Society*, 2, 192-236, (1974).
- [BESA86] J. Besag — On the statistical analysis of dirty pictures, *J. of Royal Statistical Society*, series B, 48, (1986).
- [BOLO82] B. Bolobas, W. Fernandez de la Vega — The diameter of random regular graph, *Combinatorica* 2, 2, 125-134, (1982).
- [BOLO85] B. Bolobas — Random graph, *Academic Press*, London, (1985).
- [CHAL88] B. Chalmond — Image restauration using an estimated Markov model, *Signal Processing*, n° 15, 115-129, (1988).
- [CHAL89] B. Chalmond — An iterative gibbsian technique for reconstruction of m-ary images, *Pattern Recognition*, n° 22, 747-761, (1988).
- [DEMP77] A. Dempster, N. Laird et D. Rubin — Maximum likelihood from incomplete data via the EM algorithm, *J. of Royal Statistical Society*, series B, 39(1), 1-38, (1977).
- [GEMA84] S. Geman, D. Geman — Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. on PAMI*, vol. 6, 721-741, (1984).
- [GEMA87a] S. Geman, C. Graffigne — Markov random fields and image models and their application to computer vision, *Proc. ICM86*, Ed A.M. Gleason, Math. Soc. -Providence, (1987).
- [GEMA87b] S. Geman, D. Geman, C. Graffigne — Locating structure and object boundaries, *Pattern recognition theory and applications*, P.A. Devivjer and J. Kittler, *Springer-Verlag*, Heidelberg, (1987).
- [GEMA87c] S. Geman, D. Geman — A stochastic algorithm for global optimization with constraints, *Technical Report*, Brown University, (1987).
- [GEMA90] S. Geman, D. Geman, C. Graffigne, P. Dong — Boundary detection by constrained optimization, *IEEE Trans. on PAMI*, vol. 12, 609-628, (1990).
- [HARA73] R.M. Haralick, K. Shanmugan, I. Denstein — Textural features for image classification, *IEEE Trans. on SMC*, vol. 6, 610-620, (1973).
- [KATO92] Z. Kato, J. Zerubia, M. Berthod — Satellite image classification using a modified Metropolis dynamics, *Proc. ICASSP*, San Francisco, (1992).
- [KATO93] Z. Kato, J. Zerubia, M. Berthod — Bayesian image classification using Markov random fields, in *Maximum entropy and Bayesian Methods*, Mohammed-Djafari and Demoment Eds, *Kluwer Pub.*, 375-383, (1993).
- [QIAN89] W. Qian, D.M. Titterington — On the use of Gibbs markov chain models in the analysis of images based on second-order pairwise interactive distributions, *J. Applied Statistics*, 16, 267-281, (1989).
- [YOUN89] L. Younes — Parameters estimation for imperfectly observed gibbsian fields, *Prob. Theory and Rel. Fields*, 82, 625-645, (1989).

Chapitre XII

- [AHUJ85] N. Ahuja, B. An, B. Schachte — Image representation using Voronoi tessellation, *CVGIP*, vol. 29, 286-295, (1985).
- [BRET89] F. Brétaudeau — Une nouvelle approche orientée étiquetage pour la détection des frontières en milieu texturé, *Actes de PIXIM'89*, Paris, (1989).
- [BURT83] P.J. Burt, E.H. Adelson — The laplacian pyramid as a compact image code, *IEEE Trans. on Comp.*, vol. 31, 4, 337-345, (1983).
- [CHAN91] F. Chantemargue, M. Popovic, R. Canals, P. Bonton — Parallelization of the merging step of the region segmentation method, *7th Scandinavian Conference on Image Analysis*, Aalborg (Denmark), 993-940, (août 1991).
- [CHAS84] J.M. Chassery, C. Garbay — An iterative segmentation method based on a contextual color and shape criterion, *IEEE Trans. on PAMI*, vol. 6, n°6, 794-800, (1984).
- [CHAS91] J.M. Chassery, M. Melkemi — Diagramme de Voronoï appliqué à la segmentation d'images et à la détection d'événements en imagerie multi-sources, *Traitement du Signal*, vol. 8, n°3, 155-164, (1991).
- [CHEN] X. Chen, F. Schmitt — Split and merge image segmentation based on Delaunay triangulation, *7th Scandinavian Conference on Image Analysis*, Aalborg (Denmark), 910-917, (août 1991)
- [CLER91] P. Clermont, S. Cartier — Implementation and evaluation of region growing algorithms on CM2, *Proc. of the Int. Colloquium on Parallel Image Processing*, Paris, 91-109, (juin 1991).
- [GAGA86] A. Gagalowicz, O. Monga — A new approach for image segmentation, *8th ICPR*, Paris, October 1986.
- [HORO74] S.L. Horowitz, T. Pavlidis — Picture segmentation by directed split-and-merge procedure, *Second Int. Joint Conference on Pattern Recognition*, (1974).
- [HUNT79] G.M. Hunter, K.S. Steiglitz — Operations on images using quadtree, *IEEE Trans. on PAMI*, vol. 1, 2, 145-153, (1979).
- [JOLI92a] J.M. Jolion, A. Montanvert — The adapted pyramid: a framework for 2D image analysis, *CVGIP: Image Understanding*, vol. 55, 3, 339-348, (1992).
- [JOLI92b] J.M. Jolion, A. Rosenfeld — Pyramid framework for early vision, *Kluwer*, (1992).
- [KROP90] W.G. Kropatsch, A. Montanvert — Irregular versus regular pyramid structures, *GEOBILD'91, 5th Workshop on Geometrical Problems of Image Processing*, Georgenthal (Germany), Published in *Research in Informatics*, vol. 4, ed. *Eckhardt et al.*, 11-22, (mars 1991).
- [LOWI83] G.E. Lowitz — Can a local histogram really map texture information, *Pattern Recognition*, 16, 141-147, (1983).

- [LUBY86] M. Luby — A simple parallel algorithm for the maximal independent set problem, *SIAM J. Comp.*, vol. 15, n°4, 1036-1053, (1986).
- [MATH93] C. Mathieu — Segmentation d'images par pyramides souples : Application à l'imagerie médicale multi-dimensionnelle, Thèse de l'INSA, Lyon, (1993)
- [MATH95] C. Mathieu, I. Magnin — On the choice of the first level of graph pyramids, *Journal of Mathematical Imaging and Vision*, (1995).
- [MEER89a] P. Meer, S. Connelly — A fast parallel method for synthesis of random patterns, *Pattern Recognition*, 22, 189-204, (1989).
- [MEER89b] P. Meer — Stochastic image pyramids, *CVGIP*, vol. 45, 269-294, (1989).
- [MEYE92] Y. Meyer — Les ondelettes : Algorithmes et applications, *Armand Colin*, Paris, (1992).
- [MONG87] O. Monga — An optimal growing algorithm for image segmentation, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 1, n°3, 351-376, (1987).
- [MONT91] A. Montanvert, P. Meer, A. Rosenfeld — Hierarchical image analysis using irregular tessellations, *IEEE Trans. on PAMI*, vol. 13, 307-316, (April 1991).
- [MUER68] J.L. Muerle, D.C. Allen — Experimental evaluation of techniques for automatic segmentation of object in a complex scene, *Pictorial Pattern Recognition* (G.C. Cheng & al., Eds.), Thompson, Washington, 3-13, (1968).
- [PAVL77] T. Pavlidis — Structural pattern recognition, *Springer-Verlag*, Heidelberg, (1977).
- [SECC90] P. Secchi — Segmentation d'images et fermeture de contours par corrélation d'histogrammes locaux: Application aux images de télédétection SPOT, Thèse de l'Université P. et M. Curie, Paris VI, (Juin 1990).
- [TUCE90] M. Tuceryan, A.K. Jain — Texture segmentation using Voronoi polygones, *IEEE Trans. on PAMI*, vol. 13, 211-216, (1990).

Chapitre XIII

- [BROD66] P. Brodatz — Textures, *Dover Publications*, New-York, (1966).
- [JOUR89] M. Jourlin, J.C. Pinoli, R. Zéboudj — Contrast definition and contour detection for logarithmic images, *Journal of Microscopy*, 156, 33-40, (1989).
- [EDMO65] J. Edmonds — Paths, trees and flowers, *Canad. J. of Math.*, 17, 449-467, (1965).
- [KUHN55] H.W Kuhn — The hungarian method for the assignment problem, *Nav. Res. Log. Quat.*, 2, (mars-juin 1955).
- [LECH91] Y. Lechevallier, J. Lévy-Véhel, P. Mignot — Arthur, un système d'analyse de texture, 8^{ème} *AF CET-RFIA*, Lyon-Villeurbanne, (nov. 1991).
- [VINE91] L. Vinet — Segmentation et mise en correspondance de régions de paires d'images stéréoscopiques, Thèse de l'Université de Dauphine, Paris IX, (1991)

- [ZEBO88] R. Zéboudj — Filtrage, Seuillage automatique, Contraste et Contours : du prétraitement à l'analyse d'image, Thèse de l'Université de Saint-Etienne, (1988).

Annexe A

- [AURE91] F. Aurenhammer — Voronoi, diagrams. A survey of a fundamental geometric data structure, *ACM Computing surveys*, vol. 33, n°3, 345-405, (1991).
- [BERT94] E. Bertin — Diagrammes de Voronoï 2D et 3D : application en analyse d'images, Thèse de l'Université J. Fourier, Grenoble, (1994).
- [CHAS91] J. M. Chassery, A. Montanvert — Géométrie discrète en analyse d'images, *Hermès*, Paris, (1991).
- [EDEL88] H. Edelsbrunner — Algorithms in combinatorial geometry, Tests and monographs in Computer Science, *Springer-Verlag*, New-York, (1988).
- [GOND79] M. Gondran, M. Minoux — Graphes et algorithmes, *Eyrolles*, (1979).
- [GREE78] P.J. Green, R. Sibson — Computing Dirichlet tessellation in the plane, *Computer Journal*, vol. 21, 168-173, (1978).
- [LEE80] D.T. Lee — Two-dimensional Voronoi diagrams in Lp-metric, *Journ. of ACM*, vol. 27, n°4, 604-618, (1980).
- [OHYA84] T. Ohya, M. Iri, K. Murota — A fast Voronoi diagram algorithm with quaternary tree bucketing, *Information Processing Letters*, vol. 18, 227-231, (1984).
- [PREP88] F. Preparata, M.I. Shamos — Computational geometry, an introduction, *Springer-Verlag*, Heidelberg, (1988).
- [SHAM75] M.I. Shamos — Gometric complexity, *7th ACM Symp. on theory of computing*, 224-233, (1975).
- [THIE92] E. Thiel, A. Montanvert — Etude et amélioration des distances du chanfrein pour l'analyse d'images, *Technique et science informatiques*, vol. 11, n°4, 9-41, (1992).
- [TOUS85] G.T. Toussaint — Computanional geometry, machine intelligence and pattern recognition, *North-Holland*, (1985).

Annexe B

- [AZEN87] R. Azencott — Image analysis and Markov fields, *Proc. of ICIAM*, Paris, (1987).
- [BESA74] J. Besag — Spatial interaction and the statistical analysis of lattice systems, *J. of Royal Statistical Society*, 2, 192-236, (1974).
- [BESA86] J. Besag — On the statical analysis of dirty pictures, *J. of Royal Statistical Society*, series B, 48, (1986).
- [CELE88] G. Celeux, J. Diebolt — A random imputation principle : the stochastic E.M. algorithm, I.N.R.I.A., rapport interne n°901, (1988).

- [CHAL88] B. Chalmond — Image restauration using an estimated Markov model, *Signal Processing*, n°15, 115-129, (1988).
- [DEMP77] A. Dempster, N. Laird et D. Rubin — Maximum likelihood from incomplete data via the EM algorithm, *J. of Royal Statistical Society*, series B, 39(1), 1-38, (1977).
- [GEIG91] D. Geiger et F. Girosi — Parallel and deterministic algorithms for MRFs : surface reconstruction, *IEEE Trans. on PAMI*, vol. 13, 401-412, (1991)
- [GEMA84] S. Geman et D. Geman — Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. on PAMI*, vol. 6, 721-741, (1984).
- [GEMA87] S. Geman et C. Graffigne — Markov random fields and image models and their application to computer vision, *Proc. ICM86*, Ed A.M. Gleason, Math. Soc. Providence, (1987).
- [GUYO93] X. Guyon — Champs aléatoires sur réseaux : modélisations, statistique et applications, *Masson*, Paris, (1993).
- [KIRK83] S. Kirkpatrick, C. Gelatt et M. Vecchi — Optimization by simulated annealing, *Science*, 220, 671-680, (1983).
- [METR53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller — Equations of state calculations by fast computing machines, *J. Chem. Phys.*, 21, 1087-1091, (1953).
- [PARI88] G. Parisi — Statistical Field Theory, *Addison Wesley*, (1988).
- [VANL87] P. Van Laarhoven, E. Aarts — Simulated annealing : Theory and Applications, *Reidel Pub.*, (1987).
- [YOUN89] L. Younes — Parameters estimation for imperfectly observed gibbsian fields, *Prob. Theory and Rel. Fields*, 82, 625-645, (1989).

Annexe C

- [COST89] M. Coster et J-L. Chermant — Précis d'analyse d'images, *CNRS*, Paris, (1989).
- [GOLA69] M.J.E. Golay — Hexagonal parallel pattern transformation, *IEEE Trans. on Comput.*, vol. 18, 733-740, (1969).
- [MATH69] G. Matheron — Théorie des ensembles aléatoires, Cahiers du Centre de Morphologie Mathématique, ed *Ecole des Mines de Paris*, fasc. 4, (1969).
- [MATH75] G. Matheron — Random Sets and Integral Geometry, *Wiley*, New York, (1975).
- [MATH83] G. Matheron — Filters and lattices, N 851, *C.G.M.M.*, (1983).
- [SERR82] J. Serra — Image analysis and Mathematical Morphology, *Academic Press*, London, (1982).

