



Université catholique
de Louvain

**LABORATOIRE DE TELECOMMUNICATIONS
ET TELEDETECTION**

B-1348 LOUVAIN-LA-NEUVE

BELGIQUE

CENTRE DE MORPHOLOGIE MATHEMATIQUE

F-77300 FONTAINEBLEAU

FRANCE



Ecole nationale supérieure
des mines de Paris

Traitement d'images numériques au moyen
d'algorithmes utilisant la morphologie
mathématique et la notion d'objet :
application au codage

M. Van Droogenbroeck

Thèse présentée en vue de l'obtention du grade de Docteur en Sciences Appliquées

Jury composé de :

MM.	Paul	DELOGNE	<i>Promoteur</i>
	Auguste	LALOUX	<i>Examineur</i>
	Benoît	MACQ	<i>Examineur</i>
	Christian	RONSE	<i>Examineur</i>
	Jean	SERRA	<i>Examineur</i>
	Fernand	VAN DE WIELE	<i>Président</i>

Mai 1994

Table des matières

Introduction	3
1 Représentation d'image	5
1.1 Description du cadre de travail	5
1.1.1 Présentation des techniques de traitement d'image orientées par la notion d'objet	5
1.1.2 Vers un codage d'image orienté par la notion d'objet : bref historique	6
1.1.3 Description de notre schéma de codage orienté par la notion d'objet	8
1.1.4 Cadre mathématique et notations	11
1.2 Représentation d'image	12
1.2.1 Définition	12
1.2.2 Étude de représentations d'image par objet	13
1.2.3 Mouvement d'objets	17
Conclusions du chapitre	18
 I Description d'outils adaptés au traitement d'objet et seg- mentation d'image	 19
2 Morphologie mathématique	23
2.1 Introduction à la morphologie mathématique	23
2.1.1 Rappels sur les notions d'ensembles	24
2.1.2 Propriétés de base des transformations morphologiques	25
2.2 Transformations morphologiques élémentaires	26
2.2.1 Érosion	26
2.2.2 Dilatation	27
2.2.3 Propriétés des opérations d'érosion et de dilatation	28
2.2.4 Ouverture	29
2.2.5 Fermeture	29
2.2.6 Propriétés de l'ouverture et de la fermeture	30
2.2.7 Propriétés générales	30
2.3 Géodésie et reconstruction	35
2.3.1 Dilatation géodésique	36
2.3.2 Reconstruction	37
2.4 Segmentation d'image par ligne de partage des eaux	38
2.4.1 Que penser de la ligne de partage des eaux?	38
Conclusions du chapitre	40

3	De la représentation au codage des textures	41
	Introduction	41
3.1	Représentation des textures	42
3.1.1	Représentation d'une texture dans le domaine spatial	42
3.1.2	Représentation de la texture dans le domaine spectral	42
3.2	Extrapolation de textures	43
3.2.1	Objectifs	43
3.2.2	Éléments de théorie	47
3.3	Description d'un opérateur d'extrapolation itératif général par représentation matricielle	51
3.3.1	La matrice S , cœur de l'algorithme : vers un filtrage adaptatif? . . .	53
3.3.2	Convergence et unicité de la déconvolution sélective	54
3.3.3	Un pré-traitement souhaitable : recherche du maximum de la fonction d'autocorrélation	55
3.3.4	Algorithmes hiérarchiques de déconvolution	58
3.3.5	Exemples d'utilisation de l'extrapolation	61
3.4	Codage des textures	65
3.4.1	Codage par l'envoi de positions et d'amplitudes des coefficients transformés	67
3.4.2	Codage par prédiction et DCT	71
	Conclusions du chapitre	72
4	Segmentation d'image	75
	Introduction	75
4.1	Présentation du problème	76
4.1.1	Synthèse de la littérature consacrée à la segmentation	76
4.1.2	Vers un modèle d'image	79
4.1.3	Description d'un schéma de segmentation	80
4.2	Détection des régions texturées	81
4.2.1	Le principe	81
4.2.2	Expression des filtres de Gabor	82
4.2.3	Détection des textures : description, détails pratiques et résultats . .	84
4.3	Recherche des régions actives	90
4.4	Segmentation des régions à variations lentes	93
4.5	Étude du mouvement	96
	Conclusions du chapitre	96
II Traitement de la forme des objets : filtrage et codage des contours		101
5	Algorithmes rapides et réflexions fondamentales	105
	Introduction	105
5.1	Définitions	106
5.2	Algorithmes connus	107
5.3	Érosion et ouverture d'images numériques et de labels par utilisation de l'histogramme local	108

5.3.1	Calcul de l'érosion par histogramme local	110
5.3.2	Réalisation de l'ouverture en un seul balayage	111
5.3.3	Résultats	112
5.3.4	Discussion	116
5.4	Ouverture d'images binaires et de labels par propagation de l'élément structurant	117
5.4.1	Le principe	118
5.4.2	Détails de l'algorithme	119
5.4.3	Exemple d'ouverture par propagation	121
5.4.4	Résultats expérimentaux	124
5.5	À propos de l'implémentation d'opérateurs morphologiques composés . . .	127
5.5.1	Développements théoriques	128
5.5.2	Une illustration: implémentation des opérations morphologiques linéaires	134
	Conclusions du chapitre	136
6	Morphologie à seuillage et polymorphisme	139
	Introduction	139
6.1	Morphologie binaire à seuillage	140
6.1.1	Érosion partielle	142
6.1.2	Dilatation partielle	143
6.1.3	Correspondance entre les opérations d'érosion partielle et de dilatation partielle	144
6.1.4	Les opérateurs composés: pseudo-ouverture et pseudo-fermeture . .	145
6.2	Illustrations des opérations partielles: détection des pointes d'un objet, lissage des contours et filtrage d'images numériques	146
6.2.1	Détection des pointes d'un objet	147
6.2.2	Algorithmes de lissage des contours	149
6.2.3	Filtrage d'images en niveaux de gris	152
6.3	Le polymorphisme, un nouvel outil de description géométrique: application au lissage des contours	155
6.3.1	Définition du polymorphisme	155
6.3.2	Propriétés du polymorphisme binaire	158
6.3.3	Comment calculer le polymorphisme?	161
6.3.4	Application au filtrage	162
	Conclusions du chapitre	164
7	Représentation et codage des contours	165
	Introduction	165
7.1	Éléments de modélisation	166
7.2	Description de représentations de formes	167
7.2.1	Les représentations externes	167
7.2.2	Les représentations internes	168
7.3	Notre représentation des contours: description de la forme par une famille d'éléments structurants avec recouvrement	172
7.3.1	Représentation fondamentale d'un objet unique: description de la forme par une famille d'éléments structurants	172

7.3.2	Représentation d'une collection d'objets	174
7.4	Simplification de la représentation	174
7.5	Réduction de la représentation : vers une représentation semi-externe auto- risant le recouvrement de régions	179
7.5.1	Objectifs de la réduction	179
7.5.2	Description interne ou externe?	179
7.5.3	Lignes directrices	179
7.5.4	Résumé de la méthode et résultats	188
	Conclusions du chapitre	190
	Conclusions et perspectives	195
III	Annexes	197
A	Convergence de l'algorithme de déconvolution	199
A.1	Espaces métriques, applications et points fixes	199
A.2	Thèse	201
A.3	Démonstration	201
B	Morphologie mathématique à seuillage	205
	Introduction aux développements théoriques	205
B.1	Étude des propriétés élémentaires des opérations partielles	206
B.2	Ajout d'une mesure : la notion de volume	215

Liste des figures

1.1	Comparaison d'images codées dans le domaine des hauts débits.	9
1.2	Un schéma de codage orienté par la notion d'objet.	10
1.3	Un exemple de texture (d'après LANTUÉJOUL).	14
1.4	Étude de la sensibilité de notre méthode de codage orientée par la notion d'objet en fonction du taux de compression.	16
2.1	Érosion de X par un disque B	27
2.2	Dilatation de X par un disque B	28
2.3	Ouverture et fermeture de X par un disque B	30
2.4	Deux hypothèses physiques pour traiter les objets qui touchent le bord de l'image.	33
2.5	Comparaison de l'érosion de X par un disque centré B pour les deux hypothèses physiques.	34
2.6	Pour illustrer que l'ouverture n'est pas unique.	34
2.7	Dilatation géodésique de taille 1 d'un ensemble.	36
2.8	Extraction de particules pré-sélectionnées dans une image.	37
2.9	Minima, bassins versants et ligne de partage des eaux.	39
2.10	Construction d'un barrage au raccord de deux bassins versants.	39
3.1	Exemple d'extrapolation d'un signal. L'image de droite est l'image extrapolée avec quelques coefficients spectraux seulement (3%).	45
3.2	Un signal unidimensionnel dans le domaine spatial et dans le domaine spectral.	49
3.3	Récapitulatif des techniques de caractérisation des signaux.	50
3.4	Bloc-diagramme de l'algorithme d'extrapolation par déconvolution sélective, proposé par FRANKE.	54
3.5	Évolution du spectre d'un signal unidimensionnel durant un processus de déconvolution sélective.	55
3.6	Premières étapes de l'algorithme de déconvolution sélective.	56
3.7	Fonction d'autocorrélation d'une image.	58
3.8	Influence de la taille sur le résultat de la déconvolution.	58
3.9	Extrapolation de signaux issus d'une décomposition en sous-bandes.	60
3.10	Segmentation d'une texture par croissance d'un germe.	62
3.11	Comparaison de l'approximation d'un signal sans ou avec extrapolation préalable.	64
3.12	Une texture et son approximation avec un facteur de sélection de 100.	68
3.13	Évolution du taux de compression (sans contour) et du SNR en fonction du facteur de sélection.	70
3.14	Effet de la phase dans une DCT.	71

3.15	Codage d'une texture par prédiction des blocs 8×8 à venir.	72
3.16	Deux exemples de synthèse d'une texture à partir d'un échantillon.	73
3.17	Synthèse d'une texture aléatoire par extrapolation d'un échantillon.	74
4.1	Les trois types de régions du modèle d'image.	80
4.2	Algorithme de détection d'objets en trois phases.	82
4.3	Processus de détection des germes de texture.	84
4.4	Trace d'un filtre de Gabor dans le plan transformé.	86
4.5	Premières étapes de la détection des textures: un premier exemple.	87
4.6	Premières étapes de la détection des textures: un second exemple.	88
4.7	Processus itératif de croissance pour la segmentation de textures.	90
4.8	Initialisation du processus de croissance d'un germe de texture.	91
4.9	Quelques résultats de la détection des textures par croissance de germes.	92
4.10	Initialisation de la segmentation des régions à variations lentes.	94
4.11	Détection et croissance des régions à variations lentes.	95
4.12	Analyse spectrale des textures extrapolées pour la reconnaissance d'objets dans des séquences d'images.	97
4.13	Synthèse des étapes intervenant dans le procédé de segmentation. Les parties itératives sont indiquées en italique.	98
4.14	Résumé des principales étapes du procédé de segmentation.	100
4.15	Schéma complet d'un codeur de contours.	104
5.1	Glissement d'une fenêtre et histogramme local associé à la fenêtre initiale.	110
5.2	Illustration de deux situations distinctes rencontrées lors de la réalisation de l'ouverture en un seul passage.	112
5.3	Images à teintes de gris utilisées dans les essais.	113
5.4	Comparaison d'algorithmes d'érosion.	114
5.5	Comparaison d'algorithmes d'érosion morphologique.	115
5.6	Comparaison d'algorithmes d'ouverture morphologique.	116
5.7	Étude du coût de l'érosion partielle (échelle semi-logarithmique).	118
5.8	Un ensemble X , un élément structurant B et les points d'ancrage possibles.	119
5.9	Exemple d'un jeu de directions de propagation.	120
5.10	Exemple d'ouverture par propagation d'un élément structurant.	122
5.11	Temps de calcul de deux algorithmes d'ouverture morphologique par un carré sur une image binaire complexe et de grande taille.	125
5.12	Temps de calcul de deux algorithmes d'ouverture morphologique par un carré sur une image binaire simple.	126
5.13	Images de labels utilisées dans les essais.	127
5.14	Temps algorithmiques de plusieurs procédés d'ouverture d'une image de labels simple par un carré (échelle semi-logarithmique).	127
5.15	Temps algorithmiques de plusieurs procédés d'ouverture d'une image de labels complexe par un carré.	128
5.16	Deux paires d'ensembles qui vérifient la condition de réduction.	131
5.17	D_α intersecte ∂B à plusieurs endroits.	132
5.18	Ici, $\partial B \oplus B = ([a, b] \cup \{c\}) \oplus B$	133
5.19	Pour montrer que B et C ne sont liés par aucune relation d'inclusion, même si $A \oplus C \supseteq B \oplus C$ et $A \subseteq B$	134

5.20	Temps de calcul pour différentes opérations morphologiques par un segment sur une grande image.	135
6.1	Comparaison de deux différences ensemblistes.	148
6.2	Détection des parties anguleuses convexes d'un objet par application du chapeau haut-de-forme partiel.	148
6.3	Influence de la forme d'un élément structurant dans l'élaboration d'un filtre.	152
6.4	Images filtrées pour une comparaison de leur représentation.	153
6.5	Traitement d'une image en niveaux de gris par un filtre partiel et comparaison avec des filtres alternés séquentiels.	154
6.6	Interprétation du polymorphisme.	156
6.7	Polymorphisme des points d'une région pour un élément structurant carré 3x3.	156
6.8	Le polymorphisme des points d'une image de labels.	157
6.9	Marquage des points à polymorphisme élevé dans une image à teintes de gris.	159
6.10	Placement des axes de $\mathbb{R}^2$	160
6.11	Comparaison des contours d'images de labels filtrées par application du polymorphisme.	163
7.1	Une image de labels et le graphe qui modélise l'organisation spatiale des régions.	167
7.2	Le squelette de quelques figures simples.	169
7.3	Comparaison de deux approximations géométriques d'un rectangle par des disques.	171
7.4	Le recouvrement des boules maximales du squelette.	172
7.5	Détermination des points essentiels de l'érodé pour remplir l'ouvert.	178
7.6	Exemple de description semi-externe d'une région.	180
7.7	Comment déterminer le nombre de points frontières correspondant à chaque élément de l'érodé?	183
7.8	Représentation de la forme par squelette propagé.	193
B.1	Pour illustrer que $(X \overset{m}{\ominus} B) \overset{n}{\ominus} B \neq (X \overset{n}{\ominus} B) \overset{m}{\ominus} B$	215
B.2	Pourquoi le concept de 'taille morphologique partielle' n'est pas possible.	216
B.3	Illustration de l'effet d'une érosion partielle volumique suivie d'une dilatation morphologique.	219

Liste des tableaux

3.1	Relevé de la valeur moyenne des erreurs absolues pour différents signaux obtenus en ne sélectionnant qu'un certain nombre de coefficients spectraux.	63
3.2	Moyennes des erreurs absolues.	66
3.3	Temps de calcul en secondes pour une déconvolution des bandes de schémas hiérarchiques.	66
3.4	Comparaison des temps de calcul (en secondes) de deux architectures. . . .	67
6.1	Tableau comparatif de représentations d'images filtrées.	152
7.1	Description d'une famille d'éléments structurants et nombre d'occurrences de chacun d'eux dans la représentation réduite de l'essai.	190
7.2	Coût de codage des composantes de notre représentation.	190

Avant-propos

Ce travail est le fruit de trois années de recherches menées conjointement au Laboratoire de télécommunications et télédétection de l'Université catholique de Louvain, et au Centre de morphologie mathématique de l'École des mines de Paris, grâce à la bienveillance de Messieurs Delogne, Meyer et Serra. Qu'ils trouvent en ces mots l'expression de ma gratitude.

En tant que boursier IRSIA, je remercie l'Institut pour l'encouragement de la recherche scientifique dans l'industrie et l'agriculture, ainsi que Sylvain d'Agostino et Bruno Rahir pour leur soutien.

Ma reconnaissance va à Paul Delogne, promoteur de ce doctorat, et aux autres membres du jury : Auguste Laloux, Benoît Macq, Christian Ronse, Jean Serra et Fernand Van de Wiele. Je tiens aussi à souligner l'importance de l'aide apportée par Benoît Macq, par Jean Serra et par Christian Ronse, que je remercie tout particulièrement pour m'avoir guidé durant ces trois années. Le concours de Christian Ronse a été plus qu'appréciable et apprécié !

Merci aussi à tous les amis rencontrés dans les deux laboratoires : Amal Ibrahim, Benoît Maison, Bernard Raquet, Christian Lantuéjoul, à la fois un ami et un conseiller, Dmitry Gorokhovik, Fernand Meyer, Hugues Talbot, l'autre 'NeXTman' qui m'a si souvent dépanné, Jean-François Rivest, l'inventeur de Xlim, Jose Crespo, Laura Andriamasinoro, Laurent Cuvelier, Liliane Pipault, Luc Vandendorpe, Marc Waroquier, Michel Bilodeau et Nicole, prêts à m'accueillir à tout moment, Oscar Ribes, Pierre Lemaire, Pierre Soille, avec lequel j'ai eu tant de discussions intéressantes, Renaut Ryckaert, René Peyrard, Roland Bremond, Ronald Jones, Serge Comes et Stéphane Maes.

J'adresse mes remerciements sincères à Michel Bilodeau et à Pierre Soille, pour la relecture du document et pour tous les bons moments passés ensemble. Merci aussi à Liliane Pipault pour sa bonne humeur communicative et sa gentillesse, et à Renaut Ryckaert.

Merci enfin à mes parents qui m'ont tant aidé et soutenu, et à Linda que j'aime.

Marc Van Droogenbroeck
Mai 1994

Introduction

Plan de la thèse

Le premier chapitre expose les objectifs visés par le travail de doctorat. À la base se trouve le principe qui sous-tend tous les développements : le nouveau concept de *représentation d'image* est une sorte de langage de haut niveau destiné à une description précise et rigoureuse de tous les traitements orientés par la notion d'objet, opérés sur l'image.

La première étape importante est la détection des objets dans l'image, opération appelée *segmentation*. Sa réalisation pratique fait intervenir plusieurs outils appartenant à deux grandes familles : la *morphologie mathématique* et l'*extrapolation de textures*. La première décrit principalement les formes des objets ; elle est développée dans le chapitre 2. La seconde sert à analyser les signaux de texture. Le chapitre 3 énonce les principes théoriques de l'extrapolation et aboutit à des tests d'efficacité de codage. Il ressort des essais que le taux de compression est inversement proportionnel au nombre de raies spectrales conservées pour représenter la texture. Pour des grandes textures, il atteint aisément la valeur de 50.

La deuxième partie du document est consacrée à l'étude de la forme. Filtrage et codage des contours en constituent l'ossature principale.

En étudiant de près les opérations de filtrage des contours, il est apparu que le temps de calcul constituait une limitation importante pour les techniques de filtrage et de codage développées. Pour lever cette difficulté, nous examinons, dans le chapitre 5, des nouveaux algorithmes adaptés à nos besoins mais dont le champ d'application est plus étendu. Par l'intermédiaire de nos algorithmes, nous sommes même parvenu à réduire le temps de calcul des opérations morphologiques usuelles que sont l'ouverture et la fermeture. Des développements théoriques complètent la description des algorithmes par une réflexion fondamentale.

Le chapitre 6 explique comment se réalise le filtrage des contours. C'est l'aboutissement d'idées visant à assouplir les opérations morphologiques élémentaires. Nous proposons également le nouveau concept de *polymorphisme*, sorte de fonction distance, qui fournit un moyen d'estimer les effets du filtrage.

Le dernier chapitre termine l'étude de la représentation de la forme, intègre les notions de polymorphisme dans une nouvelle description des contours et propose, comme perspective, une représentation par squelette propagé.

Démarche

Il n'est pas vain de rappeler que l'objectif du travail est double : *développer une batterie d'outils orientés par la notion d'objet* et mettre au point une *technique de codage en*

termes d'objets. En filigrane, l'ensemble des travaux se présente comme le développement d'une série d'outils adaptés aux objets. Par contre, en surface, le rôle principal revient à une application particulière qui s'épanouit dans un cadre cohérent : le codage synthétique d'image par objets.

Enfin, le cadre global et la notation commune entre les différentes parties de ce document sont destinés à rassembler une panoplie de nouveaux outils convenant à des objets (représentation d'un objet, extrapolation de la texture d'un objet, filtrage de sa forme, ...) en vue de les appliquer au codage d'image.

Chapitre 1

Représentation d'image

1.1 Description du cadre de travail

Ce chapitre développe et présente un cadre général axé sur la notion d'objet, dans lequel on pourra modéliser toutes les étapes de traitement d'image intervenant dans la construction d'un procédé de codage d'image évolué, qu'il s'agisse par exemple de la segmentation, d'un filtrage non-linéaire ou encore d'une simplification du contenu de l'image. La raison d'une telle entreprise est qu'il n'existe pas à l'heure actuelle, du moins à notre connaissance, de formalisme universellement adopté ; sans doute parce que certains domaines du traitement d'image demeurent fortement cloisonnés. C'est donc un cadre de travail que nous développons dans ce chapitre.

1.1.1 Présentation des techniques de traitement d'image orientées par la notion d'objet

Comme l'univers se compose d'objets, la description précise d'une scène emprunte souvent un vocabulaire lié aux objets. Pourtant, le domaine du traitement d'image y fait rarement allusion. C'est vrai notamment pour le codage d'image où l'on voit a priori mal par quel biais introduire une information en termes d'objet. De plus, les auteurs commettent l'erreur de croire que les objets doivent toujours correspondre à des objets physiques, alors que définir un objet comme une surface présentant une certaine homogénéité spatiale, statistique ou spectrale, est plus utile en traitement d'image.

Une difficulté supplémentaire provient sans doute de l'*inexistence d'un modèle d'image orienté par la notion d'objet* et d'une étude trop succincte des schémas de codage. Pour remédier à ces manquements, nous proposons dans ce travail un modèle d'image axé sur la signature spectrale locale et un schéma de codage entièrement axé sur la notion d'objet. Ce cadre général et cohérent permettra d'intégrer les trois tâches qui incombent à toute technique de codage par objet. En effet, pour ce type de codage il faut

- segmenter l'image, autrement dit identifier les objets présents dans la scène ;
- qualifier les objets, les décrire précisément. Le concept de représentation introduit plus loin y pourvoit ;

- puis il appartient de coder ces objets de la manière la plus compacte possible. Cette fonction est le fruit d'un schéma de codage.

Ces trois points sont abordés dans le présent document sous un angle différent de celui des travaux présentés dans la littérature, puisque nous avons veillé à adopter une démarche globale depuis l'étape de segmentation jusqu'au codage. Si le thème pivot de ce travail est le codage, la mise au point de notre technique de codage nécessiterait bien des études complémentaires. C'est pourquoi *nous ne dessinerons que le fil conducteur du codage*, émaillé toutefois de résultats complets dans ce chapitre, *en mettant l'accent plutôt sur des nouvelles techniques de traitement d'image orientées par la notion d'objet*.

1.1.2 Vers un codage d'image orienté par la notion d'objet : bref historique

Représenter une image numérique nécessite un nombre élevé de bits. Dans l'avis 601, le CCIR définit une norme à 216 Mbit/s. Les canaux de transmission actuels ne s'accommodent pas de tels débits. L'objectif du codage d'images est de réduire les débits tout en demeurant capable de reconstruire à la réception une image aussi proche que possible de l'image de départ.

Les premières techniques de codage se sont inspirées de la théorie de l'information. Deux groupes de techniques de codage d'images fixes se sont développés depuis les années septante : les techniques de codage spatiales, qui travaillent directement à partir des échantillons de l'image, et les techniques de codage par transformées. À cela s'ajoute encore la compression hybride d'images qui combine les deux schémas de codage précédents.

Au début, ces méthodes fournissaient des taux de compression qui ne dépassaient guère 10. Moyennant la suppression d'informations peu visibles par l'œil, ce taux a été légèrement accru ; il s'agit du codage perceptif. En voici la description simplifiée. Chaque bloc de 8x8 pixels découpé dans l'image subit un changement de représentation par transformée en cosinus discrète (DCT¹). Les coefficients obtenus dans l'espace décorrélé se codent ensuite de manière perceptive et passent dans un codeur entropique, tel celui développé par MACQ [55]. Afin de supprimer les effets de blocs inhérents à un découpage en carrés de taille 8, on peut plutôt décomposer le signal en bandes de fréquences. C'est alors le domaine du codage en sous-bandes (cf. VANDENDORPE [109]).

Après avoir exploité toutes les possibilités de la théorie de l'information, ces techniques procurent des taux de compression allant de 5 à 100. Mais pour obtenir des taux aussi élevés, le codage spatial se double d'un codage temporel qui agit par comparaison d'images successives et interprète les mouvements. L'exemple suivant résume un codage complet.

Un exemple : la norme H261

La norme H261 [52] du CCITT reprend les éléments d'un codec² vidéo pour les services audio-visuels à $p \times 64$ kbit/s (où p va de 1 à 30), qui s'appuie sur le format d'image CIF³ de 288 lignes et 360 pixels par ligne. La configuration du codec comprend une structure hybride alliant un codage spatial par DCT à un codage prédictif inter-images pour réduire

1. Discrete Cosine Transform

2. Abréviation pour codeur-décodeur

3. Common Intermediate Format

la redondance temporelle. Les erreurs de troncature qui apparaissent lors du calcul de la transformée dans le codeur et le décodeur, ainsi que les erreurs de transmission, introduisent, à cause du traitement décorrélatif temporel, une distorsion qui croît dans le temps, ce qui nécessite un rafraîchissement régulier de l'image.

D'autres standards ont vu le jour récemment : JPEG⁴ [72, 112] pour des images fixes, MPEG⁵-1,2 pour des séquences. Par ailleurs, l'ambitieux programme MPEG-4 est en cours.

Ces schémas de codage établis considèrent peu les particularités du système visuel humain et ne conviennent pas à l'interprétation robotique d'image. Poussées par le désir de se rapprocher d'une vérité physiologique et d'effectuer des traitements automatiques plus efficaces, quelques recherches ont exploré des pistes centrées sur le concept d'objet. L'intérêt du codage par objet est d'obtenir des images dépouillées mais contenant une information *intelligente*, c'est-à-dire comprenant toutes les caractéristiques importantes pour l'application désirée ; le codage par objet réalise pour ainsi dire une *synthèse* du contenu de l'image.

KUNT [43, 44] est sans doute le premier à avoir contribué à diffuser le codage par objet, qu'il nomme *codage de seconde génération*. Il distingue deux familles de techniques de codage par objet : les premières décomposent l'image en canaux de fréquences orientés puis codent l'information des contours (position et amplitude), les autres traitent des régions fermées, les textures étant représentées par des polynômes ou des fonctions de base orthogonales [29]. Au registre des sources d'inspiration, on retrouve le domaine de la synthèse ou de la discrimination de textures qui alimente la recherche de techniques de codage de textures et surtout de segmentation. Ainsi BARNSELY s'appuie sur la notion de fractal pour décrire les textures et parvenir à des hauts taux de compression (quoique la confirmation des valeurs annoncées tarde à venir...). D'autres auteurs modélisent les textures par des fonctions aléatoires (voir par exemple HARALICK [32] ou HSIAO *et al.* [35]). Par ailleurs, la Communauté européenne finance depuis 1992 le projet de recherches RACE Morpheco destiné à développer une technique de codage d'image par objet.

Dans l'état actuel des connaissances, le codage par objet est à considérer comme un complément potentiel des techniques éprouvées ou comme substitut pour des applications spécifiques, telle la vidéotéléphonie à très bas débit, qui privilégient la notion d'objet sans exiger la reproduction exacte de l'image.

Les méthodes par objet : complément des techniques actuelles

En couplant un algorithme axé sur la notion d'objet à un codec 'classique', certains défauts disparaîtraient :

- quelques modes de fonctionnement définis dans la norme JPEG permettent une reconstruction progressive de l'image. À cette fin, l'aide qu'apporte un algorithme qui détecte les objets et les classe par ordre d'importance, pourrait conduire à allouer les bits disponibles aux zones d'image à restituer préalablement ;
- un algorithme par objet peut servir à supprimer les effets de blocs qui dérivent du traitement par DCT ou lors de la compensation de mouvement.

4. Joint Photographic Experts Group

5. Motion Picture Experts Group

Ces deux modes d'action montrent les méthodes de traitement d'image orientées par la notion d'objet, analysant le contenu de l'image et influençant le comportement d'un codeur. Il est vrai qu'après l'analyse de l'image, on peut espérer optimiser le comportement du codeur.

Le codage basé sur la notion d'objet : substitut aux techniques actuelles

Deux inconvénients majeurs dus principalement à la volonté de conserver toute l'information contenue dans l'image, propre au codage à haut débit, sont apparus lors des premières tentatives de réalisation des codages par objet :

- pour le codage d'images naturelles, les résultats affichés jusqu'à présent [43, 50] ne parviennent pas à concurrencer les algorithmes plus anciens. À qualité égale, il n'est pas certain qu'un codage par objet s'avère plus efficace. C'est le sévère bilan que dressent BIGGAR *et al.* [2] ;
- les résultats dépendent fortement du contenu de l'image à traiter, conclusion que tendent à confirmer les résultats obtenus dans le cadre du projet RACE Morpheco [4]. Les techniques par objet sont moins universelles.

Dès lors, l'objectif n'est pas de restituer après décodage une image identique à l'originale, mais d'extraire de l'image une information en termes d'objets, capable de produire une image simplifiée. Le domaine des faibles débits se prête probablement davantage à l'usage d'un codec axé sur la notion d'objet. Pour l'illustrer, nous comparons une image codée par JPEG et par notre méthode à la figure 1.1. Les effets de bloc affectent l'image (b) à tel point que, sans post-traitement, ils limitent l'usage de la norme à des taux de compression de l'ordre de 30. L'image (d), codée par notre technique, présente plutôt un effet de lissage, moins gênant dans ce cas.

1.1.3 Description de notre schéma de codage orienté par la notion d'objet

Avant de poursuivre et afin de clarifier la démarche, nous présentons à la figure 1.2 le schéma de codage général utilisé pour nos travaux.

Un premier opérateur décompose l'image en objets décrits par leur forme ou *contour* et leur aspect appelé *texture*. Ces deux informations sont ensuite codées, transmises ou stockées, et enfin décodées au récepteur.

Une remarque s'impose à la vue de ce schéma : les contours et les textures sont deux types d'information pas toujours très bien décorréliées entre elles. Comme le précise la théorie de l'information, plus les informations de contour et de texture sont décorréliées, plus le code est compact. Alors, dans la mesure du possible et pour arriver à des taux de compression importants, on récupérera le contenu d'une branche pour injecter l'information dans l'autre branche. Dans la figure, c'est l'information de contour qui est introduite dans la branche de codage des textures. Cette constatation influe considérablement sur les développements d'algorithmes de codage. C'est elle qui justifie *l'extrapolation* de textures, encore appelée *déconvolution*, dont il sera question au chapitre 3. Elle est destinée à simplifier les textures d'une partie de l'information résiduelle de contour. Réutiliser l'information



(a) Image originale.



(b) Image codée par JPEG avec un taux de compression de 34.



(c) Segmentation complète : étape préliminaire d'un codage par objet.



(d) Le résultat de notre technique de codage à partir de (c). Le taux de compression vaut 37.

FIG. 1.1 - *Comparaison d'images codées dans le domaine des hauts débits.*

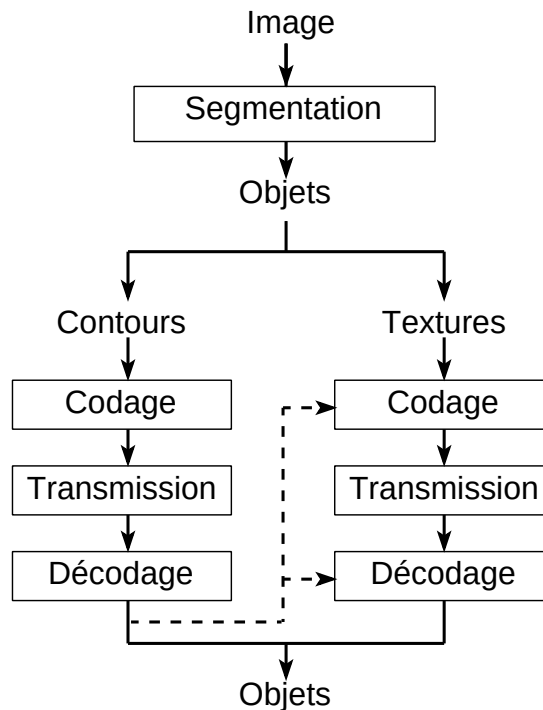


FIG. 1.2 - Un schéma de codage orienté par la notion d'objet.

d'une branche dans l'autre est la principale innovation du schéma. Bien sûr, un tel schéma exige que soit transmise en premier lieu l'information des contours mais rien n'interdit cette hypothèse.

Les quelques techniques de codage par objet déjà indiquées dans la littérature (cf. [43] par exemple) n'établissent aucune connexion entre les deux branches du codeur. Pourtant, dans la plupart des schémas publiés, le récepteur doit d'abord décoder complètement un signal pour arriver à décoder le signal de l'autre branche, ce qui signifie qu'une forte corrélation subsiste entre les deux branches sans qu'elle ne soit considérée durant le codage.

Théoriquement, il est aussi possible d'inverser le rôle des contours et des textures en récupérant à cette fois l'information décodée de texture. Il n'y a guère que si le récepteur doit recevoir simultanément les deux types d'information, qu'on évitera un schéma en boucle ouverte.

Hypothèse importante pour le codage des textures. Nous conserverons tout au long du travail le schéma de la figure 1.2, c'est-à-dire que l'information de contour servira à décorréler davantage les entités contour et texture et qu'*au moment de décoder les textures, le récepteur possède déjà l'information de contour*.

Nous allons à présent introduire un cadre mathématique et une notation pour les traitements d'image par objet.

1.1.4 Cadre mathématique et notations

Soit une image I à niveaux de gris à traiter. Nous parlerons indifféremment d'*image à niveaux ou teintes de gris*, de *fonction* ou d'*image numérique*. Les opérations entreprises en traitement d'image sont nombreuses et parfois fort différentes. L'application qui nous concerne plus particulièrement est le codage d'image par objet. Si la partie de codage entropique supprime la redondance présente dans une image sans dégradation, la quantification introduit des irréversibilités de sorte que l'image décodée est différente de l'image de départ et cela d'autant plus que les débits sont faibles. Présenté de la sorte, le codage opère une transformation de suppression de redondance d'information et fournit en fin de compte une *approximation de l'image* I notée $\pi(I)$.

Si l'écart entre l'image I et son approximation $\pi(I)$ est faible, la technique de codage est de bonne qualité. Idéalement, il y a égalité 'perceptive' parfaite entre l'image et sa représentation, c'est-à-dire qu'un observateur ne perçoit aucune différence entre les deux.

À ce stade de la discussion, il est utile de fournir un cadre mathématique plus rigoureux.

Cadre mathématique. Chaque image de luminance I définit de manière unique une application ni injective ni surjective qui, à chaque point du plan E (il s'agit du plan discret $\mathbb{Z}^2$ ou euclidien $\mathbb{R}^2$)⁶, associe une valeur de gris comprise dans un intervalle donné $\mathcal{G}$:

$$I : E \rightarrow \mathcal{G} : (x, y) \rightarrow I(x, y) \quad (1.1)$$

Si la position (x, y) est fixée, une nouvelle série d'applications à valeurs dans $\mathcal{G}$ s'obtient en définissant un opérateur ϕ qui transforme une image en une autre image :

$$\phi : \mathcal{G}^E \rightarrow \mathcal{G}^E : \forall (x, y) \in E, I(x, y) \rightarrow \phi(I(x, y)) \quad (1.2)$$

L'opérateur *approximation* est une application de ce type, tout comme l'est une opération de filtrage. L'*opérateur identité* noté 1 envoie un ensemble sur lui-même : $1(I) = I$.

Il arrive que l'ensemble de définition et l'ensemble image soient distincts. La définition de composition d'opérateurs reste valable à condition que l'ensemble d'arrivée du premier opérateur concorde avec l'ensemble de départ du second opérateur. Les opérateurs de représentation et de reconstruction décrits plus loin constituent une paire d'opérateurs appartenant à cette famille.

Égalité entre deux opérateurs. L'égalité entre deux opérateurs ϕ et φ se traduit par une égalité en tout point du plan :

$$\phi(I) = \varphi(I) \iff \forall (x, y), \phi(I(x, y)) = \varphi(I(x, y)) \quad (1.3)$$

Avec ces définitions, toute opération intervenant dans la formation d'une technique de codage acquiert une cohérence mathématique à partir de laquelle il est possible de l'étudier en détail.

6. Pour la commodité des démonstrations, nombreux sont les auteurs qui recourent à $\mathbb{R}^2$ systématiquement. Ils perdent parfois de vue que la discrétisation de l'image altère la nature de la modélisation et escamote ainsi les considérations.

Une mesure de la fidélité de l'approximation $\pi(I)$ à l'image I est par exemple la valeur moyenne des écarts absolus

$$\zeta(I, \pi) = \frac{\sum_x \sum_y |I(x, y) - \pi(I(x, y))|}{M_i M_j} \quad (1.4)$$

où $M_i M_j$ représente la taille de l'image. C'est la mesure que nous utiliserons pour présenter les résultats de l'extrapolation de textures.

L'approximation est parfaite si, en tout point, $\pi(I) = 1(I) = I$; autrement dit, il s'agit de l'opérateur identité.

D'autres types d'images : les images binaires et les images de labels. Lorsque $\mathcal{G}$ ne contient que deux valeurs, l'application agit comme une indicatrice : elle indique si un point appartient ou non à un ensemble. Nous parlerons alors indifféremment d'*image binaire* ou d'*ensemble*. Si les niveaux de gris, plus nombreux, reflètent l'appartenance à des régions et que la relation d'ordre ne joue aucun rôle, il s'agit d'*images de labels*.

1.2 Représentation d'image

1.2.1 Définition

Le codage que nous désirons synthétiser repose sur la notion d'objet. Une image est composée d'objets, véritables ensembles de points voisins ayant des propriétés homogènes entre eux. Pour la modélisation, nous introduisons le concept de *représentation d'image* $\rho(I)$ qui, par un procédé de *recomposition* adéquat noté τ , conduira à une forme particulière d'*approximation* $\pi(I) = \tau(\rho(I))$ ⁷. La représentation d'image est le concept-clef de tout algorithme s'appuyant sur la notion d'objet. Elle guide tous les traitements depuis la phase de détection d'objets à celle du codage en passant par l'étude du mouvement.

Insistons sur la nécessité d'adjoindre un procédé de reconstruction τ à toute représentation d'image $\rho(I)$. Cette remarque nous conduit à définir la représentation d'une image.

Définition 1 *En toute généralité, une représentation de l'image I est un outil de modélisation $\rho(I)$, qui par l'intermédiaire d'une paire représentation-reconstruction (ρ, τ) , définit une application d'approximation $\pi = \tau\rho$.*

Toute représentation d'image $\rho(I)$ met certains aspects en lumière. C'est par son intermédiaire que se mène une réflexion d'interprétation de l'image, tantôt en termes de fréquences, tantôt en termes d'objet. Donnons quelques exemples d'approximations π générées par des paires (ρ, τ) .

Exemples :

1. Une opération de filtrage linéaire de l'image se transcrit sans difficulté dans le nouveau formalisme. Pour ce faire, la représentation d'image $\rho(I)$ est constituée de

⁷. $\tau(\rho(I)) \in \mathcal{G}^E$, tout comme I ; ce n'est pas toujours le cas pour $\rho(I)$.

nombres complexes dans le plan transformé. Soient (u, v) les coordonnées du plan transformé et $\mathcal{F}$ la transformée de Fourier de l'image I : $\rho(I) = \{(u, v, \mathcal{F}(u, v))\}$, où

$$(u, v) \in E \text{ et } \mathcal{F}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(x, y) e^{-j(ux+vy)} dx dy$$

Le filtrage s'opère à la reconstruction par un mécanisme à deux niveaux : (i) l'amplitude $\mathcal{F}(u, v)$ est multipliée par la réponse fréquentielle du filtre $\mathcal{H}(u, v)$, (ii) vient ensuite la transformation inverse. Cet exemple montre clairement que si la représentation fixe le cadre de travail, toute liberté est allouée au procédé de reconstruction de choisir la réponse impulsionnelle du filtre.

2. Soit la représentation d'image regroupant les coefficients a_i obtenus par *une transformation orthogonale* de tous les blocs de l'image et associant une position dans le bloc et dans l'image. Cette représentation est l'ensemble des coefficients ordonnés a_i ou encore $\rho(I) = \{(a_i, o_i)\}$ où o_i est un numéro d'ordre. La reconstruction procède primo au positionnement exact de chaque coefficient, secundo à la transformation inverse. La plupart des techniques de codage actuelles (codage sous-bande, codage par transformée en cosinus discrète, ...) s'appuient sur des variantes plus ou moins complexes de l'approximation offerte par cette combinaison $\rho - \tau$. Remarquons la présence d'un concept d'ordre dans l'approximation. Bien des techniques nécessitent un ordre, ce qui n'est pas sans danger car toute perturbation dans l'ordre amène inéluctablement une distorsion de l'approximation. Dans la mise au point d'une technique de codage, il faudra veiller à protéger l'ordre ou à éliminer au mieux toute la partie d'ordre qu'il n'est pas possible de connaître a priori.
3. Une image est représentée par une collection d'objets qui se caractérisent par leur forme X et par leur aspect, leur texture T . Une représentation possible est composée de la réunion de l'information concernant la forme de tous les objets X et leur position dans l'image (x, y) : $\rho(I) = \{(X, x, y)\}$. La recomposition remplace correctement les formes dans l'image, mais l'information de texture a disparu. L'approximation résultante n'est, en toute généralité, pas à reconstruction parfaite. Les images mosaïques illustrent un style de représentation d'image où la seule information de texture disponible est la valeur moyenne de la luminance sur chacune des régions.

1.2.2 Étude de représentations d'image par objet

Représentations

Le procédé de codage exige la connaissance d'une représentation de la forme des objets X et de leur texture T : $\rho(I) = \rho_{X,T}(I)$. À des fins de transmission, c'est cette représentation qui sera codée et envoyée au récepteur après transformation. Caractériser la forme est aisé mais définir la notion de texture relève du défi. D'autre part, segmenter une image n'est pas toujours une mince affaire. La nouvelle définition de texture fournie ci-après met en lumière l'origine de ces difficultés.

Définition 2 *Une texture est un signal qui peut être étendu naturellement au-delà de son domaine de définition.*

La définition d'une texture exige la présence d'une certaine régularité dans le signal, pas nécessairement de caractère périodique, et elle s'appuie sur une extrapolation cohérente. Le signal de la figure 1.3 est un exemple de texture. En voyant ce signal, on comprend que l'extrapolation d'une texture n'est pas toujours unique; elle dépend de la manière d'exprimer cette régularité. Poussé par la volonté de coder efficacement les textures, nous avons choisi un critère de périodicité, équivalent à une interprétation fréquentielle en termes de fréquences dominantes, mais par le fait, nous avons perdu toute généralité.

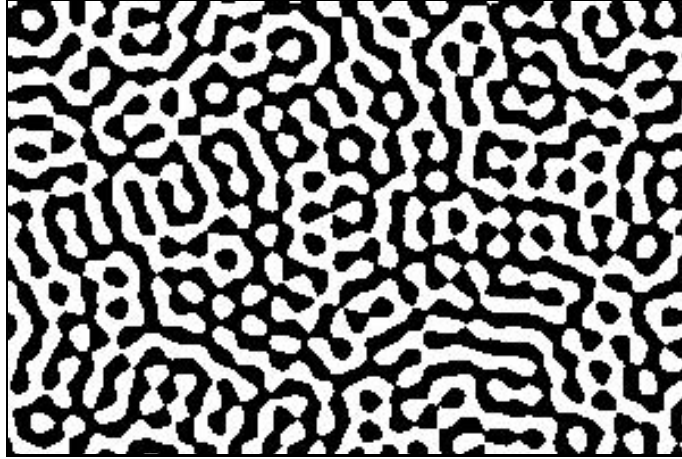


FIG. 1.3 - *Un exemple de texture (d'après LANTUÉJOUL).*

Vu la complexité des textures naturelles, un procédé de segmentation s'avère incapable de les détecter toutes avec précision. Il en résulte que, régulièrement, la décorrélation d'information entre la forme et la texture est imparfaite, tant et si bien que la représentation se compose en toute rigueur de plusieurs parties: $\rho_{X,T}(I) = \rho_{X \setminus T}(I) + \rho_{T \setminus X}(I) + \rho_{X \cap T}(I)$ ⁸ où les composantes sont

- la partie descriptive de forme $\rho_{X \setminus T}(I)$;
- l'information de texture $\rho_{T \setminus X}(I)$;
- une partie commune $\rho_{X \cap T}(I)$.

Pour la commodité de la notation et par souci de clarté, nous noterons $\rho_{X \setminus T}(I)$ par $\rho(X)$, $\rho_{T \setminus X}(I)$ par $\rho(T)$ et, par extension, $\rho_{X \cap T}(I)$ par $\rho(X \cap T)$. La partie commune $\rho(X \cap T)$ de la représentation comprend par exemple la position du coin supérieur gauche du plus petit rectangle circonscrit à chaque objet ou un ordre de description commun. Du point de vue de la démarche, $\rho(X \cap T)$ est la partie d'information extraite d'une branche et injectée dans l'autre branche, opération illustrée par la figure 1.2. Parfois il s'agit d'une part d'information plus difficile à expliciter. En effet, après codage, les représentations $\rho(X)$ et $\rho(T)$ empruntent physiquement ou virtuellement deux canaux distincts. Alors, si le codage de la texture s'effectue par l'envoi de coefficients transformés, il importe de déconvoluer le spectre – autrement dit, d'étendre le signal de texture jusqu'à ce qu'il occupe un domaine rectangulaire – par une fonction qui décrit la forme de l'objet, sinon les coefficients

8. Les symboles $\setminus$ et $\cap$ désignent respectivement la différence et l'intersection ensemblistes.

contiendraient une partie d'information relative aux contours. Or la déconvolution est souvent imparfaite : toute information de texture contiendra résiduellement une information de contour. Nous avons choisi d'inclure implicitement $\rho(X \cap T)$ dans $\rho(X)$. Il s'agit là certes d'un manque de rigueur critiquable mais ce choix privilégie le codage des contours ce qui, dans la pratique et pour des hauts taux de compression, s'explique parce que le débit nécessaire aux contours représente la plus grande part du débit total. Il paraît alors logique de concentrer les efforts sur le codage des contours et de récupérer tant que faire se peut cette information pour les textures. En moyenne, il faut compter *1,3 bit par point de contour pour décrire parfaitement $\rho(X)$* [22], valeur que nous prendrons comme référence.

Recompositions et approximations

L'approximation de l'image s'obtient au récepteur par recomposition des représentations. Elle se dédouble également en parties relatives à la forme et la texture des différents objets. En filigrane sont obtenues deux approximations d'image incomplètes si elles sont prises séparément et bien plus précises si considérées simultanément. Les approximations en termes d'objets et de textures valent respectivement $\pi(X) = \tau(\rho(X) + \rho(X \cap T))$ et $\pi(T) = \tau(\rho(T) + \rho(X \cap T))$, alors que l'approximation d'une image codée par un schéma orienté objet est $\pi(I) = \tau(\rho(X) + \rho(T) + \rho(X \cap T))$.

Il est licite de s'interroger sur les propriétés d'une approximation en termes d'objets. Sans prétendre clore le débat, il semble logique d'exiger que la représentation d'une approximation soit conforme à la représentation première, en d'autres termes, il est exigé que $\rho(I) = \rho(\tau[\rho(I)])$. Dans ce cas, $\pi(I) = \pi[\pi(I)]$, autrement dit l'approximation est un *opérateur idempotent* ou *opérateur de projection*. En fait, cette propriété est rarement vérifiée.

La figure 1.4 montre différentes approximations de $\rho(I)$. Nous sommes parti des mêmes représentations $\rho(X)$ et $\rho(T)$ pour les quatre images, seule varie la qualité de restitution des textures, c'est-à-dire $\pi(T)$. La qualité de l'approximation est inversement proportionnelle au taux de compression. Il est intéressant de remarquer que, *contrairement à un codage JPEG, le signal codé ne se dégrade pas brutalement*, avantage que présentent les techniques de codage par objet. Mentionnons aussi qu'à fort taux de compression, notre technique de codage crée des 'zones d'ombre' dans les grandes régions. Ce défaut est lié à l'usage de la transformée de Fourier et à la quantification qui s'y rattache. La technique de codage des textures est détaillée dans le chapitre 3.

Transformations de représentation

Un procédé de traitement du signal de forme ou de texture comprend généralement plusieurs étapes. Ainsi, le codage de la forme tire sa matière première des résultats de la segmentation, puis il filtre les contours et, in fine, les passe à un codeur entropique après avoir organisé la source que représente ce signal – l'organisation de la source appelée *réduction* est détaillée au point suivant. C'est donc un enchevêtrement de techniques ayant chacune un rôle propre qui dresse l'ossature du procédé de codage. Dans un tel contexte, il n'existe pas de représentation unique pouvant servir de vocable pour l'ensemble des étapes. Tantôt c'est une description spatiale qui convient, tantôt le traitement s'axe sur le spectre local de l'image. Dès lors, même si elles traduisent une information précise, les



FIG. 1.4 - Étude de la sensibilité de notre méthode de codage orientée par la notion d'objet en fonction du taux de compression. Les taux valent respectivement: 15, 21, 31 et 39. Le coût de codage des contours a été calculé sur base de 1,3 bit par point de contour [22].

représentations s'adaptent au traitement dont il est question. Par exemple, lors du filtrage de forme, les contours sont vus comme délimitant les surfaces des régions ; au moment du codage, on peut les transformer en une chaîne reliant voisin à voisin.

Réduction de représentation et codage

La dernière étape d'une chaîne de codage est la phase de codage entropique. Telles quelles, les représentations $\rho(X)$ et $\rho(T)$ ne conviennent pas spécifiquement au codage entropique ; en cause, la redondance présente dans la représentation et le manque d'organisation. Pour le codage, nous *réduisons* les représentations afin d'obtenir des représentations épurées et ordonnées, indicées par le symbole r . Pour diminuer davantage le débit binaire nécessaire à la transmission, utilisation est faite de résultats de la théorie de l'information. Dans un schéma global de codage par objet, le codage capable de supprimer la redondance agit séparément sur les contours et sur les textures. Plus précisément, c'est par l'intermédiaire de très bonnes représentations réduites $\rho_r(X)$, $\rho_r(T)$ et éventuellement $\rho_r(X \cap T)$ que nous comptons concurrencer les techniques actuelles dans certaines applications de codage.

1.2.3 Mouvement d'objets

Jusqu'à présent, nous avons attiré l'attention sur la modélisation des opérations sur une image fixe. Hormis le cas particulier du changement de scène, la plupart des objets se retrouvent dans les images successives ; quelques-uns se sont déplacés, d'autres sont demeurés immobiles. L'exploitation de la correspondance entre *blocs* similaires d'image à image se nomme *compensation de mouvement*. Dans une approche axée sur la notion d'objet, ce n'est pas le bloc qui est considéré comme l'entité de comparaison mais l'objet. Passer à la notion d'objet complique le traitement parce que la forme n'est plus fixe mais dépendante du contenu de l'image, des objets. Pour utiliser l'information de mouvement d'objets dans un codage, plusieurs phases sont indispensables :

1. identifier les objets présents, ce qui revient à découvrir une représentation de l'image au temps t , $\rho^*(I, t) = \rho^*(X, t) + \rho^*(T, t) + \rho^*(X \cap T, t)$ adaptée à l'étude temporelle. Concrètement, il s'agira de la représentation $\rho(I)$ obtenue au terme de la détection de régions, légèrement modifiée ;
2. analyser l'image au temps t^+ (t^+ est le moment qui suit immédiatement t) par le même outil afin d'obtenir la représentation $\rho^*(I, t^+)$;
3. établir la correspondance entre $\rho^*(I, t)$ et $\rho^*(I, t^+)$;
4. coder le mouvement des objets retrouvés dans la scène au temps t^+ et les nouveaux objets.

Il va sans dire que l'avant-dernière étape s'avère particulièrement difficile. Dans l'approche classique où s'utilise le critère d'identification par sommation de valeurs absolues, on fait correspondre les objets des représentations de $\rho^*(I, t)$ et $\rho^*(I, t^+)$ qui possèdent des textures identiques. Mais les objets se ressemblent aussi par leur forme, qu'ils aient subi

un mouvement de translation, de rotation ou de déformation. C'est pourquoi, l'identification d'objet se réalise parfois par l'analyse de la forme. Ce procédé n'est pas toujours adéquat pour un codage. En effet, un objet qui se déplace découvre une zone de l'espace et, par conséquent, tous les objets ayant une frontière commune avec la zone découverte et réellement cachés au temps précédent subissent une déformation importante de leur forme. Autrement dit, par le simple mouvement d'un objet, on risque de croire que les objets du fond sont différents, du moins si l'identification d'objet porte uniquement sur la forme. Nous montrerons comment utiliser des notions d'extrapolation spatiale pour établir une correspondance de formes entre objets ayant subi un mouvement planaire quelconque ou un mouvement de déformation homothétique. C'est une des raisons supplémentaires qui fait de l'extrapolation un outil important : *identifier un objet par sa texture quelle que soit sa forme.*

Conclusions du chapitre

Nous avons présenté un cadre de travail mathématique et des notations cohérentes dans ce chapitre. Le concept central est celui de représentation d'image qui, dans une technique de traitement par objet, se découpe en deux composantes principales que sont la représentation de forme $\rho(X)$ et de texture $\rho(T)$. Elles sont obtenues au terme d'une phase de segmentation.

La notion de représentation est importante ; c'est elle qui servira de fil conducteur et de lien dans ce travail. Ainsi, les chapitres suivants décrivent des outils pour construire les représentations $\rho(X)$ et $\rho(T)$. Plus loin encore, nous expliquerons comment transformer la représentation de forme pour l'adapter à une technique de codage et réduire ainsi le débit binaire.

Première partie

Description d'outils adaptés au traitement d'objet et segmentation d'image

Introduction

La première tâche à accomplir dans un procédé de codage par objet est la détermination des représentations $\rho(X)$ et $\rho(T)$. Mais la complexité de cette opération est telle qu'elle requiert une intégration de plusieurs mécanismes, sans quoi les résultats ne pourraient convenir. Ces mécanismes agissent parfois sur la forme ou sur la texture des objets à détecter.

Cette partie du document décrit les outils de traitement d'objet qui interviennent principalement dans la phase de segmentation (cf. chapitre 4). La première famille d'outils, reprise sous les termes de la théorie de la morphologie mathématique au chapitre 2, fournit une batterie de moyens pour traiter la forme des objets. Il y a par exemple la notion d'*ouverture morphologique*, qui indique de quelle manière on peut recouvrir une forme par une autre, ou la *ligne de partage des eaux*, qui servira à parfaire la segmentation. Nous présenterons ensuite l'important procédé d'extrapolation de textures dont l'utilisation permet à la fois la détection des régions texturées dans des images, la mise en correspondance de textures similaires dans des images successives, ou encore, facilite le codage de la texture en supprimant l'information résiduelle de contour. Enfin, le chapitre 4 expliquera le fonctionnement de la segmentation. La technique de segmentation est, avec l'analyse des textures, l'apport original majeur de cette partie.

Chapitre 2

Morphologie mathématique

2.1 Introduction à la morphologie mathématique

L'analyse d'image a pour objectif la description aussi complète que possible des caractéristiques d'une image. Mais une image est constituée d'objets. Pour extraire des informations quantitatives de ces objets, plusieurs attitudes sont possibles : adopter une démarche relativement passive en déterminant *tous* les paramètres relatifs à cette image –cette démarche aboutit à un surplus informatif–, ou alors se contenter de quelques paramètres suffisants à caractériser l'image. L'approche morphologique s'inscrit dans ce second courant de pensée. De nature fondamentalement géométrique, elle consiste à *comparer les objets à analyser à un autre objet de forme connue*, appelé *élément structurant*. De la sorte, chaque élément structurant fait apparaître l'image sous un jour nouveau ; tout l'art revient à choisir le bon éclairage.

Historique

La description morphologique repose sur des notions purement ensemblistes. C'est en 1901 que MINKOWSKI énonce les premiers concepts d'une théorie, qui sous l'impulsion de MATHERON et SERRA, donnera naissance à la morphologie mathématique.

Trois aspects la caractérisent : l'aspect algébrique traitant les opérations ensemblistes et plus généralement les transformations, l'aspect probabiliste qui décrit des modèles aléatoires applicables aux ensembles, et celui de la géométrie intégrale, lié à l'étude des fonctionnelles d'images.

Actuellement la théorie morphologique jouit d'une place de choix à côté des théories que sont

- le *traitement de signal classique*, nous entendons par là celle qui s'occupe de filtrage linéaire, de convolution, des transformées, des ondelettes, ... ;
- les *méthodes statistiques* où l'accent est placé sur l'extraction de paramètres, et non sur la transformation d'une image en une autre image ;
- les *méthodes syntaxiques*, qui extraient des primitives significatives de l'image, primitives souvent géométriques, et leur associent des règles grammaticales.

La théorie de la morphologie est largement diffusée dans la communauté internationale de nos jours comme l'atteste la kyrielle de conférences, d'articles et d'applications industrielles qui l'utilisent. Par ailleurs, son champ d'application couvre quantité de domaines : imagerie médicale, sciences des matériaux, vision robotique, télédétection, analyse topographique, reconnaissance de caractères, ...

L'objectif de ce chapitre est de *présenter uniquement les éléments théoriques essentiels*, permettant de manipuler les concepts de la morphologie mathématique pour la segmentation, le filtrage et le codage des contours.

Plan du chapitre

Cette introduction poursuit l'entrée dans le monde de la morphologie mathématique par le biais d'un rappel de notions de la théorie des ensembles. La section suivante définit les opérations élémentaires de morphologie mathématique sur des ensembles et décrit quelques-unes de leurs propriétés. Nous étudierons ensuite les notions de géodésie et de reconstruction (section 2.3). Comme tous ces opérateurs sont croissants, les définitions et propriétés énoncées s'étendent naturellement à des fonctions. La dernière section aborde le procédé de segmentation par la ligne de partage des eaux.

2.1.1 Rappels sur les notions d'ensembles

Les ensembles seront notés par des majuscules $A, B, \dots$ et les éléments qu'ils comprennent par des minuscules $a, b, \dots$. Faisons un bref rappel sur des notions de la théorie des ensembles.

Égalité d'ensembles. Deux ensembles sont *égaux* s'ils sont formés des mêmes éléments : $X = Y \Leftrightarrow (p \in X \Rightarrow p \in Y \text{ et } p \in Y \Rightarrow p \in X)$. L'égalité entre ensembles est réflexive, symétrique et transitive. L'ensemble vide sera noté $\emptyset$.

Inclusion. X est un sous-ensemble de Y (X *inclus* dans Y) si tous les éléments de X appartiennent à Y : $X \subseteq Y \Leftrightarrow (p \in X \Rightarrow p \in Y)$. L'inclusion est réflexive, antisymétrique et transitive.

Intersection. L'*intersection* de deux ensembles X et Y est l'ensemble des éléments qui appartiennent aux deux : $X \cap Y = \{p \text{ tel que } p \in X \text{ et } p \in Y\}$. L'intersection est commutative, associative et *idempotente*, ce dernier terme traduit $X \cap X = X$.

Union. La *réunion* de deux ensembles est constituée des éléments appartenant à l'un ou à l'autre, c'est-à-dire $X \cup Y = \{p \text{ tel que } p \in X \text{ ou } p \in Y\}$. La réunion d'ensembles est commutative, associative et idempotente. Remarquons que contrairement à l'intersection, la réunion de deux ensembles dont l'un n'est pas vide, n'est jamais vide.

Différence. Étant donnés X et Y , la *différence* de X par Y , notée $X - Y$ ou $X \setminus Y$ est l'ensemble des éléments de X qui n'appartiennent pas à Y : $X - Y = \{p \mid p \in X \text{ et } p \notin Y\}$. La différence entre ensembles n'est en général pas commutative et ne possède pas de

bonnes propriétés. Certains utilisent parfois une autre loi de composition, appelée *différence symétrique*, notée $X \Delta Y$, et définie par $X \Delta Y = X \cup Y - X \cap Y$. C'est donc l'ensemble des éléments qui appartiennent à l'un ou à l'autre mais jamais aux deux ensembles. La différence symétrique est une opération commutative et associative.

Après le rappel de ces opérations d'ensembles usuelles, nous introduisons des opérations qui seront utiles pour la définition d'opérateurs morphologiques.

Complémentaire. Soit un sous-ensemble X d'un ensemble E servant de référentiel, le *complémentaire* de X dans E est le sous-ensemble noté X^c , fourni par $X^c = \{p \text{ tel que } p \in E \text{ et } p \notin X\}$. Si le référentiel E est modifié, le complémentaire diffère. Tout comme pour les opérateurs logiques, il existe des formules de Morgan où intervient le complémentaire : le complémentaire d'une intersection est égal à l'union des complémentaires $(X \cap Y)^c = X^c \cup Y^c$ et vice versa.

Addition de Minkowski. Soient X et Y deux ensembles quelconques de l'espace E . À tout $x \in X$ et $y \in Y$, il est possible de faire correspondre la somme algébrique $x + y$. En choisissant tous les x et y , on forme un nouvel ensemble appelé *addition de Minkowski* et noté $X \oplus Y$. Cet ensemble se définit par $X \oplus Y = \{x + y \mid x \in X, y \in Y\}$. Du fait de la commutativité de l'addition, $X \oplus Y = \bigcup_{x \in X} Y_x = \bigcup_{y \in Y} X_y$ où Y_x désigne le translaté de Y par x et vaut $\{p \in E \mid p = x + y, y \in Y\}$.

2.1.2 Propriétés de base des transformations morphologiques

Toute opération morphologique est par construction le résultat d'une ou plusieurs transformations sur des ensembles, en abrégé $\psi(X)$ suivi éventuellement d'une mesure $P(\psi(X))$. Assez grossièrement, une transformation morphologique est vue comme la réalisation d'opérations ensemblistes (union, intersection, complémentarité, ...) faisant intervenir X et l'ensemble ou élément structurant B déplacé à travers tout l'espace E , opérations dont le résultat s'interprète pour tout point de cet espace. Pour que l'outil de transformation ait une quelconque utilité, les opérations satisfont souvent à quatre principes (les opérateurs morphologiques introduits par la suite ne rencontrent pas tous ces principes) :

1. Invariance en translation.

La convolution et le filtrage linéaire sont des opérations invariantes en translation. Dans le même état d'esprit, il paraît logique d'imposer l'invariance en translation des transformations morphologiques, traduite mathématiquement par la relation $\psi(X_p) = (\psi(X))_p$. Il est clair qu'ainsi l'origine de l'espace ne joue plus le moindre rôle privilégié et que les seules transformations admises seront insensibles à la position des coordonnées.

2. Compatibilité avec les homothéties.

Les transformations dépendent parfois d'un facteur d'échelle positif λ . Soient λX un ensemble homothétique de X dans $\mathbb{R}^n$ et ψ_λ une transformation de paramètre λ . La transformation est compatible avec les homothéties s'il est équivalent d'appliquer ψ_λ à l'ensemble X dilaté du facteur λ ou de dilater le résultat de la transformée de X pour $\lambda = 1$. Cela se traduit par la relation : $\psi_\lambda(\lambda X) = \lambda \psi(X)$ où $\psi = \psi_1$.

À titre d'exemple, la transformation consistant à extraire d'un ensemble les cordes horizontales de longueur λ , est compatible avec les homothéties. Ce principe concerne toutes les transformations où opère un paramètre métrique.

3. Connaissance locale.

Le principe de connaissance locale intervient lorsque le champ de mesures ne contient qu'une partie de l'ensemble X : c'est le cas de structures observées au microscope. Grâce à cette propriété, l'extérieur de l'image peut être adapté de manière à faciliter l'application, sans que cela n'ait trop d'incidence sur le contenu informatif de l'image. Soit M l'ensemble borné fermé (de volume fini) définissant le champ de mesures. L'ensemble X n'est connu qu'à l'intérieur de ce champ, car on dispose de $X \cap M$. La transformation ψ vérifie le principe de connaissance locale si, quel que soit l'ensemble M borné, il existe N borné, ne dépendant pas de X , tel que $\psi(X \cap N) \cap M = \psi(X) \cap M$; ce qui signifie que le comportement de ψ à l'intérieur de M ne dépend que de ce qu'il y a dans N .

D'un point de vue pratique, la connaissance locale soulève la question du bord de l'image ; comment faut-il traiter un objet qui touche le bord du champ ? Nous reviendrons sur ce problème dans ce chapitre.

4. Continuité.

Considérons deux ensembles X et Y . Intuitivement, une transformation est dite *continue* si, lorsque Y tend vers X , la transformée de Y se rapproche de plus en plus de la transformée de X : $X \rightarrow Y \Rightarrow \psi(X) \rightarrow \psi(Y)$. Formellement la *continuité* est obtenue lorsque, pour tout voisinage V de $\psi(X)$, il existe un voisinage V' de X tel que pour $Y \subseteq V'$, $\psi(Y) \subseteq V$. Très clairement la continuité dépend de la notion de voisinage, c'est-à-dire de la topologie. SERRA [87] parle de semi-continuité, un concept 'plus faible' de continuité.

Dans la suite du document, nous verrons apparaître des correspondances avec les propriétés énoncées ci-dessus, parfois pour constater qu'elles ne sont pas toutes respectées par certains opérateurs morphologiques.

2.2 Transformations morphologiques élémentaires

Les développements repris ci-après concernent des ensembles et non des fonctions parce que les opérations serviront principalement à traiter la forme des objets. Les définitions seront étendues à des fonctions dans le chapitre consacré à l'étude d'algorithmes d'ouverture performants.

2.2.1 Érosion

La transformation par érosion est la première à avoir été proposée ; c'est d'ailleurs la plus importante avec la dilatation. Pour définir intuitivement l'opération d'érosion, situons-nous dans l'espace euclidien $\mathbb{R}^2$ partiellement occupé par un ensemble X ¹. Prenons un élément

1. Pour plus de clarté, les notations différencient les ensembles à traiter et les éléments structurants en les désignant respectivement par $X, Y, \dots$ et $A, B, C, \dots$

structurant B représentant une figure géométrique simple, par exemple un disque. Cet élément B_p est repéré par son centre et placé en p . Il est ensuite déplacé de telle sorte que son centre occupe successivement toutes les positions de l'espace. Pour chacune de ces positions, la question suivante est posée : l'ensemble B est-il entièrement inclus dans l'ensemble X ($B_p \subseteq X$) pour cette position de p ? L'ensemble des p fournissant une réponse positive forme un nouvel ensemble appelé *érosion de X par B* et noté $X \ominus B$. Par ailleurs,

Définition 3

$$X \ominus B = \bigcap_{b \in B} X_{-b} = \{p \in \mathbb{R}^2 \mid B_p \subseteq X\} \quad (2.1)$$

ce qui signifie que la transformation par érosion d'un ensemble X par B s'obtient en translatant X par l'opposé de chaque élément de B et en ne conservant que les points appartenant à toutes les translations de X . Avec cette interprétation, il est clair que si la taille de B dépasse celle de X , l'ensemble X érodé par B est vide. Parfois l'opération d'érosion est définie avec B ayant préalablement subi une symétrie centrale autour de l'origine (cf. SERRA [87]), dans le but d'accroître le formalisme de dualité par complémentation inhérent à la plupart des transformées morphologiques.

L'opération d'érosion est illustrée par la figure 2.1. On voit que la petite composante de X a presque disparu. Les effets habituels de l'érosion sont : la séparation des objets à l'endroit des étranglements, le rétrécissement des objets de taille supérieure à B et la disparition des petites composantes. Dans le chapitre consacré au filtrage des contours, nous parlerons d'une opération appelée *érosion partielle*, qui assouplit la condition d'inclusion reprise dans la définition 3.

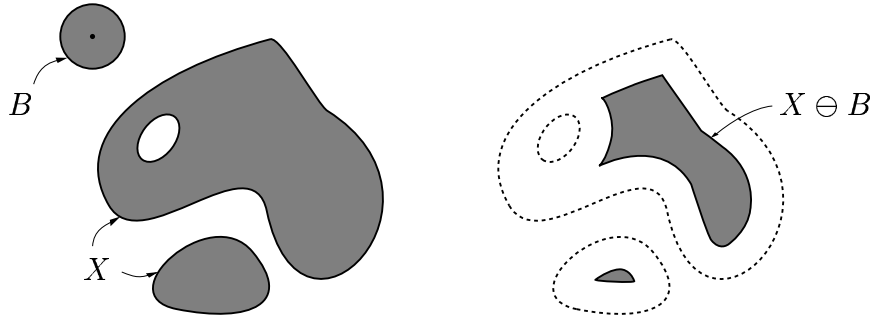


FIG. 2.1 - Érosion de X par un disque B . L'origine de l'élément structurant est représentée par un point noir.

2.2.2 Dilatation

L'opération de *dilatation*, appelée *dilation* en anglais, due à Minkowski, se définit de manière analogue. On obtient le dilaté de X par B en prenant la totalité de la surface recouverte lorsque l'origine de B occupe successivement toutes les positions $x \in X$.

Définition 4 En termes ensemblistes, la dilatation de X par B est l'union des translatés de X par des éléments de B :

$$X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = \{x + b \mid x \in X, b \in B\} \quad (2.2)$$

La figure 2.2 montre la dilatation par un disque du même ensemble X que celui traité lors de l'érosion. Dans cet exemple, les deux composantes connexes² sont réunies dans le dilaté; tout comme l'érosion, la dilatation ne préserve pas la connexité.

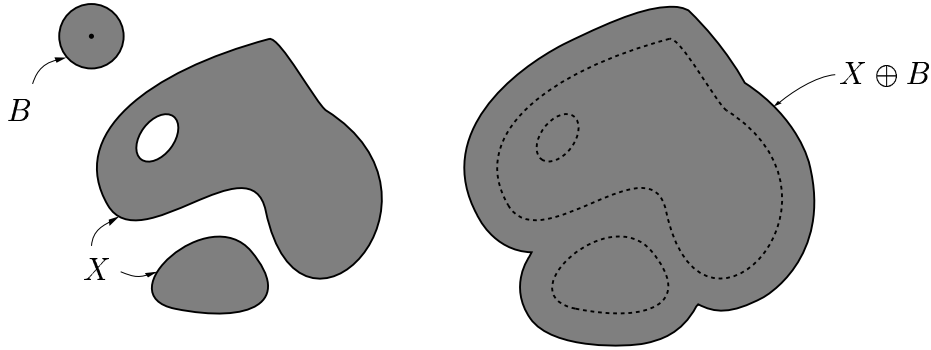


FIG. 2.2 - Dilatation de X par un disque B .

Dualité. La transformation de dilatation n'est pas indépendante de l'opération d'érosion. Éroder X revient à dilater son complémentaire X^c . Les opérations de dilatation et d'érosion sont donc deux opérations duales par rapport à la complémentation :

$$X \ominus \check{B} = (X^c \oplus B)^c \quad (2.3)$$

$$X \ominus B = (X^c \oplus \check{B})^c \quad (2.4)$$

où $\check{B}$ est le symétrique de B , défini par $\check{B} = \{-b \mid b \in B\}$.

Avec nos définitions d'érosion et de dilatation, la présence du symétrique de B est inévitable. C'est pour l'éviter que certains adoptent parfois une autre définition de l'érosion.

L'exploitation des propriétés de dualité permet de transposer le travail fait pour une érosion à celui d'une dilatation et vice versa. De plus, la pratique favorise l'emploi d'éléments symétriques, d'où découle une forme simplifiée du principe de dualité par complémentation.

2.2.3 Propriétés des opérations d'érosion et de dilatation

Les opérations d'érosion et de dilatation vérifient les principes des transformations morphologiques idéales :

1. l'érosion et la dilatation sont invariantes en translation : de fait, il revient au même de traduire un ensemble X puis de l'éroder, que de l'éroder puis de traduire l'image transformée : $X_p \ominus B = (X \ominus B)_p$. De même, $X_p \oplus B = (X \oplus B)_p$;
2. l'érosion et la dilatation sont compatibles avec les homothéties : $\lambda X \ominus \lambda B = \lambda(X \ominus B)$ et $\lambda X \oplus \lambda B = \lambda(X \oplus B)$;
3. l'érosion et la dilatation vérifient la condition de connaissance locale pour B borné ;

2. Un ensemble X est *connexe* si $\forall x_1, x_2 \in X$, il existe toujours un chemin reliant x_1 à x_2 totalement inclus dans X .

4. nous admettrons sans démonstration que l'érosion et la dilatation sont des transformations semi-continues. Pour une démonstration rigoureuse, le lecteur consultera par exemple l'ouvrage de SERRA [87]. Dans la pratique, la morphologie mathématique s'utilise dans $\mathbb{Z}^2$; la discrétisation escamote considérablement les problèmes de continuité.

Les deux opérations morphologiques jouissent aussi de propriétés algébriques :

- l'érosion et la dilatation sont des opérations croissantes : si $X \subseteq Y$, alors $(X \ominus B) \subseteq (Y \ominus B)$ et $(X \oplus B) \subseteq (Y \oplus B)$;
- si l'élément structurant contient l'origine, l'érosion est une opération anti-extensive alors que la dilatation est extensive, c'est-à-dire $X \ominus B \subseteq X$ et $X \subseteq X \oplus B$.

2.2.4 Ouverture

Puisque l'érosion et la dilatation sont des opérations morphologiques, il vient à l'esprit de combiner ces opérations élémentaires.

Définition 5 *L'opération obtenue par la succession d'une érosion et d'une dilatation est l'ouverture³. Elle se note $X \circ B$ et vaut $(X \ominus B) \oplus B$.*

En général, l'ensemble traité diffère de l'ensemble de départ : l'ensemble ouvert est plus régulier et moins riche en détails que l'ensemble initial. La transformation par ouverture adoucit donc les contours. L'ouverture joue le rôle d'un filtrage que nous retrouverons d'une façon formelle plus loin.

2.2.5 Fermeture

Définition 6 *En inversant l'ordre des opérations utilisées pour définir l'ouverture, nous obtenons une nouvelle opération appelée fermeture ; c'est la transformation $X \bullet B = (X \oplus B) \ominus B$.*

Il s'agit bien de l'opération duale de l'ouverture pour la complémentation, car $(X \circ B)^c = X^c \bullet \check{B}$ et $(X \bullet B)^c = X^c \circ \check{B}$. La fermeture d'un ensemble effectue donc une opération de filtrage similaire à celle effectuée par l'ouverture mais sur le complémentaire de l'ensemble. L'ouverture et la fermeture sont illustrées à la figure 2.3.

Interprétation de l'ouverture et de la fermeture. L'interprétation la plus commode de l'opération d'ouverture est fournie par la propriété que voici :

$$X \circ B = \bigcup_{p \in \mathbb{R}^2} B_p \quad \text{tel que} \quad B_p \subseteq X \quad (2.5)$$

Ainsi, l'ouverture d'une figure par un élément structurant B est l'ensemble des points recouverts lors du déplacement de B à l'intérieur de la figure ; cette interprétation explique

3. La morphologie mathématique propose des ouvertures ne résultant pas de la mise en série d'une érosion et d'une dilatation. Dans ce texte, il ne sera pourtant question que de ce type d'ouverture.

la présence de disques noirs dans le dessin inférieur gauche de la figure 2.3. Une propriété similaire vaut par dualité pour la fermeture mais, à cette fois, l'élément structurant parcourt le complémentaire de la figure (cf. dessin inférieur droit de la figure 2.3).

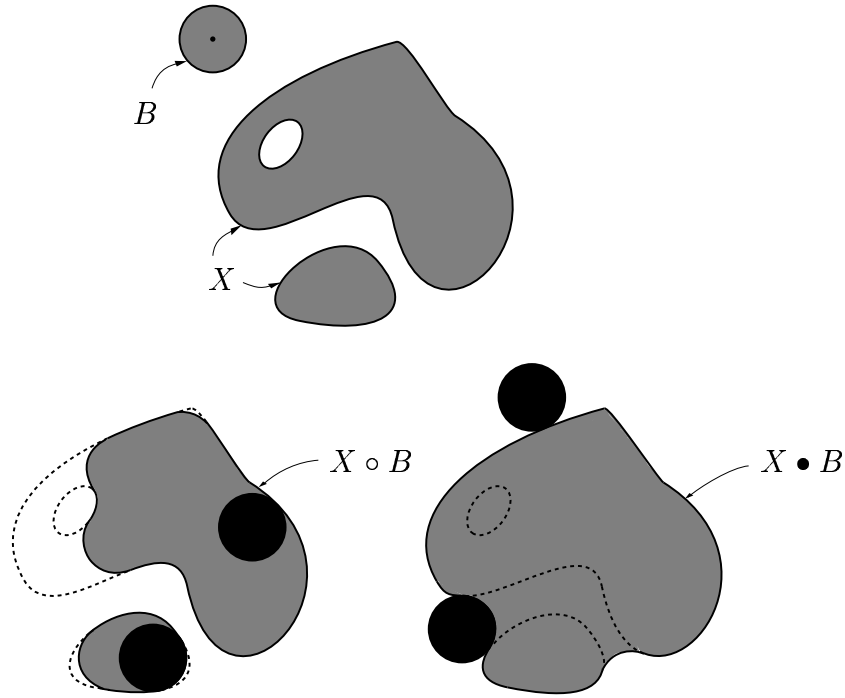


FIG. 2.3 - Ouverture et fermeture de X par un disque B .

2.2.6 Propriétés de l'ouverture et de la fermeture

Par construction, l'ouverture et la fermeture respectent les quatre principes de toute transformation morphologique. Concernant les propriétés algébriques, citons les propriétés suivantes :

- l'ouverture et la fermeture sont des transformations croissantes :
Si $X \subseteq Y$, alors $(X \circ B) \subseteq (Y \circ B)$ et $(X \bullet B) \subseteq (Y \bullet B)$;
- l'ouverture est anti-extensive alors que la fermeture est extensive :
 $X \circ B \subseteq X$, $X \subseteq X \bullet B$;
- l'ouverture et la fermeture sont des opérateurs idempotents (ou opérateurs de projection), ce qui revient à dire que $(X \circ B) \circ B = X \circ B$ et $(X \bullet B) \bullet B = X \bullet B$.

2.2.7 Propriétés générales

Pour décrire des réalisations matérielles ou par logiciel, il est utile de mettre en lumière quelques propriétés supplémentaires (les démonstrations manquantes sont disponibles dans l'article d'HARALICK [33]). Nous ferons aussi usage de ces propriétés pour certaines

démonstrations relatives à l'étude des propriétés des opérations partielles et pour justifier la description d'algorithmes rapides.

Propriété 7 *La dilatation est commutative et associative*

$$X \oplus B = B \oplus X \quad (2.6)$$

$$(X \oplus B) \oplus C = X \oplus (B \oplus C) \quad (2.7)$$

Démonstration

Relation d'associativité:

$p \in X \oplus (B \oplus C)$ si et seulement si (ssi) il existe $x \in X, b \in B$ et $c \in C$ tel que $p = x + (b + c)$. Mais l'addition dans $\mathbb{R}^2$ est une opération associative $p = (x + b) + c$, ce qui suffit à démontrer la propriété. ■

Quelques remarques sur l'implémentation des opérations morphologiques élémentaires. Dans la pratique, X et B sont des objets très différents puisque B est l'*élément structurant*. La dilatation par un carré de taille 3x3 est une opération aisée à réaliser dans une architecture actuelle. La propriété d'associativité est fondamentale pour ce qui est du nombre d'opérations à réaliser lorsque X est analysé par $(B \oplus C)$. Soit N une dimension caractéristique de B et C (largeur ou hauteur). Une application brutale de la dilatation de X par $B \oplus C$ nécessite N^2 opérations alors que la succession de la dilatation $X \oplus B$ suivie de la dilatation par C requiert $2N$ opérations. Cependant l'intérêt reste restreint pour des architectures qui effectuent la dilatation par accès aléatoire à tous les points du voisinage. Des efforts de recherche constants sont entrepris pour réduire le nombre d'opérations nécessaires. Ainsi, VINCENT [111] a mis au point un algorithme souple et performant pour déterminer les dilatations et érosions d'images binaires avec des éléments structurants quelconques. La première étape consiste en un suivi de contours de l'image et en leur codage sous forme de boucles. Dans un second temps, l'élément structurant représenté par une structure de données appropriée, est efficacement propagé le long de ces boucles. Pour traiter des images en niveaux de gris, nous avons nous-même proposé un algorithme de complexité N et un autre algorithme indépendant de la taille de l'élément structurant sur base de la manipulation d'un histogramme local [104]. Ces algorithmes, décrits dans ce document, permettent aussi d'effectuer une ouverture en une seule passe. ZHUANG [113] propose une méthode utilisable pour des éléments structurants convexes du type $B = (B_1 \oplus B_2)$; on les appelle des "Steiner". En fait, la plupart des propriétés citées ci-après induisent un algorithme particulier permettant une certaine optimisation des opérations morphologiques. Tantôt l'élément structurant est décomposé par dilatation, tantôt par intersection ou encore par union.

Propriété 8 *La dilatation distribue l'union:*

$$\left(\bigcup_j X^j \right) \oplus C = \bigcup_j (X^j \oplus C) \quad (2.8)$$

Démonstration

$$(\bigcup_j X^j) \oplus C = \bigcup_{p \in \bigcup_j X^j} C_p = (\bigcup_{p \in X^1} C_p) \cup \dots \cup (\bigcup_{p \in X^k} C_p) = (X^1 \oplus C) \cup \dots \cup (X^k \oplus C) = \bigcup_j (X^j \oplus C). \quad \blacksquare$$

Par commutativité, $X \oplus (B \cup C) = (X \oplus B) \cup (X \oplus C)$. Cette égalité est importante ; elle permet la décomposition d'un élément structurant en une union d'éléments structurants, offrant ainsi une alternative dans l'évaluation du coût d'une réalisation.

Propriété 9 *L'érosion est distributive par rapport à l'intersection :*

$$\left(\bigcap_j X^j\right) \ominus C = \bigcap_j (X^j \ominus C) \quad (2.9)$$

Démonstration

$$(\bigcap_j X^j) \ominus C = \bigcap_{c \in C} (\bigcap_j X^j)_{-c} = \bigcap_{c \in C} (\bigcap_j X_{-c}^j) = \bigcap_j (\bigcap_{c \in C} X_{-c}^j) = \bigcap_j (X^j \ominus C). \quad \blacksquare$$

La relation suivante est régulièrement utilisée pour réduire le temps de calcul d'une érosion par un grand élément structurant. C'est la *relation de chaîne* ou de *mise en cascade*.

Propriété 10

$$X \ominus (B \oplus C) = (X \ominus B) \ominus C \quad (2.10)$$

Démonstration

En vertu de l'invariance en translation de l'érosion :

$$X \ominus (B \oplus C) = \bigcap_{b \in B, c \in C} X_{-(b+c)} = \bigcap_{c \in C} (\bigcap_{b \in B} X_{-b})_{-c} = (X \ominus B) \ominus C. \quad \blacksquare$$

Cette relation a son importance car elle permet d'effectuer des érosions d'éléments structurants de grande taille et décomposables en une dilatation de deux plus petits éléments –par exemple, un carré est équivalent à une dilatation d'un segment horizontal par un segment vertical– par deux érosions successives de taille réduite.

Propriété 11

$$X \oplus B \subseteq Y \Leftrightarrow X \subseteq Y \ominus B \quad (2.11)$$

Démonstration

$$X \oplus B = \bigcup_{b \in B} X_b \subseteq Y \Leftrightarrow \forall b \in B : X_b \subseteq Y \Leftrightarrow \forall b \in B : X \subseteq Y_{-b} \Leftrightarrow X \subseteq \bigcap_{b \in B} Y_{-b} = Y \ominus B. \quad \blacksquare$$

La propriété précitée est appelée *adjonction* [34].

Propriété 12

$$X \ominus C = (X \ominus C) \bullet C = (X \circ C) \ominus C \quad (2.12)$$

$$X \oplus C = (X \oplus C) \circ C = (X \bullet C) \oplus C \quad (2.13)$$

Propriété 13 *L'ouverture et la fermeture ne dépendent pas de l'origine de l'élément structurant, comme le suggérait l'interprétation de ces opérations. Soit $p \in \mathbb{R}^2$,*

$$X \circ B_p = X \circ B \quad (2.14)$$

$$X \bullet B_p = X \bullet B \quad (2.15)$$

Un problème pratique : la gestion du bord dans les réalisations

Si l'invariance en translation découle des définitions des opérations morphologiques, la pratique contredit sa véracité parce que le bord de champ limite la connaissance de l'objet à un certain domaine. Une translation de l'image entraîne une perte irréversible d'information. Deux attitudes, correspondant à des hypothèses physiques sur l'extérieur du domaine d'observation et illustrées à la figure 2.4, sont possibles : soit l'objet s'étend au-delà du masque (hypothèse 1), soit on suppose que, physiquement, il n'y a rien en dehors du masque (hypothèse 2).

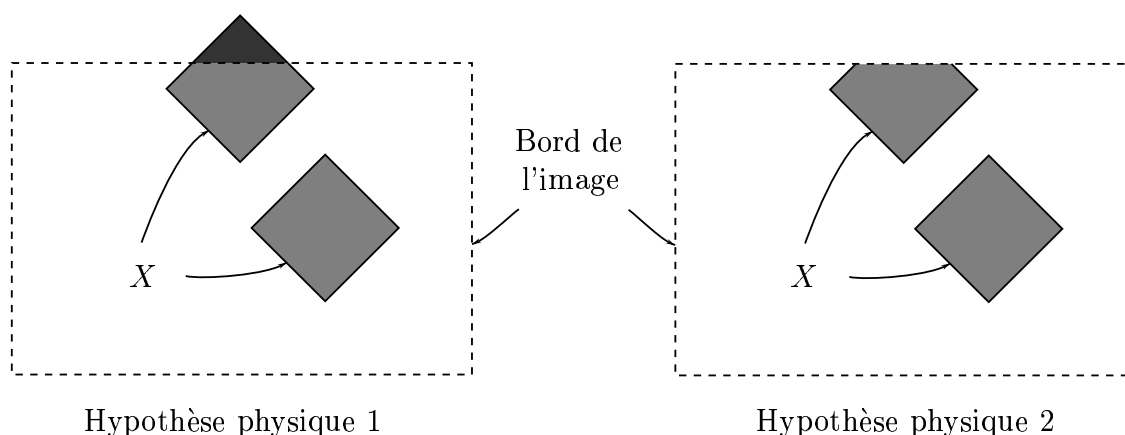


FIG. 2.4 - Deux hypothèses physiques pour traiter les objets qui touchent le bord de l'image. La première hypothèse, illustrée sur l'image de gauche, considère que le masque de l'image coupe en partie l'objet. Dans la seconde hypothèse, on admet que X ne s'étend pas au-delà du masque.

Dans la seconde situation, les traitements morphologiques ne posent pas de problème car on dispose de toute l'information nécessaire. Il en va autrement pour la première hypothèse. Soit à effectuer, par exemple, une érosion. On a beau admettre que l'objet s'étende en dehors de l'image, cela ne signifie pas pour autant que l'on puisse délimiter cette zone d'extension. Le pis-aller habituel consiste à ajouter à X tous les points situés à l'extérieur de l'image lors du calcul de l'érodé. Cela modifie le résultat de l'érosion, comme le montrent les érosions par un disque centré, représentées à la figure 2.5. Par dualité, le calcul de la dilatation s'effectue en supposant que les points extérieurs n'appartiennent pas à l'ensemble.

Contrairement aux apparences, ceci n'est pas suffisant pour traiter correctement une ouverture dans le cas de l'hypothèse 1. Prenons l'exemple de l'ouverture de X par le disque B décentré, dessiné à la figure 2.6. Malgré la présence d'un bord lors du calcul de l'érosion, le rectangle du haut ne crée aucune contribution dans $X \ominus B$, et donc dans l'ouvert, alors qu'une même surface située à droite subsistera en partie dans l'ouvert. De plus, déplacer l'origine de B vers l'extrémité droite du disque inverse le résultat. Il n'est pourtant pas logique que le résultat dépende à ce point de la position de l'origine, surtout après avoir énoncé la propriété d'indépendance de cette position dans le résultat de l'ouverture.

Soit M le masque d'observation, celui fixé par le bord de l'image. Comme remède, nous proposons une 'définition pratique' de l'ouverture, à savoir

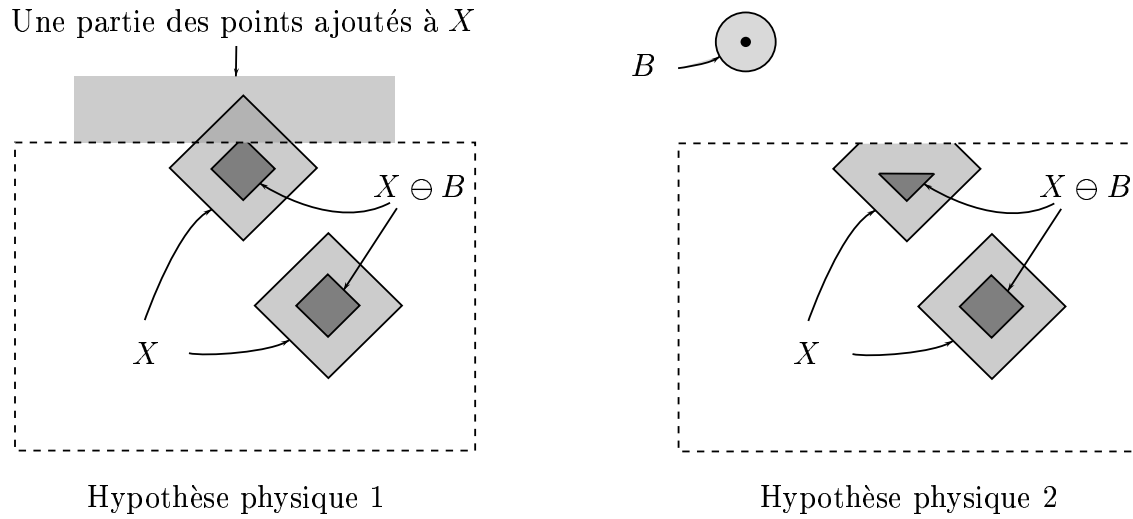


FIG. 2.5 - Comparaison de l'érosion de X , composé de deux losanges dans le plan $\mathbb{R}^2$, par un disque centré B pour les deux hypothèses physiques.

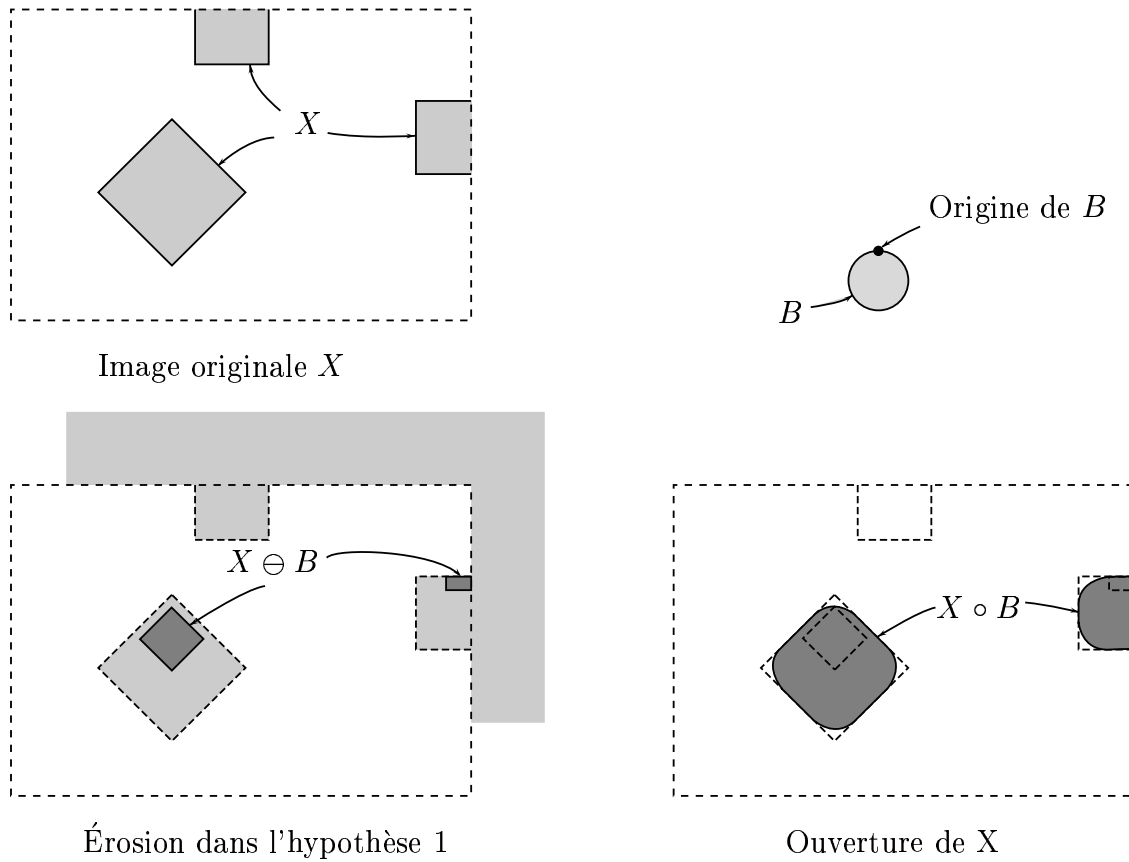


FIG. 2.6 - Pour illustrer que l'ouverture n'est pas unique malgré l'ajout d'un bord lors de l'érosion.

$$X \circ B = \bigcup_{p \in \mathbb{Z}^2} [(X \cup M^c) \ominus B_p] \oplus B_p \cap M \quad (2.16)$$

Pour la plupart des éléments structurants, quelques positions p suffiront à obtenir la totalité de l'ouvert. Dans le pire des cas, il faudra les prendre en nombre égal au périmètre de B .

2.3 Géodésie et reconstruction

Pour reconstituer une particule connexe X à partir d'un de ses points x , il faut déterminer les points connexes de proche en proche. Ce procédé appelé *reconstruction* appartient à la famille des opérateurs géodésiques. Comme la technique de segmentation proposée au chapitre 4 recourt explicitement à la reconstruction, nous développons son expression. Mais avant, il convient encore de préciser quelle est la relation de voisinage utilisée. Cette notion de voisinage joue aussi un rôle dans l'expression de $\rho(X)$ (cf. chapitre 7), où elle permet de construire le graphe des régions voisines.

Relations de voisinage. Sur une grille d'échantillonnage ou *trame*, carrée ou hexagonale, il y a plusieurs chemins possibles pour joindre un point à un autre ; le plus court chemin étant appelé *chemin géodésique*. Même ce dernier n'est pas unique et dépend en outre de la relation de voisinage. Nous introduisons pour cela le concept de *4-voisinage* ; les autres voisinages s'obtiennent similairement.

Définition 14 Dans un réseau rectangulaire discrétisé, le *4-voisinage* du point (i, j) est l'ensemble $V_4(i, j)$ des points voisins dans les quatre directions cardinales. Voici la définition de l'ensemble en extension :

$$V_4(i, j) = \{(i - 1, j), (i, j - 1), (i + 1, j), (i, j + 1)\} \quad (2.17)$$

Le voisinage intervient également pour définir la frontière d'un ensemble discret.

Définition 15 Soit X une région du plan $\mathbb{Z}^2$. La frontière ∂X est l'ensemble :

$$\partial X = \{(i, j) \in \mathbb{Z}^2 \mid (i, j) \in X \text{ et } V_4(i, j) \cap X^c \neq \emptyset\} \quad (2.18)$$

C'est la définition d'une 4-frontière. On pourrait tout aussi bien définir une notion de voisinage en 8-adjacence et déduire l'expression de frontière analogue à la définition 15. En trame carrée se pose toujours la question de savoir quelle relation d'adjacence sert de support théorique. Par contre, il n'existe qu'une connexité en trame hexagonale (6-connexité ou 6-adjacence). C'est la principale raison qui pousse la plupart des morphologues à travailler en trame hexagonale quand bien même la caméra d'acquisition ou l'écran produisent des images en trame carrée.

2.3.1 Dilatation géodésique

Une dilatation géodésique fait toujours intervenir deux images. La première est le dilaté par un élément structurant élémentaire isotropique adapté à la trame ; il s'agit d'un carré de trois pixels de large pour une trame carrée en 8-adjacence et d'un hexagone dans une trame hexagonale. Quant à la seconde image, elle limite l'extension de la dilatation de la première image.

Définition 16 La dilatation géodésique de taille 1 de l'ensemble X conditionnellement à Y , notée $D_Y^{(1)}(X)$, est définie comme l'intersection du dilaté de taille 1 et de Y :

$$\forall X \subseteq Y, \quad D_Y^{(1)}(X) = (X \oplus B) \cap Y \quad (2.19)$$

où B est le voisinage de l'origine pour l'adjacence choisie.

Suite à l'apparition de l'intersection dans la définition, $D_Y^{(1)}(X)$ est toujours inclus ou égal à Y . On dit aussi que Y sert de *masque géodésique*. La figure 2.7 illustre le principe de la dilatation géodésique de taille 1 dans une trame carrée.

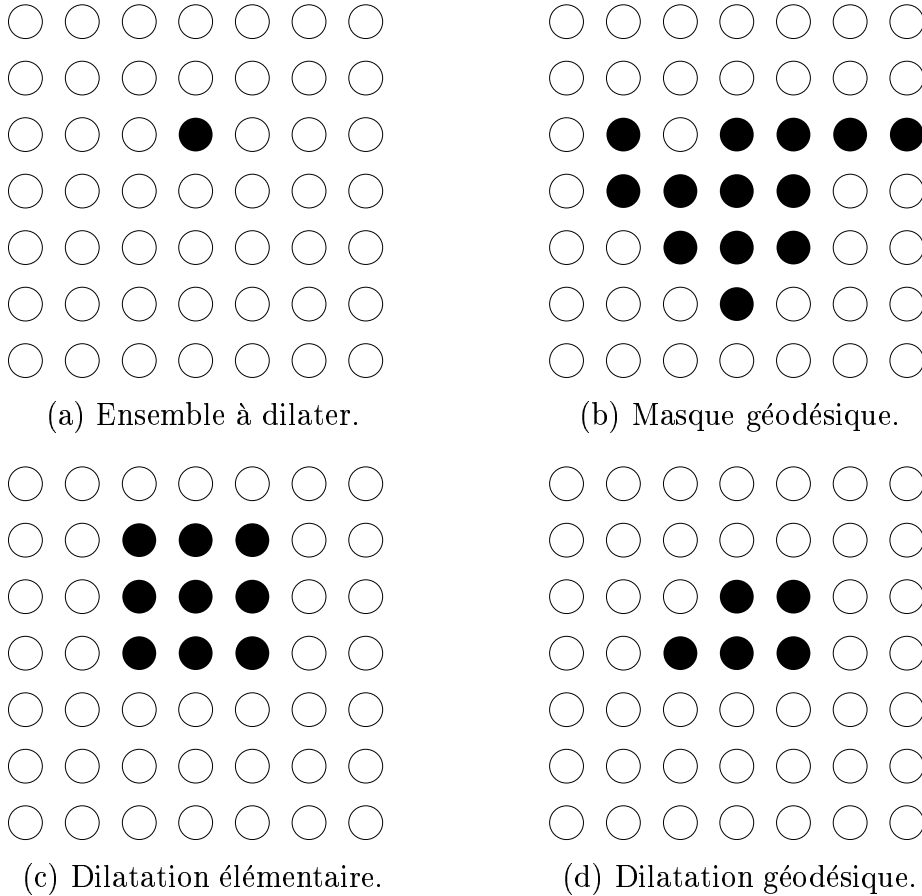


FIG. 2.7 - Dilatation géodésique de taille 1 d'un ensemble.

Définition 17 La dilatation géodésique de taille n de l'ensemble X conditionnellement à Y , notée $D_Y^{(n)}(X)$, est définie comme une succession du dilaté géodésique de taille 1 :

$$\forall X \subseteq Y, \quad D_Y^{(n)}(X) = \underbrace{D_Y^{(1)}(D_Y^{(1)}(\dots D_Y^{(1)}(X)))}_{n \text{ fois}} \quad (2.20)$$

où B est l'élément le plus simple adapté à la trame.

Il est important de remarquer que $D_Y^{(n)}(X)$ n'est pas toujours égal à l'ensemble obtenu en intersectant X dilaté par l'élément structurant B de taille n avec Y . Prenons le cas d'un ensemble Y formé de plusieurs composantes connexes et de X réduit à une composante de Y . Suite à la définition de la dilatation géodésique, $D_Y^{(n)}(X) = X$ alors que le résultat de l'intersection après plusieurs dilatations pourra contenir des parties des autres composantes connexes de Y .

2.3.2 Reconstruction

L'opération de dilatation géodésique est rarement utilisée en tant que telle. C'est plutôt l'ensemble vers lequel converge cette opération lorsque n croît qui permet de résoudre certains problèmes pratiques. En fait, la dilatation géodésique converge après un certain nombre d'itérations. Le procédé de reconstruction tire profit de ce principe.

Définition 18 *La reconstruction de X conditionnellement à Y est la dilatation géodésique de X jusqu'à idempotence. Soit i , la valeur à partir de laquelle l'idempotence est acquise, la reconstruction de X est définie par*

$$R_Y(X) = D_Y^{(i)}(X) \quad \text{avec} \quad D_Y^{(i+1)}(X) = D_Y^{(i)}(X) \quad (2.21)$$

Comme l'indique son nom, cette opération permet de reconstruire des détails perdus éventuellement lors d'une autre opération. La figure 2.8 représente les trois étapes mises en œuvre pour extraire des particules choisies dans une image. L'image (a) est le signal de départ. Dans l'image (b), on sélectionne certaines particules en insérant quelques points appelés *marqueurs* dans celles-ci. La reconstruction conduit alors à l'image (c) où ne sont reconstituées que les particules sélectionnées; les autres ont été gommées.

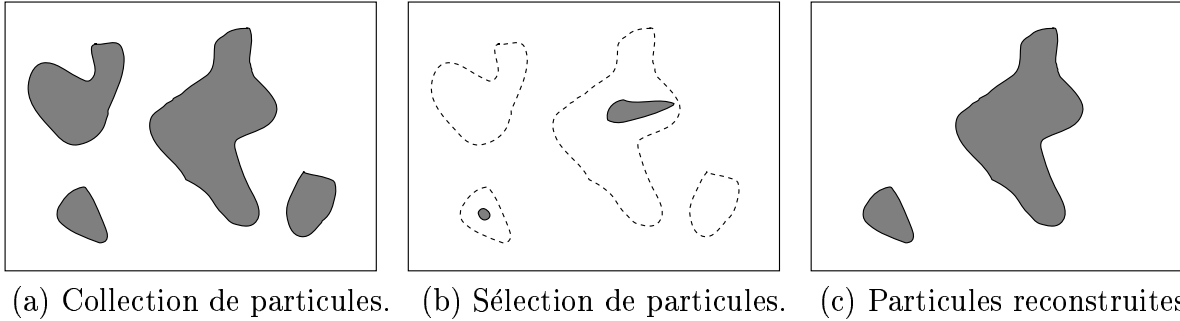


FIG. 2.8 - *Extraction de particules pré-sélectionnées dans une image.*

L'itération de la dilatation dans la phase de reconstruction pousse à penser que le temps de calcul de la reconstruction est élevé. En fait, il existe des algorithmes rapides et simples pour réaliser la reconstruction. Ils ont même été implantés dans un processeur spécialisé de traitement d'images par morphologie mathématique.

Avant de conclure ce chapitre, nous détaillons le procédé de segmentation par ligne de partage des eaux, parce qu'il interviendra dans notre technique de segmentation.

2.4 Segmentation d'image par ligne de partage des eaux

Dans le domaine de l'analyse d'image et plus particulièrement en morphologie mathématique, on considère que les fonctions sont des surfaces où la valeur de luminance est synonyme d'élévation. Cette représentation est pratique pour l'étude de techniques de segmentation utilisant des outils morphologiques. Avec quelques variantes, les techniques de segmentation morphologique se façonnent à partir de deux outils que sont la *transformation de ligne de partage des eaux* et le *marquage*.

À l'origine, l'algorithme de la ligne de partage des eaux a été proposé par LANTUÉ-JOUL [46] dans le cadre des fonctions en escalier. BEUCHER [1] a ensuite eu l'idée d'étendre l'outil à la segmentation d'images à teintes de gris. SOILLE et VINCENT [94] ont le mérite d'avoir mis au point des algorithmes de calcul performants.

2.4.1 Que penser de la ligne de partage des eaux ?

Si dans une interprétation topographique, la valeur de luminance représente l'élévation, cela ne signifie pas que le profil puisse être découpé correctement par un simple effet de seuillage. Un seuil élevé détectera bien les pics et un seuil faible extraira les vallées mais ces fonctions ne sont pas compatibles.

Puisque les valeurs de luminance ne semblent pas convenir directement, on est tenté de recourir à des gradients morphologiques⁴. L'inconvénient majeur de ceci est que le gradient est lui-même une fonction et que, comble de malheur, il faut à nouveau essayer des seuils pour segmenter. Bien sûr, a priori, l'information du gradient est plus pertinente que la simple valeur de luminance mais le problème reste entier. On peut encore remplacer le gradient par un outil qui analyse les transitions.

Trouver la procédure optimale pour placer des contours sur la fonction gradient n'est pas simple du tout. Par contre, identifier les régions où il n'y a pas de contours est plus simple : il s'agit de régions dans lesquelles le gradient est minimum. La surface de la fonction gradient décrit en quelque sorte des bassins séparés par des crêtes. Il paraît assez logique d'identifier ces crêtes à des contours idéaux. En termes imagés, un bassin $\mathcal{C}(M)$ est associé à chaque minimum M . Si une goutte d'eau tombe en un point de la surface, elle descendra jusqu'à se retrouver dans le fond d'un bassin. Ainsi chaque bassin M a sa propre zone d'attraction constituée des points à partir desquels le minimum sera atteint. Deux bassins voisins possèdent une frontière commune, frontière à partir de laquelle une goutte d'eau pourra glisser au choix dans l'un ou l'autre des bassins. L'ensemble de ces frontières forme la *ligne de partage des eaux (LPE)*. On peut comprendre ces notions sur la figure 2.9.

Une seconde interprétation est envisageable : l'approche par immersion. L'eau vient par le bas, des sources jaillissent dans le fond de la surface. Les bassins d'inondation se rejoignent et forment des frontières qui, elles aussi, sont des contours. Avant de poursuivre, il nous faut parler encore d'un remède à la sur-segmentation. Une application pure et simple

4. En principe, le gradient est un vecteur. Cependant, les gradients morphologiques, comme le *gradient morphologique interne* obtenu par différence entre une image et cette image érodée, se caractérisent uniquement par une amplitude. C'est la raison pour laquelle, dans cette section, nous parlons de *gradient* et non d'*amplitude du gradient*, terme plus précis dans un contexte général.

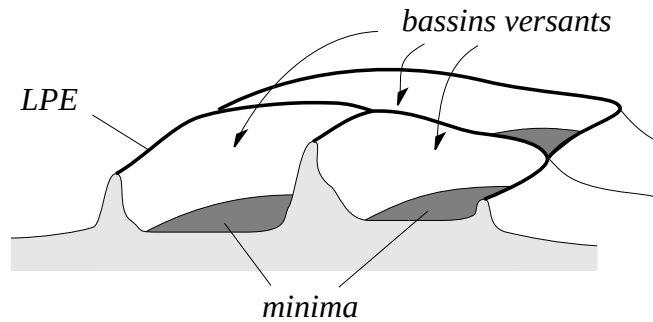


FIG. 2.9 - Minima, bassins versants et ligne de partage des eaux (extrait de [94]).

du schéma de détection de contours décrit conduit inéluctablement à une sur-segmentation parce que l'opérateur gradient est très sensible au bruit, ce qui a comme conséquence désastreuse d'introduire des minima artificiels et donc des bassins indésirables. Pour éviter ce désagrément, il nous faut une information supplémentaire : les centres des véritables vallées et sommets. L'introduction d'une distinction entre vrais et faux extrema modifie légèrement l'algorithme de segmentation ; c'est l'étape de *marquage*.

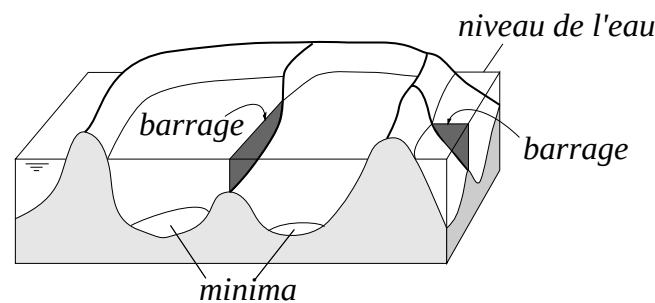


FIG. 2.10 - Construction d'un barrage au raccord de deux bassins versants (extrait de [94]).

Lors d'une montée des eaux, on construit un barrage entre deux vrais bassins. Ce mécanisme est illustré à la figure 2.10. Par contre, le faux bassin sera immergé par le flux d'eau en provenance de son voisin sans qu'un quelconque barrage ne soit construit. Après avoir inondé complètement le profil, seuls les barrages feront office de vrais contours. N'avons-nous pas simplement déformé le problème de la sur-segmentation ? La valeur du gradient n'est plus suffisante, il faut aussi des *marqueurs* différenciant les vraies et les fausses sources. En fait, le problème est souvent plus simple parce que l'expérimentateur dispose d'une information a priori. On peut faire les remarques suivantes :

- dans un mode interactif d'interprétation d'image où l'intéressé a la possibilité de donner des indications, la détermination des véritables extrema se reporte sur ce que fournit l'opérateur. Il n'est évidemment plus question d'interprétation automatique d'image ;
- si l'image à traiter se compose d'objets de luminance fort distincte, un algorithme un tant soit peu intelligent pourra focaliser la recherche des extrema sur les centres des véritables régions ;

- en recourant à une décomposition pyramidale, il est possible de franchir un premier pas dans la direction d'une automatisation de l'élimination des faux contours. Ainsi, une étape de la pyramide fournit une version filtrée et sous-échantillonnée de l'image de départ. L'image interpolée reconstruite ne contiendra plus les objets dont la taille est inférieure à celle de l'élément structurant de filtrage. Plus généralement, il peut être souhaitable de supprimer par filtrage des régions non désirées, par exemple des régions étroites orientées dans une certaine direction;
- les ouvertures et les fermetures offrent un moyen commode de détection des pics et des vallées et elles effectuent un filtrage intéressant. L'opérateur de chapeau haut-de-forme $P(f)$ obtenu par $f - (f \circ B)^5$ est un opérateur toujours positif qui repère à merveille les pics présents dans le signal f . Par contre, les vallées seront localisées par l'opérateur suivant $V(f) = (f \bullet B) - f$, signal lui aussi toujours positif. Avec ces deux filtres, il est possible de détecter les quelques pics ou vallées accentués.

Toutes ces démarches se réduisent à exiger une information a priori concernant l'image analysée. Si la pratique s'accorde mieux à un marquage des vraies régions par un humain ou par un algorithme semi-automatique, le problème d'une détermination automatique et universelle est loin d'être résolu. Ceci explique que nous n'ayons appliqué la ligne de partage des eaux que pour segmenter des régions à variations lentes.

Conclusions du chapitre

Ce chapitre a présenté toute une panoplie d'opérations de la morphologie mathématique que nous utiliserons par la suite. La combinaison d'opérations simples que sont l'érosion et la dilatation aboutissent à définir l'ouverture dont il sera abondamment question lors du traitement de la forme. Ces opérations, ainsi que la reconstruction, servent aussi lors de la segmentation. Elles délimitent des masques géodésiques dans lesquels pourront s'étendre les textures. Quant à la ligne de partage des eaux, son application est destinée à terminer la segmentation en détectant les régions à variations lentes, présentes dans une image.

5. L'ouverture $f \circ B$ et la fermeture $f \bullet B$ d'une fonction sont définies à la page 106.

Chapitre 3

De la représentation au codage des textures

Introduction

Dans une chaîne de télécommunications, le signal de texture est transformé en un signal analogique ou numérique, puis il est adapté au canal et transmis. On pourrait être tenté de le traiter comme tel sans se soucier de sa signification physique et de son interprétation. Une approche orientée par la notion d'objet prend cette démarche à contre-pied : la représentation des textures $\rho(T)$ précède tout codage. Cette dernière est construite pour conserver les éléments visuellement importants mais il importe également de décorréler au mieux les textures des contours sinon l'information des contours serait transmise par deux voies distinctes. Dans le schéma de codage que nous avons retenu, le décodeur reçoit d'abord l'information concernant les contours. Elle est donc disponible au récepteur lorsque parvient l'information relative aux textures. Le thème qui guide nos travaux consiste à utiliser au maximum l'information des contours pour réduire le débit des textures.

Plan du chapitre Le chapitre s'articule autour du concept d'extrapolation de texture. Dans un premier temps, nous décrirons sommairement quelques techniques existantes de représentation des textures.

La cohérence de toute la chaîne depuis la segmentation jusqu'au codage, thème central de ce travail, est réalisée grâce à une représentation spectrale de la texture. Le traitement spectral des textures est introduit à la section 3.2. Cette description s'accompagne d'une modélisation du problème d'extrapolation.

Dans la section 3.3, nous proposons l'expression d'un algorithme tout à fait général d'extrapolation par itération. Il apparaît que de nombreuses solutions exposées précédemment dans la littérature correspondent à des cas particuliers de l'algorithme général, mais que la forme de l'opérateur se prête aussi à la création de plusieurs procédés pratiques d'extrapolation. La section 3.4 fournit quelques résultats d'extrapolation de textures et évalue la performance de schémas inspirés de l'analyse en sous-bandes.

L'extrapolation de textures se réfère à la notion de facteur de sélection, lié au nombre de raies spectrales qui suffisent à décrire la texture. La relation entre le facteur de sélection et un véritable taux de compression est examinée à la fin de ce chapitre. Nous évoquerons

également l'utilisation de l'extrapolation pour une prédiction dans une phase de codage et la possibilité de synthétiser une texture.

3.1 Représentation des textures

3.1.1 Représentation d'une texture dans le domaine spatial

Le profil de luminance que dessine l'aspect d'une région peut être décrit à l'aide d'une fonction à deux dimensions; c'est à tout le moins l'interprétation la plus sommaire et la plus ancienne du signal de luminance. Dans certaines disciplines dont la morphologie, la fonction de luminance est vue comme une surface topographique sur laquelle s'appliquent les opérateurs. La technique de segmentation par ligne de partage des eaux se réfère à ce vocabulaire de géologue ou d'hydrologue.

Une simple description du signal de luminance consiste à choisir une fonction polynomiale. Il est évidemment impensable de réaliser cette représentation de textures avec un seul polynôme qui serait alors de degré prohibitif. La solution proposée par l'analyse numérique est d'utiliser une fonction définie par morceaux de polynômes de degré 2 ou 3, procédé qu'a repris KUNT [43] dans ses articles relatifs aux méthodes de codage de 'seconde génération'. Le codeur du projet RACE Morpheco [4] fait également usage d'une approximation polynomiale après orthogonalisation. Ce codeur réalise aussi le codage de certaines textures par une quantification vectorielle ou par la méthode BTC¹ (voir RABBANI *et al.* [77] pour une description de ces techniques). Citons encore les travaux récents de LEOU et CHEN [51] qui présentent une description de textures unidimensionnelle simplifiée, résultant d'un parcours en spirale de la région.

La littérature suggère de nombreuses techniques d'approximation et d'interpolation, tant locales que globales. On pourra consulter les "Numerical Recipes" [76] ou l'étude de FRANKE [23] pour une liste de ces méthodes. Dès lors que nous disposons d'une description de la fonction de luminance, il suffit d'envoyer les paramètres de la fonction de représentation pour transmettre l'information. La réduction de débit peut être importante.

Il importe que la reconstruction de la texture soit visuellement pertinente, c'est-à-dire que la texture codée soit interprétée comme la texture originale. Bon nombre des techniques citées ci-dessus souffrent, de ce point de vue, d'un manque d'interprétation physique. De plus, ces méthodes n'ont que rarement été appliquées au traitement d'image numérique.

Pour ces raisons, nous n'avons pas retenu de *traitement spatial* d'extrapolation. Nous ne prétendons pas qu'une telle méthode mène à des résultats décevants; mais les avantages (exposés plus loin) issus de l'emploi d'une analyse spectrale et l'expérience acquise dans ce domaine nous ont poussé à choisir un traitement spectral. D'autre part, il faut voir l'extrapolation comme un outil général qui peut même servir dans l'utilisation de techniques de codage spatiales en étendant la texture à un domaine plus adéquat, ou comme un filtre.

3.1.2 Représentation de la texture dans le domaine spectral

Les transformées spectrales utilisées habituellement en traitement d'image (transformée de Fourier ou transformée en cosinus) possèdent des propriétés décorrélatives remarquables.

1. Block Truncation Coding

En effet, elles approximent correctement la transformation décorrélatrice idéale : la transformée de Karhunen-Loève (KLT). Celle-ci transforme les échantillons de départ en un ensemble de coefficients non corrélés. Elle est malheureusement inapplicable ici car elle nécessite la connaissance des propriétés statistiques de corrélation des échantillons de données.

Grâce à cette décorrélation et puisque les transformations orthonormées conservent l'énergie, elles réalisent une concentration d'énergie dans les coefficients les plus bas. De plus, les transformations orthogonales offrent des possibilités de codage pondératif.

En résumé, ces transformations possèdent un double avantage :

- d'une part, elles réalisent une décorrélation et la théorie de l'information nous apprend que le débit binaire est minimum pour des échantillons codés indépendamment ;
- d'autre part, l'œil étant moins sensible aux hautes fréquences, le pas de quantification peut être plus important pour les coefficients d'ordre élevé.

Ces propriétés expliquent l'usage généralisé des transformations orthogonales en traitement d'image numérique. Combinées à une quantification pondérée et à un codage entropique, elles permettent d'atteindre des débits inférieurs à 1 bit par pixel pour la plupart des images (cf. MACQ [55]).

Dans le problème qui nous concerne, nous avons segmenté l'image en différents objets ou segments. Parmi ceux-ci figurent des régions au contenu spectral limité. L'objet peut donc être décrit à l'aide d'un nombre restreint de coefficients transformés. Encore faut-il que la forme de la région se conjugue à une description spectrale discrète définie sur un domaine rectangulaire ! L'extrapolation des textures exposée ci-après apporte une solution pour l'analyse de textures définies sur un support de forme quelconque.

3.2 Extrapolation de textures

3.2.1 Objectifs

Conformément au modèle d'image proposé plus loin (cf. section 4.1.2), trois types de régions sont issues de l'étape de segmentation : les régions à variations lentes, les régions texturées et les régions actives à contenu énergétique élevé. Rien ne les différencie quant à leur contour puisque nous avons pris le parti de caractériser l'aspect des objets, information contenue dans $\rho(T)$ mais pas le moins du monde la forme $\rho(X)$. Les *régions texturées* et les *régions à variations lentes* présentent une certaine régularité dans leur motif, si bien qu'il paraît possible d'étendre les motifs au-delà de la forme apparente des objets. Cette extension appelée *extrapolation* est à la base de trois idées maîtresses qui animent nos travaux :

1. *Détection de textures par croissance de germes.*

Des petites régions au contenu *spectral local* spécifique sont extraites grâce à l'utilisation de filtres gaussiens. Ces régions constituent le cœur de textures ou de régions basses fréquences. Pour obtenir les régions complètes, nous avons imaginé d'étendre progressivement ces germes et de comparer, dans les zones de croissance, les valeurs

des pixels obtenues par extrapolation et les valeurs véritables des pixels. Si la correspondance est ‘bonne’, la surface est étendue². Par ce procédé d’extension inhabituel s’établit une liaison entre pixels, qui englobe à la fois l’aspect local et l’aspect global des rendus de régions. De plus, grâce à la correspondance établie entre pixels d’une même région dès le procédé de segmentation, le codage qui exploiterait cette uniformité de propriété, s’effectuera à moindre coût. De fait, par le regroupement de pixels ayant une même *signature spectrale locale*, les régions formées se coderont avec un nombre restreint de fréquences. Ce traitement est propre aux textures ; les régions actives nécessitent la transmission d’une grande partie de leur spectre. Si besoin en était encore, nous rappelons qu’il est toujours souhaitable d’assurer une cohérence totale depuis l’étape de segmentation jusqu’à celle du codage.

2. *Extrapolation à un rectangle pour faciliter le codage, décorrélation des contours et des textures, et synthèse de textures à partir d’un échantillon.*

Les objets issus de la segmentation possèdent une forme quelconque, ce qui complique le traitement spectral de codage essentiellement pour deux raisons :

- (a) les transformations décorrélatives classiques (DFT³ ou DCT) ne peuvent être calculées que sur des domaines réguliers, rectangulaires par exemple, et pas sur un domaine quelconque ;
- (b) une transformée orthogonale appliquée directement au bloc rectangulaire contient en partie l’information relative au contour parce que la texture n’occupe pas la totalité du bloc. Or, la forme étant codée séparément, il est totalement inutile de l’envoyer à nouveau indirectement par l’entremise de la représentation de textures $\rho(T)$. Étendre une texture jusqu’à ce qu’elle occupe entièrement le rectangle circonscrit à une région supprime la redondance d’information due au contour, contenue dans $\rho(X \cap T)$. Dans la mesure où seul un certain nombre de fréquences sont reprises pour décrire et coder les textures, il importe de concentrer l’information relative à la texture et de se départir complètement de la forme.

La figure 3.1 illustre le procédé d’extrapolation et l’intérêt de la méthode pour du codage. Le signal de gauche n’est connu qu’à l’intérieur d’un domaine (en forme de fleur dans cet exemple). Après extrapolation, 3% du contenu spectral suffisent à reconstituer le signal de droite. Dans cette dernière image, sans la surimpression du contour du domaine en noir, il serait impossible de dire où s’arrête la texture connue bien que le nombre de raies sélectionnées soit petit. Par le biais de la sélection des raies spectrales, le codeur privilégie inéluctablement certaines directions ce qui, sur le plan du principe, se rapproche très fort des méthodes de codage orientées, proposées par KUNT [44]. Cette analyse est saine dans la mesure où, comme l’affirme DAUGMAN [12], l’aire visuelle du cortex cérébral humain serait composée de cellules sensibles à des fréquences et des orientations bien précises.

Pour être complet sur la question de l’extrapolation en vue du codage, nous devons encore mentionner les travaux de GILGE *et al.* [29]. Ces derniers ont décidé

2. Pour y parvenir, nous proposons un critère de décision dans le chapitre consacré à la segmentation.

3. Discrete Fourier Transform

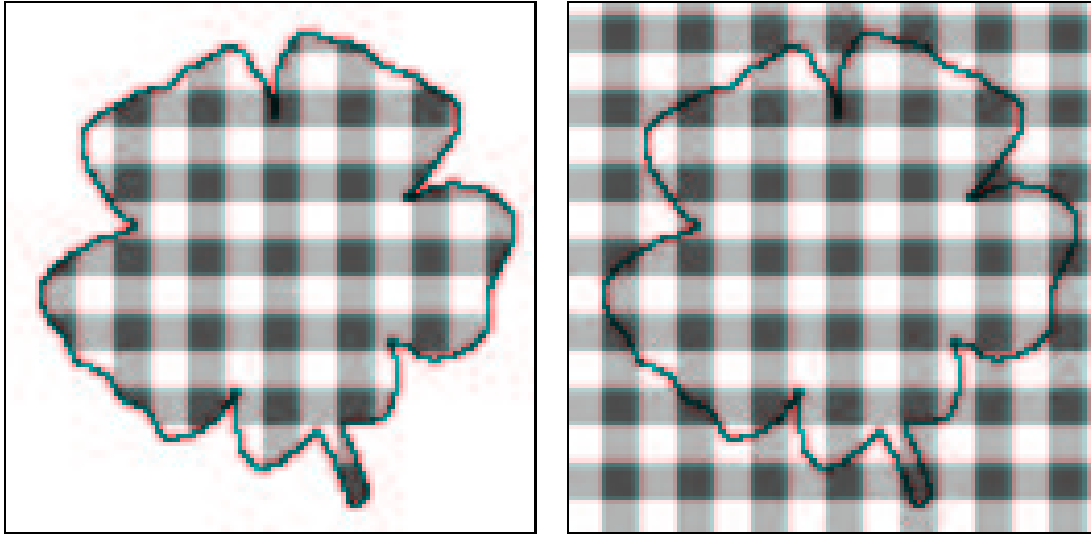


FIG. 3.1 - Exemple d'extrapolation d'un signal. L'image de droite est l'image extrapolée avec quelques coefficients spectraux seulement (3%).

de transformer toutes les fonctions de base d'une DCT pour les adapter à la forme d'une région; il suffit d'ajouter la fonction fenêtre $w(n)$ (définie plus loin) dans la définition du produit scalaire et d'orthogonaliser les fonctions de base les unes par rapport aux autres pour y parvenir. On dispose alors d'un jeu de nouvelles fonctions de base adaptées à la forme de la région. Cette méthode présente plusieurs défauts majeurs:

- les fonctions de base diffèrent pour toutes les régions, ce qui signifie qu'il faut les calculer à l'émetteur et au récepteur pour chaque région. Pour cette raison, les approximations sont souvent limitées à quelques coefficients et, en conséquence, elles ne s'appliquent qu'à des régions à variations lentes. D'autre part, limiter le nombre de fonctions de base s'effectue en prenant les fonctions aux variations lentes, par exemple les fonctions proches de la composante continue (DC) dans une description en cosinus. À part des arguments d'ordre empirique, rien ne garantit que ces fonctions contribuent activement à la description de la texture car il se peut que les coefficients importants correspondent à des fonctions de base plus éloignées du DC;
- le nombre total de fonctions de base est fixé par le rectangle qui inscrit la région, produisant un *surplus inutile de coefficients spectraux*;
- malgré les efforts de certains auteurs dont PHILIPS [74], le temps de calcul de l'orthogonalisation est considérable et largement supérieur aux méthodes que nous proposons dès que l'approximation ne se limite pas à quelques fonctions de base;
- quel sens physique faut-il accorder au processus d'orthogonalisation? Ainsi, dans une région en forme de 'C', le sommet de la fonction cosinus d'ordre (1,0) épousera la courbure après orthogonalisation, alors qu'il dessine une ligne verticale dans un domaine rectangulaire.

Durant l'expérimentation nous est venue une autre idée. *Une partie du signal suffit-elle à reconstruire toute la texture ?* C'est une manière de se rapprocher de la notion de *texton* de JULESZ [40], sorte d'élément minimal de représentation de texture. Le procédé d'extrapolation que nous avons développé permet de synthétiser des textures à partir d'un échantillon⁴ minimal de texture. Il reste alors à choisir l'échantillon et à déterminer sa taille et sa position, en prenant soin de sélectionner de préférence un échantillon défini sur un rectangle entier pour se défaire des problèmes de décorrélation d'information de contour et de texture sur ce segment. *La synthèse ou reconstruction de texture, menée au moment du codage, ouvre une voie nouvelle vers des 'procédés d'hyper-compression'.* Assez curieusement, même des régions actives homogènes peuvent être synthétisées et fournir un résultat visuellement acceptable (cf. figure 3.17 à la page 74). Nous reviendrons sur ce sujet dans la conclusion de ce chapitre.

3. Prédiction des zones découvertes et identification des régions dans une séquence d'images.

Les objets, qui se déplacent d'une image à l'autre, couvrent ou découvrent des zones de l'image. Si le cas des zones couvertes n'est pas critique parce que la surface n'est plus codée, il en va tout autrement des régions découvertes. Apparemment, le seul moyen de prédire l'aspect de la région consiste à extraire l'information des images futures ; dans ce schéma, la variable temps sert à détecter le déplacement d'objet, alors que l'outil d'extrapolation sert à prédire les zones découvertes. Quelquefois, par un mécanisme similaire à celui de l'extraction de textures par croissance de germes, certaines parties des zones découvertes se prédisent correctement par une extrapolation spatiale des régions voisines. La proximité spatiale permet de penser qu'il est possible de prédire en partie la texture de certaines régions découvertes par l'analyse d'une image fixe. Dans certaines séquences d'images comme en vidéotéléphonie, l'extrapolation de l'arrière-fond convient bien à une prédiction des zones découvertes par un mouvement de tête.

Un dernier objectif visé par l'extrapolation de textures est l'identification de textures entre images successives. Le recouvrement d'objets interdit d'élaborer une méthode d'identification axée uniquement sur la forme des objets ; il suffirait d'un léger mouvement relatif pour déformer plusieurs objets et perdre la correspondance. Le signal extrapolé offre l'occasion de s'affranchir en grande partie de la forme de l'objet. Un examen du spectre des signaux extrapolés permet alors de rétablir les correspondances entre objets. Cette idée est illustrée à la section 4.5 du chapitre relatif à la segmentation.

La démarche habituelle d'extrapolation consiste à construire d'abord une approximation ou une interpolation du signal sur son support à partir d'un modèle. Les fonctions de décomposition mises en œuvre lors de l'approximation sont ensuite étendues hors support en conservant les coefficients des fonctions exprimées à l'intérieur du domaine de départ. La littérature propose bien des techniques d'interpolation et d'approximation : surfaces-splines, méthode de Shepard, techniques d'éléments finis, etc. La méthode d'extrapolation de texture que nous avons mise au point part des mêmes principes : elle évalue les paramètres significatifs de la texture sur le segment de départ et le reproduit tout autour

4. Échantillon de texture est pris au sens d'une 'partie du signal de texture'.

bien que, à cette fois, le traitement ne se fasse plus dans le plan spatial mais dans le domaine spectral.

3.2.2 Éléments de théorie

Introduction à la déconvolution

Les algorithmes d'extrapolation travaillant dans un plan transformé sélectionnent dans celui-ci des amplitudes et des positions. Malheureusement, pour le calcul du spectre, la région est étendue à un support rectangulaire si bien que le spectre d'une région de forme quelconque comprend l'information propre au motif à étendre mais aussi celle due au contour. Pour déjouer l'imbrication des deux informations, nous désirons estimer le spectre du seul motif. Au premier abord, le problème paraît insoluble car l'imbrication des informations introduit une dispersion spectrale subséquente (voir par exemple l'article de FRANKE [24], figure 2) qui interdit toute sélection efficace des fréquences. La modification spectrale due à l'imbrication est connue sous le nom d'*effet de fenêtre*. Dans une image, la fenêtre est une fonction bidimensionnelle w valant 1 à l'intérieur de l'objet et 0 à l'extérieur. La fenêtre est en quelque sorte l'*indicatrice* de la forme de l'objet traité. Pour supprimer l'influence du contour, une solution consiste à extrapoler la texture jusqu'à occupation complète d'un bloc rectangulaire. De cette façon, le réseau de données ne contient plus la moindre information concernant le contour mais uniquement l'aspect.

Modèle

Pour la commodité du développement, nous utiliserons une notation unidimensionnelle. Le passage à un signal bidimensionnel ne présente pas de difficulté particulière. Les signaux de variable n sont des échantillons des signaux analogiques correspondants.

Soit $y(n)$, le signal observé. $y(n)$ s'obtient par '*fenêtrage*' ou '*apodisation*' de la fonction⁵ $f(n)$ à extrapoler :

$$y(n) = w(n)f(n) \quad n = 0, 1, \dots, N - 1 \quad (3.1)$$

avec

$$w(n) = \begin{cases} 1 & \text{pour } n \in D_w \\ 0 & \text{si } n \notin D_w \end{cases} \quad (3.2)$$

D_w est la partie du domaine de la fonction $w(n)$ sur laquelle la valeur de la fenêtre est 1. Bien qu'inclus dans $\mathbb{Z}$, cet ensemble n'est pas nécessairement un intervalle. Cette précision compte pour le traitement d'image car, si le traitement s'effectue d'abord dans une direction puis dans la direction orthogonale à la première par séparation de variables, l'échantillonnage d'une ligne conduit à définir un ensemble D_w composé de plusieurs parties de $\mathbb{Z}$.

5. Les fonctions dont il est question ici sont des applications de $\mathbb{Z}$ dans $\mathbb{R}$. Si les niveaux de gris d'une image sont forcément des valeurs discrètes, il faut obligatoirement travailler dans $\mathbb{R}$ durant la phase d'extrapolation, sous peine de perte de précision importante. La quantification aura lieu à la fin.

La détermination de la fonction $f(n)$ à l'extérieur de l'ensemble D_w requiert une modélisation plus fournie du problème sans quoi il n'offre aucune solution. *Nous choisissons l'hypothèse qui considère que l'équation 3.1 résulte de l'action d'un système linéaire, invariant en translation.* L'équation s'exprime donc aussi par une convolution dans le domaine spectral (cf. CANDY [8]). Par analogie avec le problème classique qui fait correspondre une convolution temporelle à une multiplication dans un domaine transformé et, comme les transformées directe et inverse sont des opérations duales, le problème d'extrapolation se ramène à la convolution de la transformée de $f(n)$ par celle de $w(n)$ ⁶

$$\mathcal{Y}(k) = \mathcal{W}(k) \otimes \mathcal{F}(k) \quad k = 0, 1, \dots, N - 1 \quad (3.3)$$

Cette équation indique clairement que le spectre de la fonction à extrapoler est altéré par le spectre de la fenêtre $\mathcal{W}(k)$, effet dont il convient de se débarrasser au mieux.

La figure 3.2 montre une ligne d'une texture⁷ $y(n)$ et le signal idéal $f(n)$ qui lui correspond dans les deux domaines. Pour bien distinguer la fenêtre $w(n)$ dans le domaine spatial, nous avons placé la valeur de la moyenne, c'est-à-dire 196, là où la fonction aurait dû s'annuler. Au vu des signaux, il est facile de comprendre que le passage du spectre $\mathcal{Y}(k)$ à celui du signal idéal est délicat, d'autant que le spectre d'une fonction $w(n)$ rectangulaire perd beaucoup d'énergie dans les lobes secondaires.

Forme itérative des opérateurs de déconvolution

Une simple inversion ne permet pas de résoudre le problème de la déconvolution. Et si certains auteurs affirment y parvenir (SABRI et STEENAART [83]), nous émettons de nettes réserves sur la validité des résultats rapportés pour deux raisons : les valeurs propres de la matrice d'inversion sont très proches de 1, compliquant à l'extrême les résolutions numériques, et puis, nous ne sommes pas parvenu à reproduire les résultats présentés ni à obtenir des résultats similaires sur d'autres signaux ! Ceci explique sans doute pourquoi les auteurs ont eu recours à une résolution itérative, usage usuel pour les problèmes numériques. Le principe est simple : par le choix d'un opérateur O , on construit une suite de valeurs $\dots, f_i(n), f_{i+1}(n), \dots$ obtenues par

$$f_{i+1}(n) = O[f_i(n)] \quad (3.4)$$

Déterminer un opérateur itératif à partir d'une équation est assez direct. Encore faut-il que la convergence du procédé itératif soit assurée et que la solution soit unique ! Ceci n'est pas garanti surtout que dans certains cas l'opérateur se modifie au fil des itérations. Le processus est alors du type : $f_{i+1}(n) = O_i[f_i(n)]$. Le procédé de déconvolution sélective développé plus loin est précisément un *opérateur adaptatif* de ce type.

Illustration :

L'algorithme approximant une fonction par une somme de fonctions de base, croissante à chaque itération, génère un processus itératif. L'erreur d'approximation diminue à chaque étape ou reste à la rigueur constante (c'est ce qui se produit si une fonction de base nécessaire à la reconstruction est volontairement écartée).

6. Le facteur qui se place devant l'équation est régi par le type de transformée, nous évitons d'en placer un. C'est donc une relation valable à un facteur près.

7. Il s'agit d'une ligne centrale de la texture de la figure 3.8.

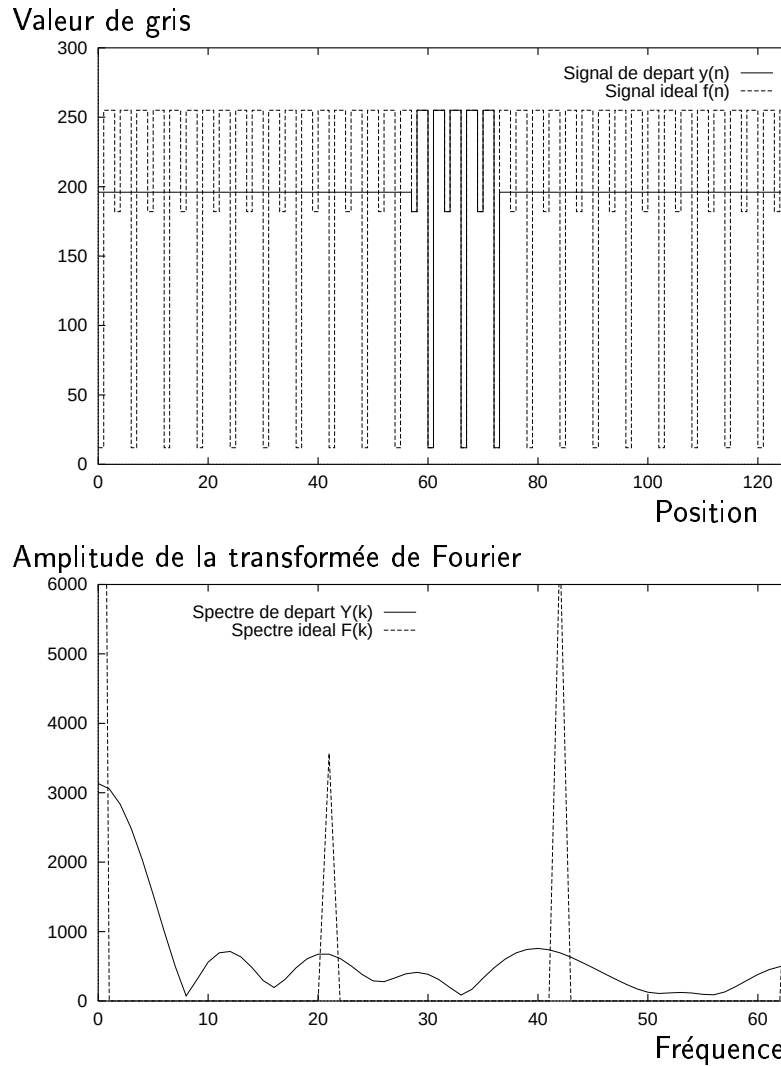


FIG. 3.2 - Un signal unidimensionnel dans le domaine spatial et dans le domaine spectral.

Approche déterministe ou aléatoire?

En toute généralité les échantillons $f(n)$ peuvent provenir d'un signal déterministe ou aléatoire. La théorie du traitement du signal rattache quantité de méthodes et d'algorithmes à chacune de ces hypothèses. Pour y voir plus clair, nous avons dressé un tableau récapitulatif à la figure 3.3.

L'*analyse déterministe* veille à tirer des renseignements de la séquence $f(n)$: étude de la moyenne, recherche d'une fréquence dominante dans la transformée de Fourier, conditionnement du signal pour éviter le repli de spectre, ...

L'*analyse 'corrélatoire'*, terme emprunté à DUVAUT [21], introduit un outil de comparaison entre deux signaux décalés, la fonction d'autocorrélation $\Gamma_{ff}(\tau)$. Si ceux-ci sont déterministes, la forme de la fonction $\Gamma_{ff}(\tau)$ reste voisine des variations de $f(n)$; elle reproduit par exemple les mêmes périodicités. S'ils sont aléatoires, l'aspect de la fonction varie à chaque réalisation et dès lors ne constitue qu'une indication. Ce n'est que lorsque les signaux sont *stationnaires (faiblement au moins)*, que l'analyse corrélatoire prend son sens. À ce stade, l'*analyse spectrale* complète l'étude par une signature spectrale de la fonction

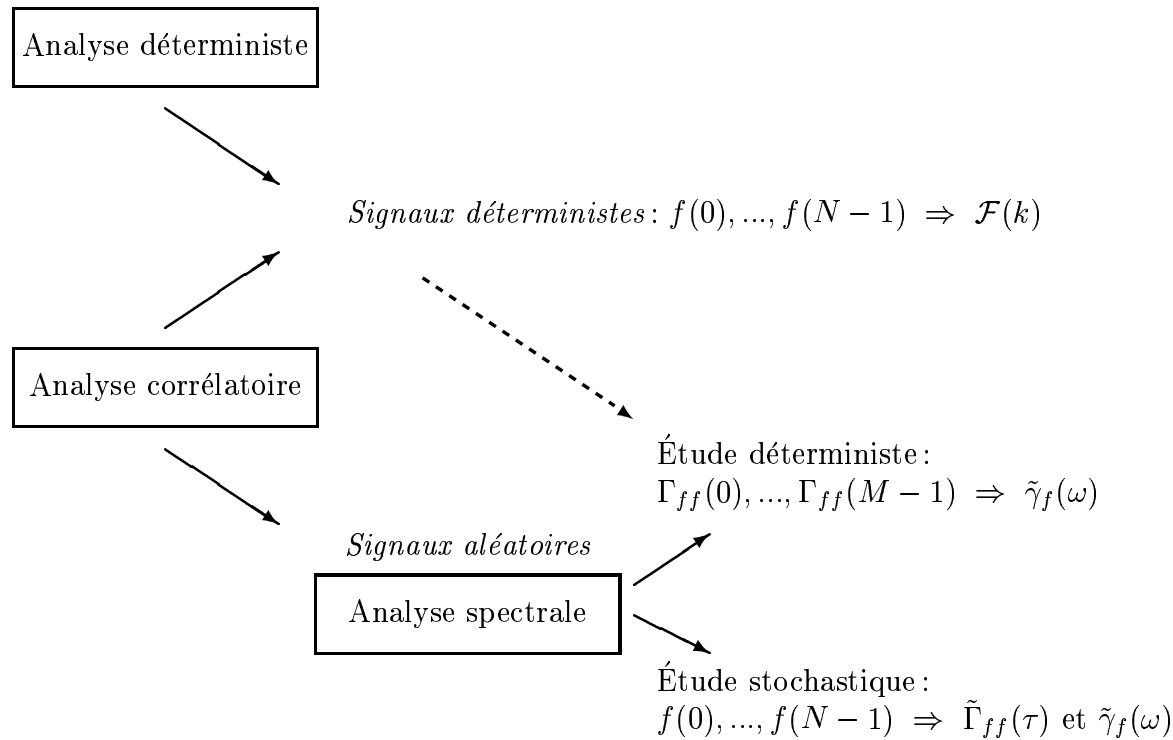


FIG. 3.3 - Récapitulatif des techniques de caractérisation des signaux.

d'autocorrélation ; il s'agit de sa transformée de Fourier appelée *densité spectrale*, notée $\gamma_f(\omega)$, et son usage, assujéti à la même contrainte de stationnarité, indique la répartition fréquentielle de l'énergie.

Selon PAPOULIS [69], l'analyse spectrale, c'est-à-dire l'analyse corrélatrice appliquée à des signaux aléatoires stationnaires, se découpe en deux catégories dépendant de la connaissance de la fonction d'autocorrélation. Supposons que soient connus M échantillons $\Gamma_{ff}(0), \dots, \Gamma_{ff}(M-1)$. L'étude s'apparente à une 'étude déterministe' où certains échantillons de $\Gamma_{ff}(\tau)$ sont manquants. Pour estimer la densité spectrale $\tilde{\gamma}_f(\omega)$, il faut tenir compte de cette apodisation soit en gérant l'influence de la fenêtre (méthodes non paramétriques), soit en identifiant le spectre à celui d'un modèle de données (méthodes paramétriques), ce qui conduit à une extrapolation de $\Gamma_{ff}(\tau)$ au-delà du segment connu. Souvent on utilise des modèles autorégressifs ou à moyenne mobile. Parfois le signal est supposé à bande spectrale limitée ou alors l'objectif consiste à maximiser l'entropie de la densité spectrale. Il arrive aussi qu'aucun échantillon de la fonction d'autocorrélation ne soit à disposition. Ce cas, repris sous le terme d'une 'étude stochastique', nécessite le calcul conjoint des estimées $\tilde{\Gamma}_{ff}(\tau)$ et $\tilde{\gamma}_f(\omega)$. Nous renvoyons à DELOGNE [13] pour une discussion complète de l'analyse spectrale.

Ce survol des techniques spectrales suffit à comprendre la réelle difficulté de la sélection d'un modèle. Nous avons jugé préférable de restreindre le travail à une analyse déterministe. Pourquoi un tel choix ?

- (1) La raison principale est la facilité de mise en œuvre : le recours à l'analyse spectrale

nécessite un complément de modélisation, or aucune méthode ne satisfait tous les points de vue (qualité de l'estimation, réduction du bruit, minimisation des biais, etc). Il ne faudrait pas en conclure que nous perdons tout bénéfice des méthodes propres à des signaux aléatoires. En effet, les méthodes s'appuient sur des algorithmes qui eux s'appliquent à toutes les analyses. Par exemple, l'extrapolation itérative d'une séquence, à bande étroite, de signaux déterministes ou de la fonction d'autocorrélation, peut se réaliser de la même manière.

- (2) Un autre argument justifiant notre choix est la mise en doute de l'hypothèse de stationnarité qui sous-tend les développements de l'analyse spectrale. HUNT [37] met sa validité en cause dans le cas d'une image. Pour y remédier, il propose de stationnariser le signal, de le traiter et de revenir au signal de départ par inversion du processus de stationnarisation.
- (3) Enfin, si l'analyse spectrale achève une bonne discrimination de fréquences, elle remplit moins notre objectif de codage. De fait, les textures sont représentées par des échantillons de la transformée de Fourier du signal extrapolé et non de sa fonction d'autocorrélation. Une analyse corrélatoire devra donc se doubler d'un schéma pour convertir ces types d'information.

En conclusion, les développements théoriques repris dans ce chapitre dérivent d'une approche déterministe au problème d'extrapolation, quoique les algorithmes étudiés soient parfois repris par la littérature dans d'autres contextes, tel celui de l'analyse corrélatoire.

3.3 Description d'un opérateur d'extrapolation itératif général par représentation matricielle

Les principales notations intervenant dans la formulation en extension de l'opérateur d'extrapolation sont les suivantes. Soient M , l'expression matricielle d'un opérateur de transformation, M^{-1} , son inverse, et Π , un opérateur de filtrage exprimé sous forme matricielle agissant sur le spectre. Les opérations de sous et sur-échantillonnage se noteront respectivement $[\downarrow 2]$ et $[\uparrow 2]$. L'opérateur D réalise l'application de la fenêtre, c'est-à-dire qu'il met les valeurs correctes à l'intérieur de la fenêtre et laisse les autres valeurs intactes. Et finalement, l'opération d'écritage T rajuste la dynamique des valeurs de $f(n)$; son rôle est mineur.

Définition 19 *L'opérateur général d'extrapolation, noté G , découle de la mise en cascade, de droite à gauche, de toutes les opérations décrites ci-dessus. En voici l'expression :*

$$G = TDM^{-1}\Pi_2M[\uparrow 2]M^{-1}SM[\downarrow 2]M^{-1}\Pi_1M \quad (3.5)$$

La matrice S sélectionne certaines fréquences; elle interviendra plus tard.

L'algorithme itératif correspondant est :

$$f_{i+1}(n) = Gf_i(n) \quad (3.6)$$

L'expression de G mérite quelques explications :

- $M^{-1}\Pi_1 M$ et $M^{-1}\Pi_2 M$ exécutent des opérations de filtrage dont les objectifs sont respectivement d'éviter le repli de spectre et d'interpoler correctement les fonctions. Ce schéma correspond typiquement à celui d'une décomposition en sous-bandes.

L'insertion du filtre Π_1 permet le sous-échantillonnage du vecteur⁸ $f(n)$. C'est là tout l'intérêt de ce nouveau schéma qui ne travaille plus avec N échantillons, mais avec un nombre d'échantillons réduit. Cet élément est fondamental dans l'application d'un schéma itératif où le nombre de données à traiter conditionne directement la durée du processus.

- Les opérations composant G servent de guide à la conception d'autres algorithmes mais elles n'indiquent pas nécessairement comment se réalise l'algorithme. Ainsi, il s'avère souvent plus commode de filtrer le signal par convolution circulaire ou cyclique, sans transiter par le plan transformé. Ou encore, l'opération de transformée inverse suivie du sous-échantillonnage ne nécessite pas le calcul d'un échantillon sur deux.
- La matrice S constitue le cœur de l'algorithme. Elle agit en sélectionnant un certain nombre de raies dans le spectre, les autres sont mises à zéro. La sélection dépend du signal de texture.

Convergence et unicité

L'expression de l'opérateur d'extrapolation itératif couvre par sa généralité la plupart des algorithmes cités dans la littérature. On conçoit le profit à tirer d'une condition de convergence et d'unicité. Cette dernière fait l'objet d'un théorème :

Théorème 20 (Convergence et critère d'unicité stricte) *Quel que soit le vecteur de données initial $f_0(n)$, l'itération $f_{i+1}(n) = Gf_i(n)$, où G est fixe et a l'expression indiquée par la relation 3.5, converge vers une solution $f^*(n)$, unique moyennant la condition suffisante d'unicité stricte qui stipule que le nombre de raies spectrales conservées (c'est-à-dire non nulles) doit être inférieur au cardinal de la fenêtre.*

Commentaires. La démonstration de ce nouveau théorème nécessite des notions de l'analyse fonctionnelle et manipule des concepts topologiques. Elle est développée dans l'annexe A.

Si la démonstration de la convergence ne requiert pas d'hypothèse particulière quant au signal, bande passante limitée ou autre, l'algorithme n'indique malheureusement pas comment sélectionner les raies spectrales significatives. Ce qui veut dire que si théoriquement la solution est unique, l'algorithme risque de détecter et de conserver quelques raies moins significatives, effet dû au bruit qu'introduit le spectre de forme sur le spectre de texture. Pour un signal à périodicité marquée, l'algorithme détectera sans erreur les raies parce que le rapport signal à bruit est élevé. C'est la raison pour laquelle l'extrapolation de textures de synthèse pose moins de difficultés.

8. Dans ce qui suit $f(n)$ est mis parfois sous la forme d'un vecteur.

3.3.1 La matrice S , cœur de l'algorithme : vers un filtrage adaptatif?

Introduction. L'utilisation d'un procédé itératif d'extrapolation par déconvolution spectrale remonte à 1974. C'est GERCHBERG [27] qui, le premier, a appliqué la technique au traitement d'image. Quasi simultanément, PAPOULIS [68] proposa d'estimer le spectre d'un signal à bande limitée multipliée temporellement par une fonction fenêtre $w(n)$; la méthode correspond à un cas particulier de G , à savoir $G = DM^{-1}SM$. Ces auteurs partent d'un signal limité en fréquences et lui appliquent un filtre S ayant une fréquence de coupure σ_c , ce qui nécessite de connaître la valeur de la fréquence de coupure du signal non corrompu par le spectre de forme.

Le choix de la fréquence de coupure σ_c du filtre est critique : une bande passante de filtre trop petite contrarie la reconstitution du signal dans la fenêtre temporelle, et dans la situation inverse, l'algorithme perd de son efficacité et aboutit à une solution qui n'est pas unique et qui dépend de la valeur initiale ainsi que l'a montré FRANKE [24]; le théorème précédent appuie par ailleurs cette remarque. Nous avons tenté d'utiliser l'algorithme de GERCHBERG sur les textures en ignorant l'hypothèse de contenu spectral limité. Les résultats sont inutilisables parce que l'algorithme ajoute de l'énergie près de σ_c pour compenser la perte d'énergie des raies situées au-delà. Quels que soient les arguments heuristiques invoqués par PAPOULIS, il n'y a même pas convergence car l'erreur résulte d'une suite non bornée.

Déconvolution sélective. Résoudre l'extrapolation, quand aucune information relative au signal n'est disponible, doit s'accomplir autrement. Nous disposons toutefois d'une caractérisation qualitative partielle des signaux : les signaux sont des régions texturées ou des régions à variations lentes ayant une cohérence spectrale. Dès lors que la région est grande, *le spectre du signal, défini sur le rectangle qui circonscrit la région, contient une série de fréquences dominantes* et ce, malgré la perturbation introduite par la fenêtre. Cette supposition conditionne toutes les démarches qui suivent. Il est clair qu'en l'absence d'une telle hypothèse, étendre le signal n'a plus vraiment de sens. Pourtant, à des fins de codage, il est utile d'étendre tous les signaux parce que, pour un nombre de raies spectrales fixé, le signal extrapolé exprime toujours mieux le signal contenu dans la fenêtre qu'en l'absence de déconvolution. Nous y reviendrons par le biais d'une conjecture à la page 63.

Dans nos travaux, nous avons suivi le principe d'extrapolation mis en œuvre par FRANKE [24] et qui est illustré à la figure 3.4 ; il consiste à choisir $G = DM^{-1}SM$, où S opère une sélection des coefficients transformés.

À la base de l'*algorithme de déconvolution sélective* apparaît une *matrice S qui est modifiée à chaque itération*. La matrice de sélection de l'étape $i+1$ s'obtient en sélectionnant une fréquence supplémentaire non choisie dans S_i ; c'est-à-dire que, si k_s est la position du nouveau coefficient choisi, alors

$$S_{i+1}(k) = \begin{cases} 1 & \text{si } k = k_s \\ S_i(k) & \text{si } k \neq k_s \end{cases} \quad (3.7)$$

FRANKE suggère de sélectionner à chaque étape le plus grand coefficient non sélectionné auparavant. En remplissant progressivement la matrice S , le filtre est devenu *adaptatif*. Remarquons qu'il faut s'arranger pour ne conserver, à la fin du processus, qu'un faible

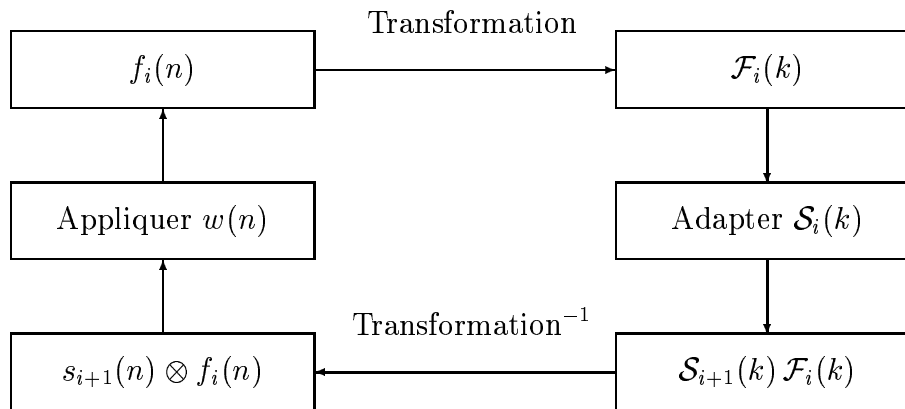


FIG. 3.4 - Bloc-diagramme de l'algorithme d'extrapolation par déconvolution sélective, proposé par FRANKE.

nombre de coefficients spectraux, inférieur à la taille de la fenêtre. Dans le cas contraire, on dépasserait le nombre de raies strictement nécessaires pour décrire le signal.

DOCAMPO et FIGUEIRAS [15] agissent à l'opposé de FRANKE en éliminant les raies peu significatives à partir d'un algorithme du gradient. Si les résultats semblent acceptables, les premières étapes du processus itératif bafouent la condition d'unicité stricte et présentent de ce fait un risque non négligeable d'instabilité. Au contraire, la déconvolution sélective extrait en premier lieu les raies importantes d'où il résulte que, en conformité avec le théorème de Parseval, l'erreur diminue au fur et à mesure.

Nous avons prolongé l'étude de FRANKE sous trois angles qui décomposent davantage le processus de déconvolution. La question de la convergence de l'algorithme et d'un ajustement de la taille des échantillons fait l'objet des deux sections suivantes. Enfin, au point 3.3.4, nous passons en revue des implémentations particulières du schéma général d'extrapolation qui entraînent une réalisation plus rapide.

3.3.2 Convergence et unicité de la déconvolution sélective

Le théorème 20 établit la convergence et l'unicité d'un opérateur fixe. Or dans l'algorithme de déconvolution, l'opérateur se modifie durant les étapes et, en toute rigueur, le théorème ne s'y applique pas ; il faudrait un nombre de boucles internes, autrement dit sans modification de S , infini avant de modifier l'opérateur. La pratique montre que le signal se stabilise après quelques boucles et que, en outre, leur nombre peut être réduit durant le processus de déconvolution. Pour assurer une démarche rigoureuse et pratique, nous avons adopté un algorithme itératif dégressif. Partant du signal original, nous sélectionnons une raie et effectuons K boucles internes ; $K = 10$ dans nos essais. Au moment d'ajouter une autre raie, K passe à $K - 1$ et ainsi de suite jusqu'au moment où $K = 1$. Cette démarche satisfait les conditions du théorème et conduit à de très bons résultats alors que prendre dès le départ $K = 1$ produit une interférence entre la stabilisation et la déconvolution, qui se répercute sur la qualité du résultat.

La figure 3.5 montre l'évolution du spectre de la figure 3.2 durant le processus de

déconvolution. Le spectre de départ est représenté au temps $t = 0$. Puis l'algorithme sélectionne la composante DC ; pour la clarté du dessin, nous avons écrêté l'amplitude de cette composante. L'effet de la déconvolution s'observe déjà près de l'origine. L'algorithme détecte ensuite les fréquences 42 et 21 au temps respectifs $t = 2$ et $t = 3$. Il en résulte un aplatissement important du spectre. Remarquons que l'amplitude des raies sélectionnées augmente au fil du temps avant de saturer, ce qui signifie qu'il ne faut pas s'arrêter trop tôt. D'autant que la fréquence 1, extraite au temps $t = 4$, ne représente plus qu'une faible partie du contenu énergétique du signal, fournissant un critère d'arrêt facile à mettre en œuvre.

Amplitude de la transformée de Fourier

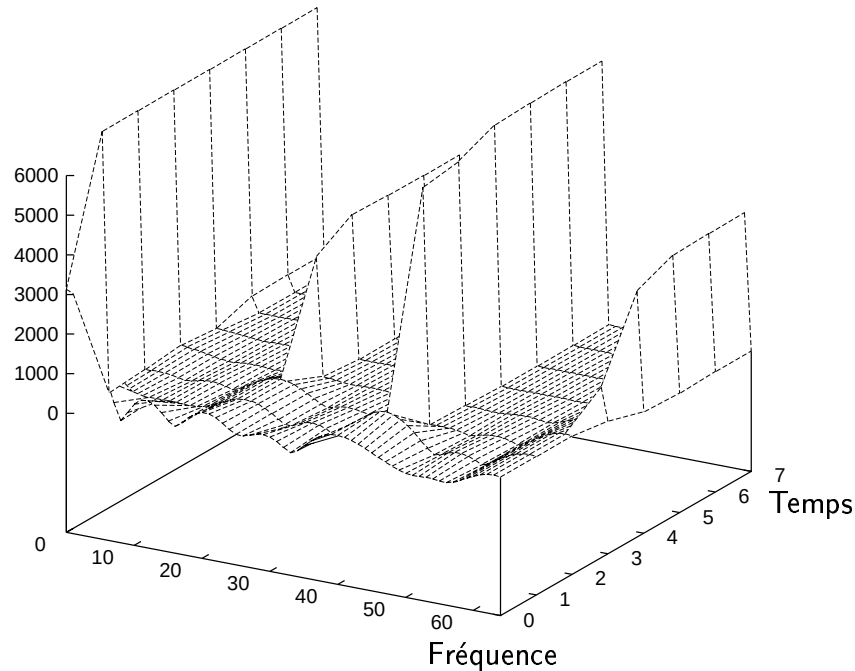


FIG. 3.5 - Évolution du spectre d'un signal unidimensionnel durant un processus de déconvolution sélective.

Les résultats de la déconvolution sélective d'une image sont présentés à la figure 3.6. Il convient de remarquer que les premières étapes de la déconvolution fournissent déjà un résultat très acceptable $-s_4(n) \otimes f_3(n)$ est obtenu avec seulement 4 raies spectrales!-, ce qui s'accompagne d'une réduction de l'entropie du spectre. La déconvolution atténue la dispersion du spectre autour des raies sélectionnées.

3.3.3 Un pré-traitement souhaitable : recherche du maximum de la fonction d'autocorrélation

Si le critère de Shannon est respecté, le signal numérique interpolé reconstitue le signal analogique de départ. Malheureusement, l'échantillonnage introduit des effets liés entre eux qui perturbent la déconvolution : la *périodisation du signal* et le *choix imposé de la fréquence d'échantillonnage*. La périodisation du signal provient de l'échantillonnage du spectre. Dans la plupart des situations, elle introduit une discordance de phase au bord du rectangle qui

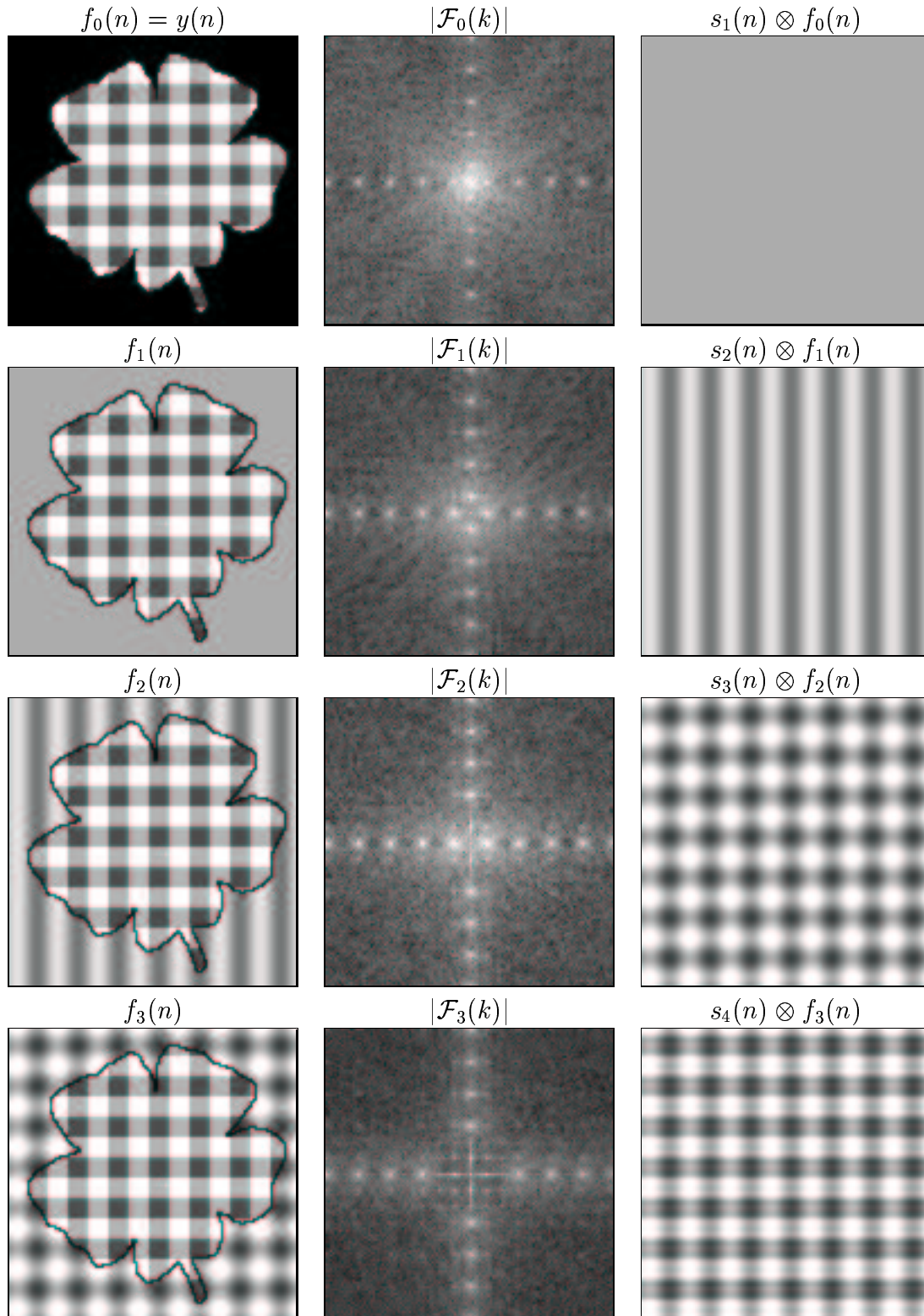


FIG. 3.6 - Premières étapes de l'algorithme de déconvolution sélective.

se traduit par un léger phénomène de dispersion dans le spectre. D'autre part, le choix de la fréquence d'échantillonnage influence aussi la qualité de la déconvolution puisqu'il indique où sélectionner les raies du signal analogique correspondant. Or, nous avons la possibilité de modifier légèrement cette fréquence en adaptant la taille du rectangle extérieur.

Pour améliorer la qualité de la déconvolution, nous proposons de recourir à l'analyse corrélatrice en *évaluant le maximum de la fonction d'autocorrélation* dans un sous-espace de son domaine de définition et en *prenant une taille de rectangle extérieur, multiple de la position de ce maximum*. En guise de justification, mentionnons que la fonction d'autocorrélation reproduit la périodicité du signal examiné.

Soit D , la partie de l'espace où le signal $f(n)$ est connu et D_p , ce domaine translaté de p . Dans l'hypothèse de signaux réels faiblement stationnaires, la fonction d'autocorrélation est définie par

$$\Gamma_{ff}(\tau) = \sum_{n \in D \cap D_\tau} \frac{1}{\mathcal{A}(D \cap D_\tau)} f(n) f(n + \tau) \quad (3.8)$$

où $\mathcal{A}$ représente l'aire. La fonction d'autocorrélation est symétrique et non nulle sur le domaine $D \oplus \check{D}$ mais, au vu du peu de signification des valeurs de $\Gamma_{ff}(\tau)$ pour τ élevé⁹, nous n'avons considéré que la moitié inférieure de $D \oplus \check{D}$. D'autre part, les valeurs $\Gamma_{ff}(0)$ et $\Gamma_{ff}(1)$, bien qu'élévées, ne permettent pas d'adapter la taille de la fenêtre extérieure; nous les avons également écartées. Dans le cas de signaux bidimensionnels, le procédé est similaire: recherche du maximum de $\Gamma_{ff}(\tau)$ dans le quart inférieur amputé des 4 valeurs à l'origine et recherche dans la direction perpendiculaire si une coordonnée vaut 0 ou 1.

L'augmentation de la variance de l'estimation de $\Gamma_{ff}(\tau)$ pour τ élevé est particulièrement visible sur le relief de la figure 3.7. On y discerne aussi la périodicité marquée de la fonction près de l'origine.

À la figure 3.8, nous illustrons l'influence des phénomènes perturbateurs précités sur la qualité de l'extrapolation d'un échantillon de texture central. La comparaison des images confirme le besoin d'adapter la taille de rectangle extérieur à la période de l'échantillon. En effet, pour passer de l'image de gauche à l'image de droite, nous avons fait en sorte que la taille du rectangle extérieur soit un multiple du maximum de la fonction d'autocorrélation.

Bien que les problèmes de l'échantillonnage de la transformée de Fourier soient connus depuis longtemps, aucune réalisation de déconvolution proposée dans la littérature n'avait considéré le besoin de modifier la taille du rectangle. Peut-être est ce dû à la lenteur du calcul des transformées pour des tailles qui ne sont pas des puissances de 2? Si dans certaines situations 'bien conditionnées', la modification de la taille se limite à accélérer le processus de convergence, la question prend tout son sens quand on sait que les deux images de la figure 3.8 sont les meilleurs signaux obtenus par déconvolution sans ou avec adaptation de la taille du rectangle extérieur (ce sont respectivement les images de gauche et de droite).

9. Pour des séquences finies, on utilise parfois l'estimateur

$$\Gamma'_{ff}(\tau) = \sum_{n \in D \cap D_\tau} \frac{1}{\mathcal{A}(D)} f(n) f(n + \tau) \quad (3.9)$$

qui, bien qu'il introduise un biais, réduit la variance pour les valeurs de τ élevées (voir PAPOULIS [69, page 491–492] ou DUVAUT [21, page 443]) Mais puisque nous avons délibérément ignoré ces valeurs, l'expression 3.8 suffit. La précision des estimateurs stochastiques, cœur de cette discussion, est examinée dans [13].

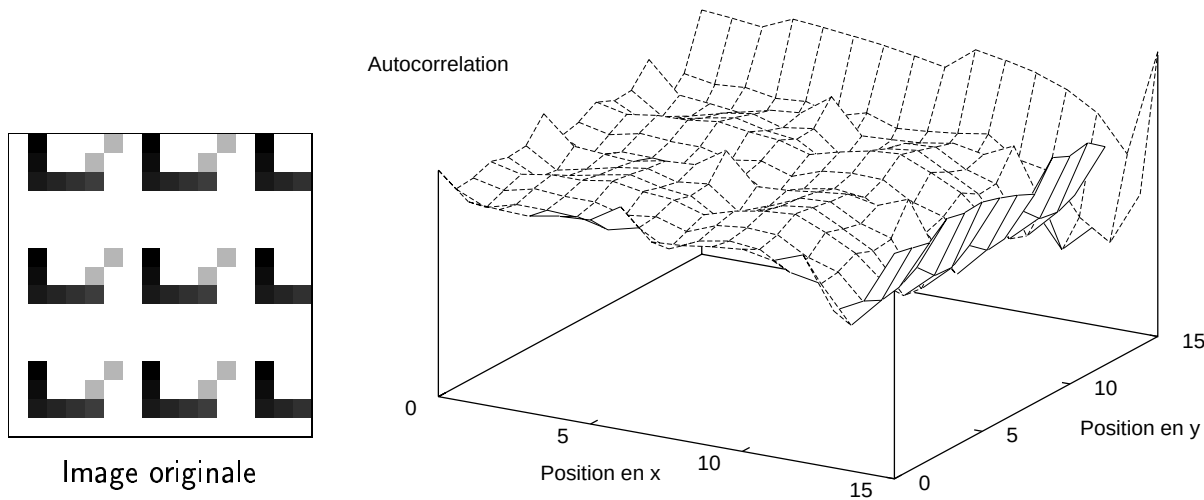


FIG. 3.7 - Fonction d'autocorrélation d'une image 16x16 périodique. Près de l'origine, la fonction d'autocorrélation présente une périodicité similaire à celle de l'échantillon mais elle perd tout intérêt pour des décalages horizontaux ou verticaux proches de 15.

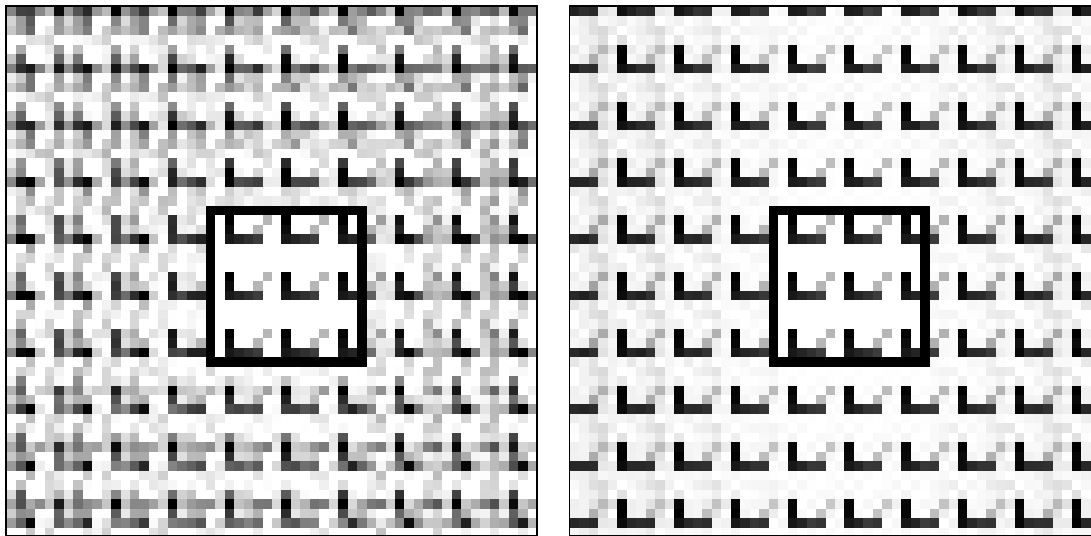


FIG. 3.8 - Influence de la taille sur le résultat de la déconvolution. L'image de droite représente la même partie utile que l'image de gauche avec un nombre de raies sélectionnées identiques mais après l'adaptation de la taille du rectangle extérieur.

3.3.4 Algorithmes hiérarchiques de déconvolution

Nous venons de voir comment améliorer la qualité de l'extrapolation par un ajustement de la taille du rectangle extérieur, mais il nous reste à examiner la question de la vitesse du traitement à partir de l'expression de G .

L'inconvénient responsable de l'insuccès de la plupart des algorithmes d'extrapolation, dont celui de FRANKE, est la lenteur de traitement. Alors plutôt que de chercher l'extrapolation du signal à pleine résolution, nous préférons réaliser l'extrapolation de signaux plus petits dérivés du signal original. La décomposition en sous-bandes offre une solution élégante au problème. Suivant son schéma, un signal de taille N est décomposé en

deux signaux de taille $N/2$, chacun limité en fréquences et capables de reconstruire à eux deux le signal de départ. Comme la réalisation par logiciel de la transformée de Fourier discrète est proportionnelle à $N \log N$, il apparaît clairement que l'extrapolation des signaux décomposés réduit considérablement le temps de calcul. En contrepartie surgissent des effets inhérents à l'échantillonnage de la fonction fenêtre et des problèmes de distribution d'énergie dans les bandes. Les algorithmes d'extrapolation intégrés à une structure en sous-bande sont principalement de deux sortes : soit ils agissent sur des bandes résultant d'une découpe parallèle, soit ils traitent les bandes d'une structure hiérarchique. Examinons ces deux possibilités.

1. *Déconvolution construite sur un schéma sous-bande travaillant à résolution fixe.*

En prenant G tel qu'exprimé par l'équation 3.5, tout le bénéfice simplificateur par rapport aux méthodes précitées est perdu. Mais si l'itération s'effectue, par exemple, par déconvolution sélective uniquement sur la partie centrale $M^{-1}SM$ de G à laquelle est adjointe un opérateur composé $\tilde{T}\tilde{D}$ similaire à TD , le gain en nombre de calculs est appréciable parce que l'image est réduite d'un facteur quatre.

L'algorithme se résume par un schéma en trois phases. Nous fournissons ci-après une formulation mathématique de l'algorithme, dans laquelle $u(n)$ et $v(n)$ sont des vecteurs intermédiaires, et où seule la partie centrale est itérative.

Étapes de l'algorithme :

$$\begin{aligned} u_0(n) &= [\downarrow 2]M^{-1}\Pi_1 My(n) \\ u_{i+1}(n) &= \tilde{T}\tilde{D}M^{-1}SMu_i(n) \\ v(n) &= TDM^{-1}\Pi_2 M[\uparrow 2]u(n) \end{aligned}$$

Le concept pivot du nouvel algorithme est celui de *décomposition sous-bande*. Par un choix adéquat d'un filtre sous-bandes de décomposition $[\downarrow 2]M^{-1}\Pi_1 M$ et d'un filtre de recomposition $M^{-1}\Pi_2 M[\uparrow 2]$ satisfaisant les conditions de reconstruction parfaite, tout le travail central de l'opérateur s'effectue à résolution réduite. En contrepartie, l'information contenue dans toutes les sous-bandes n'intervient pas dans le mécanisme d'extrapolation. De fait, un schéma de décomposition sous-bandes horizontal et vertical d'ordre 2 fournit quatre sous-bandes, contenant toutes des éléments d'interprétation du signal traité. Ces bandes permettent une reconstruction parfaite si les filtres sont bien choisis.

La figure 3.9 présente les étapes d'une décomposition en sous-bandes suivie de l'extrapolation de chacun des signaux intermédiaires.

Allocation des coefficients dans les bandes. La question centrale qui se pose est la détermination de la bande dont la contribution à l'extrapolation est la plus élevée. A priori, il est préférable de travailler d'abord avec la bande de basses fréquences horizontales et verticales car toute image accumule une partie considérable de l'énergie¹⁰ dans cette bande. Autrement dit, le nombre de coefficients à sélectionner dans cette bande sera plus élevé que le nombre de coefficients d'une autre bande. Dans ce cas,

10. L'énergie est évaluée par l'écart type du signal.

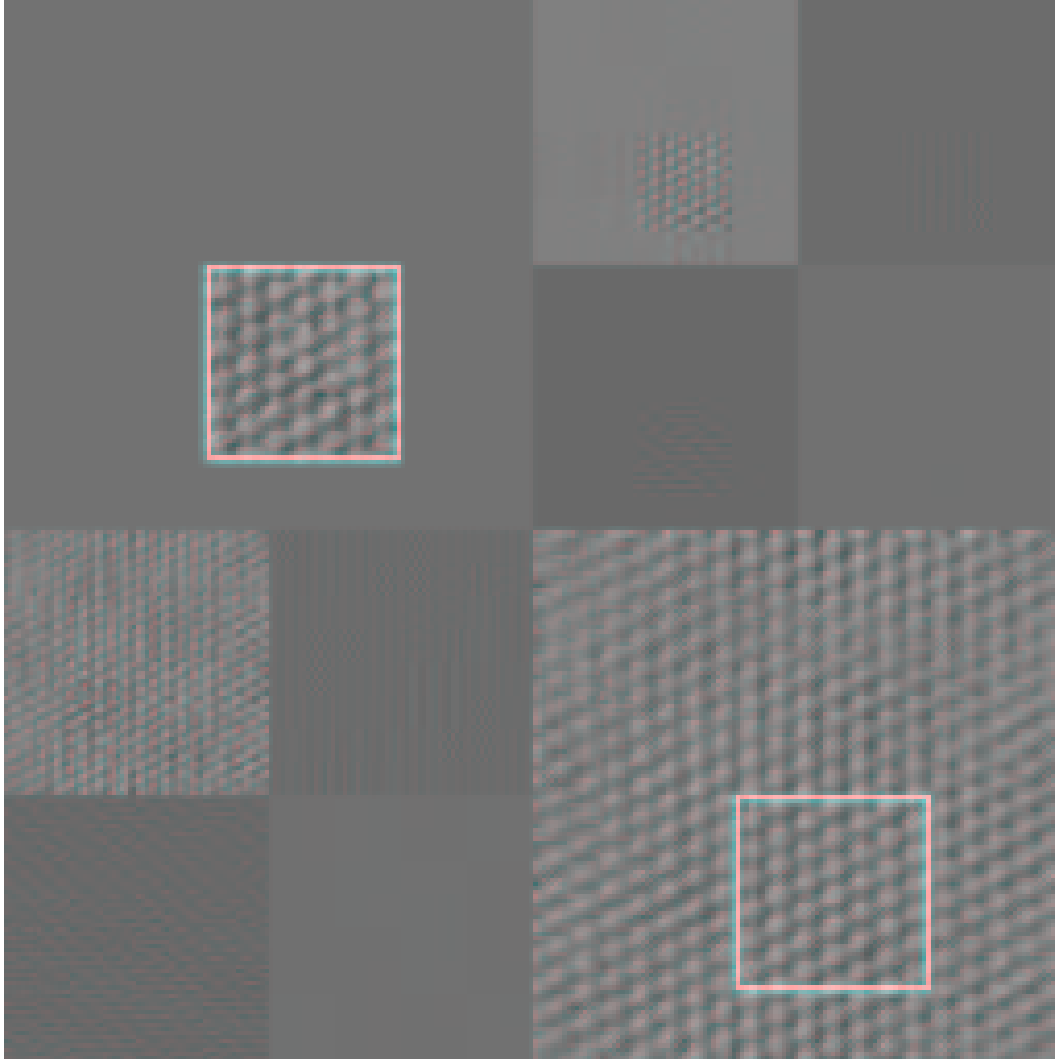


FIG. 3.9 - *Extrapolation de signaux issus d'une décomposition en sous-bandes. L'échantillon de texture est tout d'abord passé à travers un étage d'analyse en sous-bandes. Il en résulte quatre signaux au contenu fréquentiel distinct. Signalons que le signal contenant les basses fréquences est situé dans le coin supérieur gauche. Avant de revenir à la pleine résolution par l'étage de synthèse, les quatre signaux subissent une extrapolation.*

la distribution des coefficients (ou de l'énergie) sur l'ensemble des bandes suit une *allocation dynamique*. À l'autre extrême, on peut trouver une *distribution statique* des coefficients, en simple rapport avec la surface de la bande. Les meilleurs résultats s'obtiennent dans une situation intermédiaire, distribuant la moitié des coefficients suivant le schéma statique et l'autre moitié par la voie dynamique. La raison qui nous fait renoncer à un schéma purement dynamique est la disproportion des énergies contenues dans les bandes. Elle conduit à allouer trop peu de coefficients aux bandes à contenu en hautes fréquences, or ces fréquences caractérisent en l'occurrence les textures.

2. *Déconvolution construite sur un schéma sous-bande utilisant l'information de toutes les bandes par résolution progressive : approche hiérarchique.*

Nous venons de proposer un schéma qui décompose l'information spectrale par analyse en sous-bandes parallèles. Ce schéma souffre des variantes plus complexes les unes que les autres. Un amendement possible revient à poursuivre la décomposition d'une seule bande seulement, obtenant de la sorte un schéma hiérarchique. Une telle approche allie une allocation dynamique totale ou partielle des coefficients à une décomposition fonction du signal à traiter. L'objectif est de réduire la taille des bandes où sont sélectionnés de nombreux coefficients tout en conservant la résolution maximale des autres bandes. Le critère d'attribution des coefficients ou de décision de découpage inclut par exemple les aspects pondératifs de manière à atténuer l'influence des contours, information principalement contenue dans les hautes fréquences, ou considère un rapport entre l'énergie présente dans la bande et l'énergie du seul signal de forme contenue dans la même bande. Par ce procédé, l'influence de la forme de la fenêtre est prise en compte dans l'étape de sélection des coefficients.

À partir du moment où plusieurs bandes sont impliquées simultanément dans le processus itératif, cela signifie que l'opérateur $\Pi_2 M[\uparrow 2] M^{-1} S M[\downarrow 2] M^{-1} \Pi_1$ est organisé en une structure parallèle pour permettre d'utiliser toute l'information disponible. La parallélisation de l'opérateur n'affecte aucunement la convergence parce qu'elle résulte d'une sommation d'opérations toutes non extensives. Par contre, elle procure d'importants gains de temps de calcul comme l'illustreront les exemples de la section suivante. Concernant l'unicité de la solution, la condition suffisante d'*unicité stricte* proposée dans la démonstration de convergence reste d'application pour chaque bande.

3.3.5 Exemples d'utilisation de l'extrapolation

L'extrapolation de signaux de texture remplit plusieurs rôles : croissance de germes de texture, décorrélation, codage, ... Les résultats des essais s'organisent suivant ces axes. D'autres exemples seront proposés dans le chapitre consacré à la segmentation.

Croissance de germes

Au chapitre 4, nous verrons que l'opérateur de segmentation détecte les textures par croissance d'un germe de texture. L'intérêt de cette démarche tient au respect d'une

cohérence spectrale tout au long de la chaîne mais, qui plus est, elle correspond à une interprétation simple des textures.

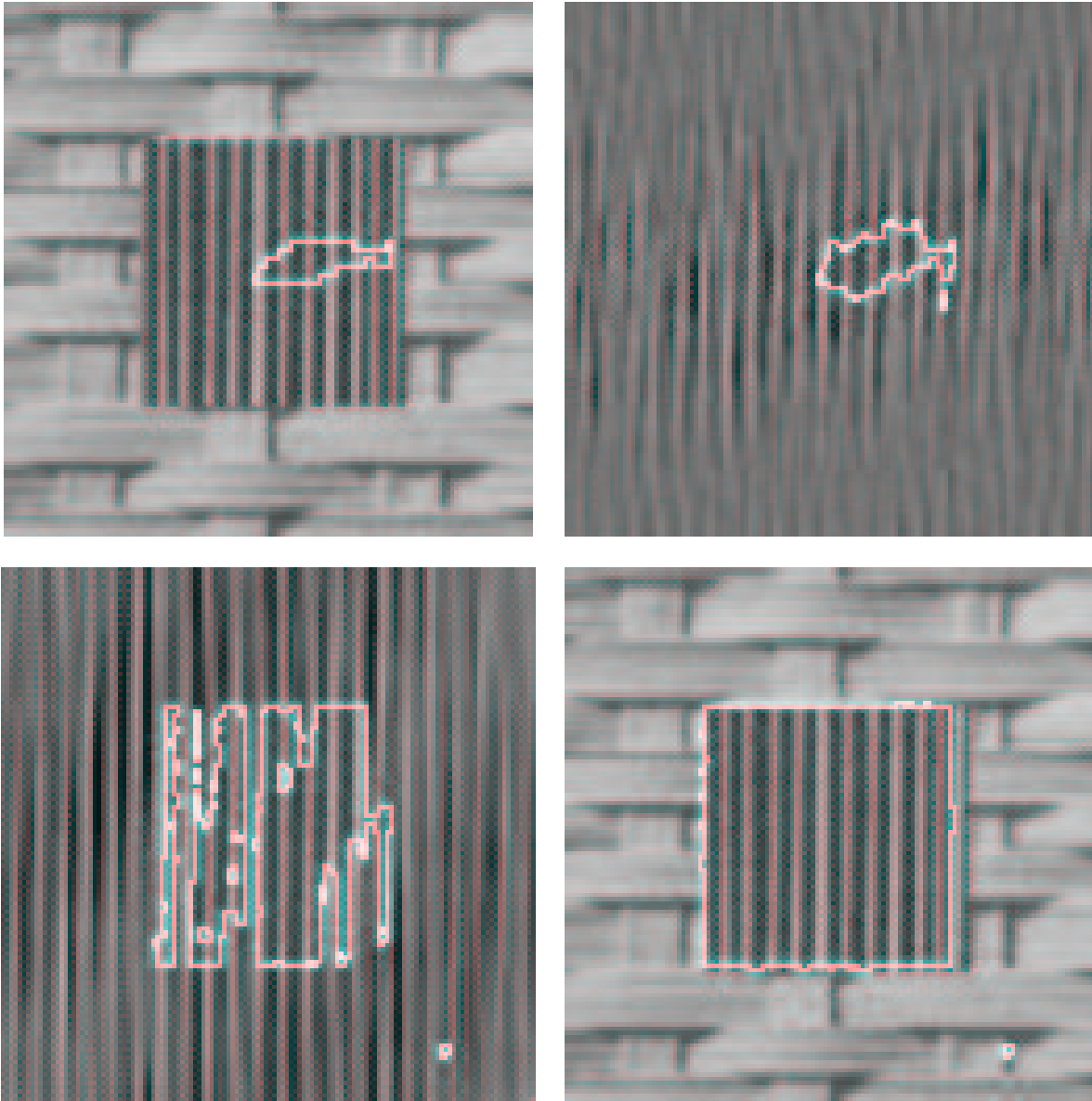


FIG. 3.10 - *Segmentation d'une texture par croissance d'un germe. Le processus de segmentation consiste à détecter un germe durant une phase d'initialisation, à l'extrapoler et à comparer les valeurs du signal extrapolé à celles du signal de départ.*

La figure 3.10 détaille une détection de texture par croissance de germes non connexes. Le procédé de segmentation consiste à détecter un germe de texture pendant une phase d'initialisation, à étendre ce signal par extrapolation et à comparer les valeurs extrapolées aux valeurs de l'image. S'il y a concordance des valeurs, on ajoute ces valeurs au germe et le processus recommence. Comme critère de comparaison entre les valeurs de l'image extrapolée et celles de l'image originale, nous avons pris, dans cet exemple, la moyenne

Nombre de coefficients	Sans extrapolation	Après extrapolation
50	10.5	7.0
100	9.2	5.5
200	7.7	3.8
300	6.7	3.0
400	6.1	2.2
500	5.5	1.8
1000	3.6	0.2

TAB. 3.1 - *Relevé de la valeur moyenne des erreurs absolues pour différents signaux obtenus en ne sélectionnant qu'un certain nombre de coefficients spectraux.*

locale sur une fenêtre 3x3. Le germe initial est entouré d'un trait dans la première image. Le résultat de l'extrapolation est surtout conforme au contenu de la texture à la périphérie du germe. C'est pourquoi le germe commence par grossir. La deuxième image reprend l'extrapolation d'un germe légèrement étendu. Étant donné que le germe s'étend en respectant la cohérence spectrale de la région, il incorpore des points qui renforcent les traits dominants de la texture. Cet effet est visible sur la troisième image : les germes croissent en identifiant les points qui respectent et accentuent la cohérence spectrale. La texture identifiée correspond exactement à notre souhait.

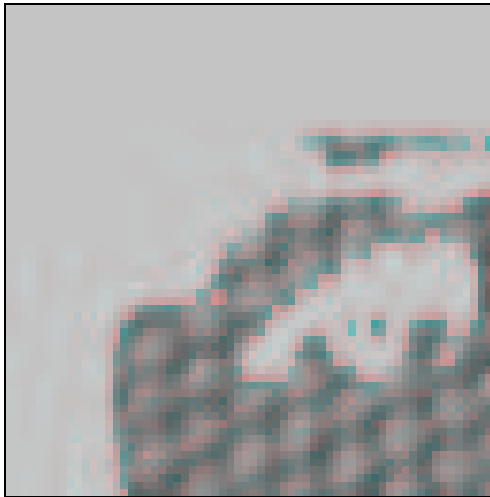
Décorrélation des informations relatives aux contours et aux textures

Pour une compression du signal de texture, il faut que la qualité du signal à l'intérieur de l'objet, peu importe l'extérieur, soit la meilleure avec un minimum de raies spectrales. La figure 3.11 compare l'approximation obtenue en ne gardant qu'un certain nombre de raies spectrales du spectre. Les images sont respectivement : le signal original de taille 64x64 connu sur un segment de forme assez quelconque, une extrapolation possible où toute information de contours a disparu, le signal approximé avec seulement 2% des raies spectrales et le même signal après extrapolation.

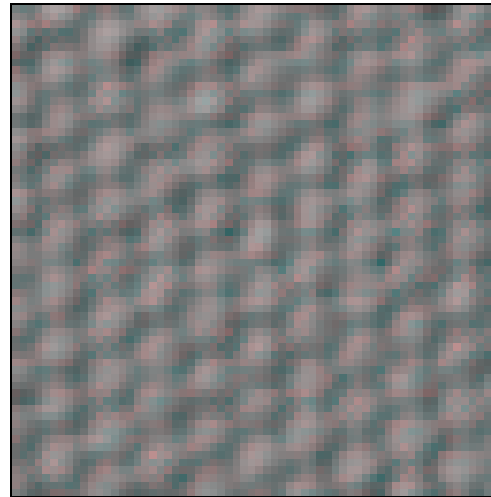
La dernière image représente mieux le signal de texture. Des valeurs numériques de la qualité d'approximation ont été rassemblées dans le tableau 3.1. Elles correspondent aux valeurs moyennes des erreurs absolues à l'intérieur de la fenêtre. Ces résultats confirment l'impression visuelle ; dans cet exemple, il faut à peu près 5 fois plus de coefficients pour approximer un signal non extrapolé de qualité comparable ! On arrive à des conclusions similaires en examinant le SNR ¹¹ (voir figure 3.13). Ces exemples montraient des textures régulières, pourtant la conclusion est tout à fait générale comme nous l'ont prouvé tous nos essais. Nous conjecturons donc que *la déconvolution sélective réduit l'entropie du spectre*.

Conjecture 21 *Quel que soit le contenu spectral d'une région (texture régulière, région active ou à variations lentes), l'approximation résultant d'une sélection d'un nombre fixe de raies spectrales est toujours meilleure après extrapolation.*

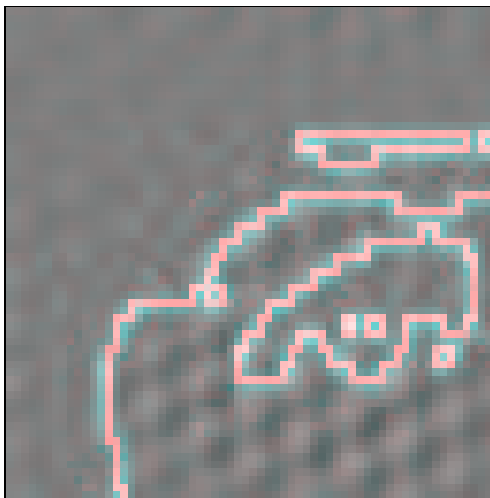
11. Signal to Noise Ratio



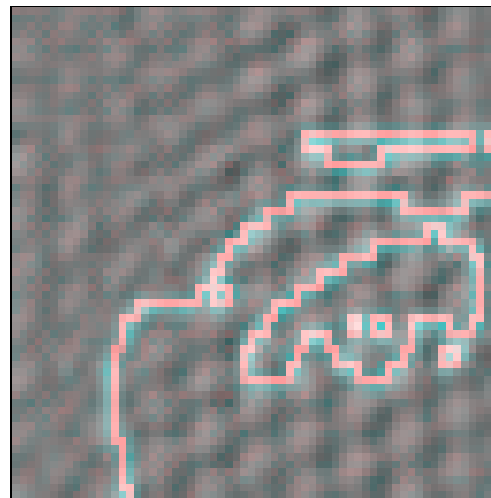
(a) Image originale.



(b) Une extrapolation complète.



(c) Approximation sans extrapolation préalable.



(d) Approximation avec un même nombre de raies spectrales que (c) mais après extrapolation.

FIG. 3.11 - Comparaison de l'approximation d'un signal sans ou avec extrapolation préalable (extrait de [10]).

Cet énoncé résume l'action de décorrélation des signaux de contour et de texture que nous souhaitions obtenir dans le procédé de codage complet.

Avant de terminer ce chapitre, nous parlerons encore de la qualité d'approximation dans un schéma en sous-bandes, des vitesses correspondantes, du codage des textures et d'une perspective de synthèse.

Vitesse et qualité de l'extrapolation. Les tableaux 3.2 et 3.3 reproduisent des résultats présentés à une conférence [105]. Les valeurs proviennent de l'extrapolation des composantes de représentations en sous-bandes hiérarchiques d'une image de synthèse. Les symboles ϵ_f , ϵ_{fc} et ϵ désignent les moyennes des erreurs absolues par rapport à l'original respectivement dans la fenêtre, sur le complémentaire et totale pour une image à 255 niveaux de gris. En suivant la diagonale orientée à -45° dans les tableaux, la comparaison indique une qualité similaire d'approximation pour un gain de temps de calcul d'un facteur 3. Si en plus de recourir à un schéma hiérarchique, nous avons la possibilité d'implanter l'algorithme, les calculs pourront être menés en parallèle. Le tableau 3.4 permet d'apprécier le gain potentiel énorme d'une réalisation matérielle. Il compare les temps de calcul requis pour une architecture séquentielle, comme celle d'un ordinateur classique, et les temps prévus pour une architecture parallèle, c'est-à-dire les temps de la colonne précédente divisés par le nombre de sous-bandes. Indiquons qu'il s'agit de signaux issus de décompositions parallèles avec allocation statique des coefficients dans les bandes.

Il n'y a aucune commune mesure entre les 520 secondes de calcul que demande l'algorithme initial de FRANKE et les 3 secondes d'un schéma parallèle : le gain atteint presque 200. En plus de l'économie réalisée sur le temps de calcul, le traitement s'effectue sur des signaux plus petits, donc plus faciles à mettre en œuvre.

3.4 Codage des textures

Pour décrire des *régions texturées* ou des *régions à variations lentes*, quelques composantes du spectre de Fourier suffisent, de l'ordre de quelques pour cent. Le codeur se trouve alors devant la difficile tâche de devoir coder un plan de valeurs, qu'il s'agisse d'une DFT ou d'une DCT, dont la plupart sont nulles. De plus, à notre connaissance, aucune étude n'examine sérieusement l'impact perceptif d'une mise à zéro de nombreux coefficients d'une transformée spectrale de taille quelconque.

Il n'entre pas dans nos intentions d'entreprendre l'examen complet d'une technique de codage des textures car cet exercice met en jeu un grand nombre de paramètres. Notre approche est très pragmatique : établir, à travers un exemple détaillé, une relation entre le nombre de raies spectrales, le pas de quantification et le taux de compression.

Signalons tout d'abord que nous proposons que les *régions actives* soient codées par transmission des coefficients transformés après découpage en blocs 4x4. Il est logique de prendre une taille 4x4 plutôt que 8x8 pour éviter quantité de blocs de régions actives incomplets, et de choisir la DCT parce qu'elle décorrèle mieux des échantillons quelconques d'une image. De plus, cela ne pénalise nullement le codage car, vu la faible corrélation entre blocs de régions actives, la fonction qui exprime le coût de codage par rapport à la taille des blocs présente un minimum très plat sur une large plage de tailles de bloc différentes ; cette conclusion est empruntée à MACQ [55]. Ainsi, après un découpage préalable en blocs

Nombres de raies conservées	Type de l'erreur	Nombre d'étages de décomposition				
		0	1	2	3	4
30	ϵ_f	10.67	14.82	15.91	17.08	17.04
	ϵ_{fc}	12.67	15.20	16.90	18.08	18.03
	ϵ	11.89	15.01	16.40	17.57	17.53
60	ϵ_f	2.02	8.33	9.04	9.85	9.87
	ϵ_{fc}	2.65	8.64	9.87	12.31	12.29
	ϵ	2.40	8.48	9.46	11.08	11.07
100	ϵ_f	0.5	2.13	2.63	3.70	3.63
	ϵ_{fc}	0.5	2.31	4.17	7.70	7.69
	ϵ	0.5	2.22	3.40	5.69	5.65
200	ϵ_f	0.5	1.06	1.50	2.52	2.40
	ϵ_{fc}	0.5	1.97	3.83	7.06	7.02
	ϵ	0.5	1.52	2.66	4.78	4.70
300	ϵ_f	0.5	1.11	1.50	2.46	2.52
	ϵ_{fc}	0.5	2.53	4.15	7.18	7.14
	ϵ	0.5	1.82	2.82	4.80	4.68
600	ϵ_f	0.5	1.13	1.55	2.44	2.28
	ϵ_{fc}	0.5	3.27	4.70	7.62	7.59
	ϵ	0.5	2.20	3.12	5.02	4.92
1000	ϵ_f	0.5	1.13	1.57	2.44	2.32
	ϵ_{fc}	0.5	3.62	5.00	7.88	7.84
	ϵ	0.5	2.37	3.28	5.15	5.06

TAB. 3.2 - Moyennes des erreurs absolues (extrait de [105]).

Nombre de raies spectrales conservées	Nombre d'étages de décomposition				
	0	1	2	3	4
30	184	38	33	32	32
60	264	62	44	44	44
100	408	91	63	60	63
200	802	164	109	104	106
300	1194	241	149	148	146
600	2372	457	285	276	273
1000	3556	678	415	402	395

TAB. 3.3 - Temps de calcul en secondes pour une déconvolution des bandes de schémas hiérarchiques (extrait de [105]).

de taille 4x4 et une éventuelle extrapolation destinée à compléter certains blocs, les régions actives seront codées par l'entremise de coefficients DCT, laquelle conduit à des taux de compression de l'ordre de 10.

Nombre d'étages de décomposition	Type d'architecture	
	Séquentielle	Parallèle
0	520	520
1	126	31.5
2	42	3

TAB. 3.4 - Comparaison des temps de calcul (en secondes) de deux architectures.

Suivant le double axe *substitut-complément des techniques de codage actuelles* mentionné dans le chapitre 1, nous nous intéressons ci-après à une technique de codage par coefficients transformés et à l'utilisation de l'extrapolation comme moyen de prédiction. Ces deux utilisations s'articulent sur le concept central de *facteur de sélection*.

Définition 22 *Le facteur de sélection est le nombre de composantes spectrales, partie réelle ou imaginaire dans le cas de coefficients complexes, non nulles et distinctes, utilisées pour décrire une texture.*

3.4.1 Codage par l'envoi de positions et d'amplitudes des coefficients transformés

Il nous reste encore à résoudre la question du codage des *régions texturées* et des *régions à variations lentes*. Puisque ces régions sont représentées par quelques raies dispersées dans le plan spectral, *il est préférable de coder uniquement les positions et les amplitudes*. Parfois, ces opérations sont confondues à l'intérieur même du codeur comme c'est le cas du codeur UVLC¹² développé par MACQ [14, 55]. Pour apprécier le coût de codage, nous préférons une approche analytique qui met en relief les principaux paramètres. Les voici passés en revue :

le type de transformée. Pour un modèle d'image markovien d'ordre un, il est connu que la DCT approxime mieux la KLT. Bon gré mal gré, la DCT introduit un effet de miroir qui détériore les signaux orientés autrement qu'à l'horizontale ou à la verticale. Conséquemment, on pourrait suggérer de conjuguer un codage par DFT pour les signaux orientés à 45° et 135°, et par DCT pour les signaux qui concentrent leur énergie plutôt le long de l'axe horizontal et vertical [50]. Il s'avère que cette distinction n'est pas indispensable pour des signaux texturés ou à variations lentes, étant donné le faible écart du coût de codage par DFT ou DCT, et de plus, il peut arriver qu'une texture suive une orientation marquée avec en filigrane une autre direction ; d'où notre décision de prendre la DFT. À titre illustratif, la texture de la figure 3.12 (image de taille 128x128) a été codée par DFT¹³. À sa droite est reproduite une approximation avec un facteur de sélection de 100, soit moins de 1% du contenu spectral du rectangle qui circonscrit la texture.

12. Universal Variable Length Coding

13. Nous rappelons au passage que la transformée de Fourier d'un signal réel est symétrique conjuguée.

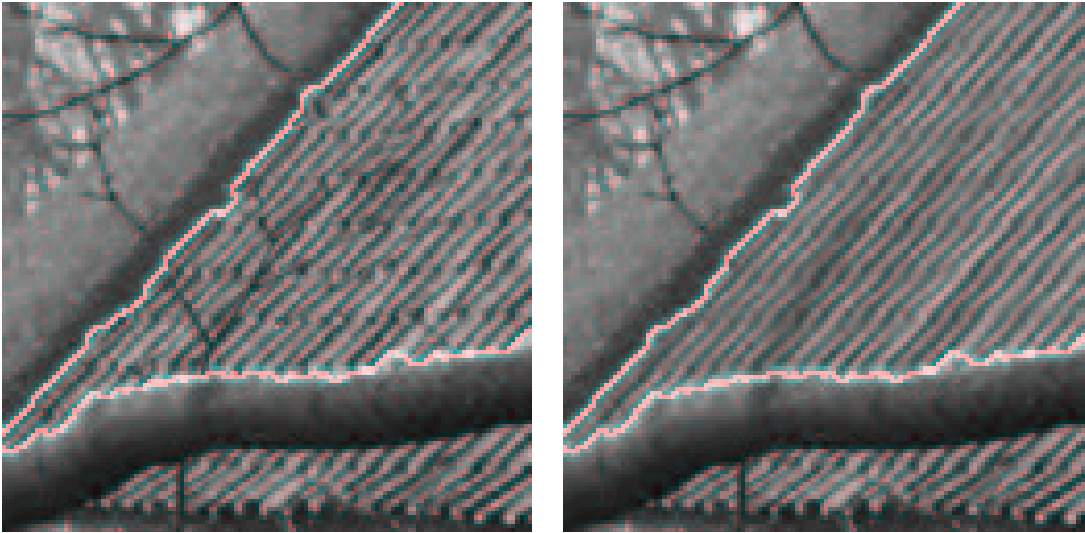


FIG. 3.12 - Une texture et son approximation avec un facteur de sélection de 100. Le taux de compression (sans contour) correspondant vaut 42. Le maximum de la fonction d'autocorrélation se situe au point (5,12).

le codage des positions. Le taux de compression dépend aussi de la surface du rectangle qui circonscrit la région, laquelle est liée à sa taille. Par ailleurs, il doit s'exprimer en *nombre de bits rapportés à la surface de la région* et non à la surface du rectangle, confusion qui a parfois été faite dans la littérature.

Tout au plus faut-il un bit par position pour dire si un coefficient non nul est placé en chaque point. Grâce à l'usage de codeurs performants, il y a moyen d'améliorer le codage puisque la probabilité d'un coefficient nul est nettement plus élevée que celle d'un coefficient non nul, c'est-à-dire que l'entropie de la source est inférieure à 1 bit. La plupart des codeurs entropiques (run-length, codeur arithmétique, ...) d'ordre 0 approchent la limite théorique, égale au produit du nombre de symboles par l'entropie d'un symbole, avec une efficacité de l'ordre de 0.97 [73, 99]. Après l'essai de plusieurs codeurs, nous avons choisi d'utiliser un *codeur arithmétique d'ordre 1*, dont le code en langage C a été fourni par NELSON [66]. Expérimentalement, le nombre de bits nécessaires au codage des positions suit une loi linéaire. Autrement dit, le coût du codage est directement proportionnel au facteur de sélection, avec toutefois une pente inférieure à celle d'un codeur arithmétique d'ordre 0.

le codage des amplitudes. Quelle que soit la transformée utilisée, les variables aléatoires qui modélisent les valeurs des coefficients n'obéissent pas à une loi unique, surtout pour des blocs aussi grands. En l'absence d'un modèle, il est logique de coder chaque coefficient avec un nombre fixe de bits et d'appliquer une technique de la théorie de l'information au flot de bits résultant. Cette manière d'agir soulève la double question de la quantification et du choix du codeur.

Au départ, les coefficients représentés par des réels occupent 32 voire 64 bits. L'action utilisée pour réduire le nombre de bits requis s'appelle la *quantification*. Elle n'est autre que le processus d'approximation qui mène à une perte de précision sous la forme d'une fonction en escalier dans l'espace *valeurs spectrales – valeurs quantifiées*,

et dont l'allure dépend du délicat compromis compression-distorsion. En traitement d'image, la *matrice de quantification* est l'ensemble des *pas de quantification* associés à chaque coefficient, le pas étant la hauteur d'une marche d'escalier. Si le choix d'une matrice fait souvent l'objet de vives discussions, tant il y a de critères d'optimisation, les propositions s'accordent pour octroyer une précision plus importante aux coefficients proches de l'origine. Cette conclusion ne tient plus dans notre situation en raison d'une taille de bloc nettement supérieure et de la caractérisation des textures par des hautes fréquences. Ainsi, à défaut d'une analyse en profondeur, nous avons opté pour une quantification uniforme ; sans doute pourrions-nous utiliser une pondération des coefficients. Pratiquement, la gamme dynamique du coefficient DC est tout d'abord ramenée au quadruple de la plage des valeurs de la fonction de luminance. Par exemple, si $f(n) \in [0, 255]$ alors $\mathcal{F}(0) \in [0, 1020]$. Sauf de très rares exceptions, cela garantit que les autres valeurs de la transformée restent dans $[-127, 128]$. En guise de référence, nous considérons qu'un pas de quantification $q = 1$ revient à arrondir à l'entier le plus proche.

Pour étayer la discussion du codage des textures, nous avons rapporté les résultats de plusieurs essais sous forme de graphiques à la figure 3.13. Les courbes comparent les taux de compression (sans l'information des contours!) et SNR de deux séries de codage de la texture présentée à la figure précédente, à pas de quantification différent, après une éventuelle extrapolation ou ajustement de la taille. La comparaison appelle plusieurs remarques :

- l'ajustement de la taille par recherche du maximum de la fonction d'autocorrélation ne sert à rien pour le codage. À faible ou forte quantification et dès réalisation d'un processus d'extrapolation, les courbes se superposent, ce qui ne signifie nullement son inutilité ! En effet, si le contenu de la région est peu influencé par un ajustement de la taille, la qualité de l'extrapolation varie du tout au tout. Il suffit de revoir la figure 3.8, page 58, pour s'en convaincre. Toutefois, dans le cadre strict du codage de textures, *il ne s'avère pas utile de modifier la taille de l'échantillon*. Cette constatation autorise, par extension, à choisir une taille proche de celle du rectangle qui circonscrit la région et plus commode pour les calculs de transformée rapide.
- à l'évidence, *qualité et taux de compression sont inversement proportionnels*. En outre, la valeur du pas de quantification joue un rôle déterminant. Si q vaut 1, il est presque indifférent, du point de vue du taux de compression, de coder le signal extrapolé ou non, constatation contredite pour $q = 10$. Par contre, la qualité de l'extrapolation exprimée par le SNR, quoique nous soyons conscient de sa piètre signification physique, varie fortement pour $q = 1$ comme pour $q = 10$. *À taux de compression et pas de quantification identiques, l'extrapolation améliore toujours la qualité du signal codé*.
- par le passé, le problème de la quantification a suscité de nombreuses réflexions, aboutissant pour la plupart à des schémas d'optimisation. Il y a fort à parier qu'une stratégie de quantification optimale permette d'obtenir des meilleurs compromis entre le taux de compression et la qualité. Ainsi, pour des facteurs de sélection inférieurs à 50, les courbes spécifiées par les symboles $\diamond$ et $\times$ se rejoignent dans le graphe du SNR

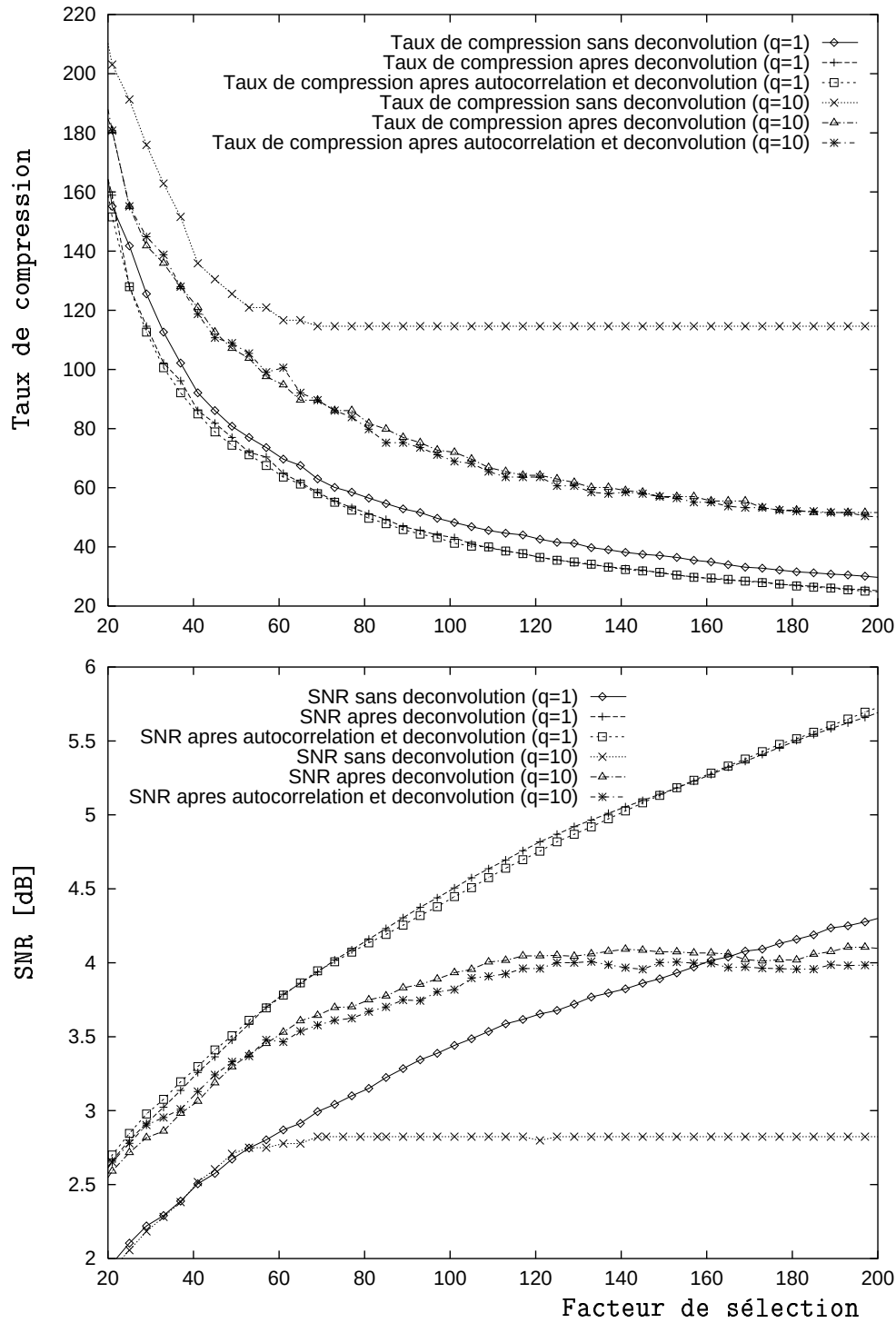


FIG. 3.13 - Évolution du taux de compression (sans contour) et du SNR en fonction du facteur de sélection.

alors que les taux de compression varient de 40%. Cette conclusion s'étend aux autres courbes. *Une stratégie optimale consiste donc à prendre un pas de quantification élevé lorsque la région est codée par quelques raies spectrales seulement.*

3.4.2 Codage par prédiction et DCT

La DFT, à la différence de la DCT, permet une bonne représentation des textures dans toutes les directions, bien qu'elle décorrèle moins les signaux quelconques. Malheureusement, son usage nécessite de traiter des nombres complexes. Pour ces raisons, on utilise plus souvent la DCT, avec son inévitable effet de miroir. Imaginons qu'il faille coder une texture périodique. Quel sens y a-t-il à lui appliquer un découpage en blocs 8x8 suivi du codage des coefficients DCT correspondants? Pour peu que la période d'échantillonnage diffère de la taille du bloc, la liaison entre blocs DCT voisins disparaît. La figure 3.14 illustre ces propos; l'image du haut représente deux blocs 8x8 voisins. Les valeurs des coefficients DCT sont dessinées en dessous. Elles ne présentent aucune ressemblance alors que dans le plan spatial le second bloc prolonge naturellement la texture du premier!

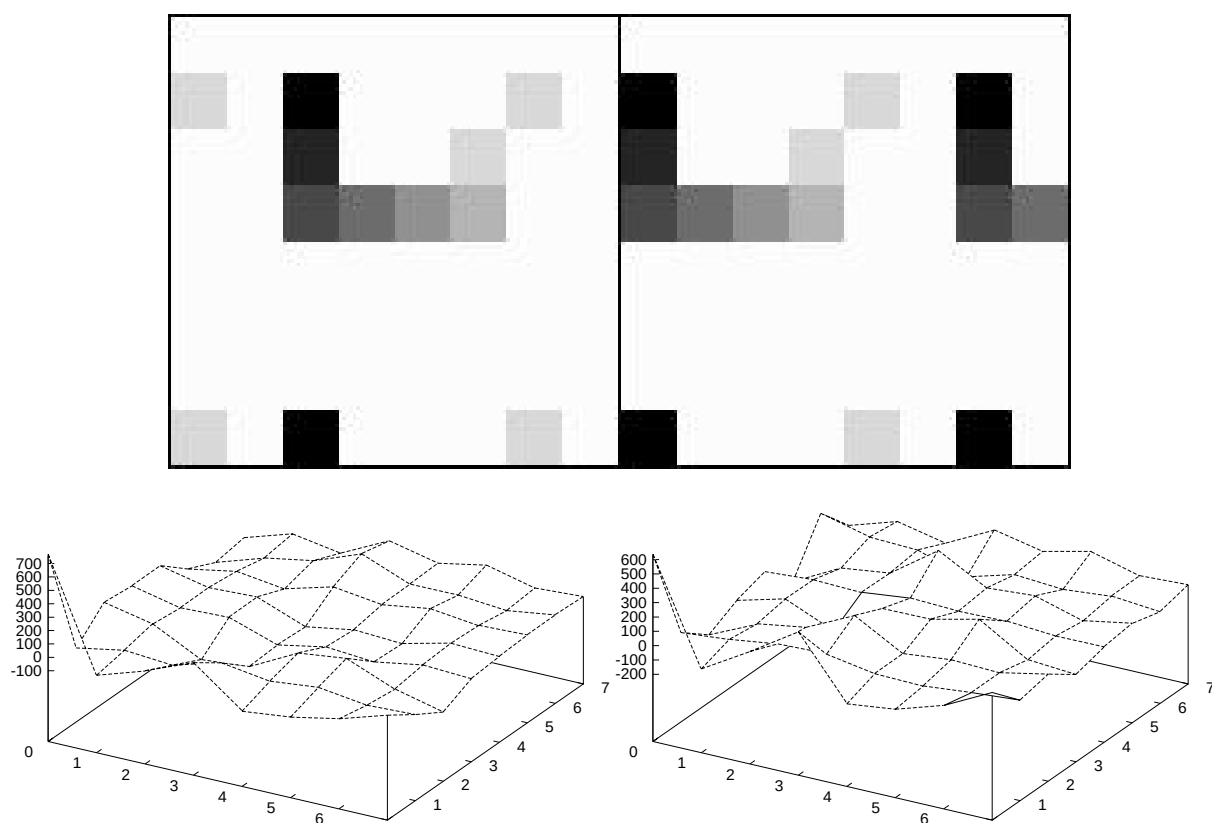


FIG. 3.14 - *Effet de la phase dans une DCT. La première image représente les deux blocs 8x8 voisins d'une texture. En dessous, nous avons reproduit le profil des coefficients DCT correspondants.*

Prédiction des blocs durant le balayage de l'image. *Pour déjouer la perte de cohérence entre blocs DCT voisins, nous avons mis en œuvre une technique de prédiction*

des blocs à venir sur base de l'extrapolation des échantillons déjà transmis. L'image de droite de la figure 3.15 montre la différence, centrée autour de 128, entre l'image de gauche et la prédiction de chaque bloc par extrapolation des blocs précédents dans un balayage horizontal partant du coin supérieur gauche. Le premier bloc est codé tel quel. En balayant la première bande, l'algorithme évalue rapidement la période horizontale puisque la différence observée est proche de 0. Au moment d'arriver au premier bloc de la seconde bande, l'algorithme ne connaît pas encore la période verticale mais, dès le bloc voisin, la prédiction est suffisante pour arriver à un faible signal d'erreur.

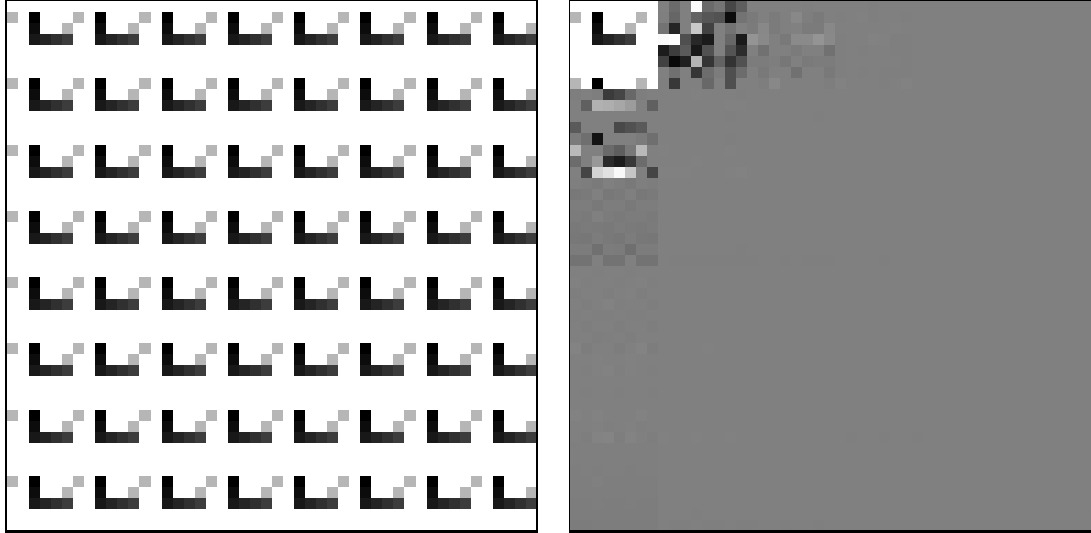


FIG. 3.15 - Codage d'une texture par prédiction des blocs 8×8 à venir.

Afin d'apprécier la pertinence de la démarche, nous avons mis les deux signaux de la figure 3.15 à l'entrée de l'UVLC en veillant à adopter un pas de quantification uniforme qui conduit à une reconstruction parfaite des signaux. Le résultat parle de lui-même : il faut à peine 25% du nombre de bits du codage de l'image de gauche pour représenter celle de droite. Le même procédé a été appliqué à une partie rectangulaire complète de la texture de la figure 3.12, filtrée au préalable par sélection d'un certain nombre de raies spectrales. Suivant ce nombre, la proportion passe de 75% à 85%. On pourrait croire à tort que ce procédé ne convient qu'aux textures dans un schéma orienté par la notion d'objet. Certes l'extrapolation exploite la périodicité des signaux ; néanmoins, il est possible de l'utiliser à l'intérieur de l'algorithme JPEG, en indiquant par un bit si chaque bloc appartient à la même texture que le bloc précédent. Si ce bit vaut 1, l'extrapolation des blocs connectés au bloc à coder et pour lesquels ce bit valait aussi 1 sert de prédiction, sinon le codeur l'ignore. Ce simple découpage permettra le codage efficace de régions homogènes, caractérisées par quelques raies spectrales dominantes. Il s'agit bien d'un complément à une technique de codage actuelle.

Conclusions du chapitre

Ce chapitre a développé tous les aspects associés à un algorithme itératif général de déconvolution. Nous avons d'abord montré qu'il s'agit d'une formulation suffisamment

générale pour recouvrir les algorithmes existants tout en proposant de nouveaux schémas. La question de convergence et d'unicité est examinée par le biais d'un théorème et traitée dans le cas pratique de la déconvolution sélective. Ensuite, nous avons proposé d'améliorer la qualité de l'extrapolation en adaptant la taille du rectangle extérieur suite à une analyse de la fonction d'autocorrélation, et nous avons décrit de nouveaux algorithmes de déconvolution spectrale fondés sur la déconvolution sélective et la décomposition d'un signal en sous-bandes. Il ressort des expériences que les objectifs fixés, à savoir la croissance de germes de texture pour l'identification de textures lors de la segmentation, la suppression de l'information de contours dans le spectre d'une texture et la réduction du temps de calcul, sont tous remplis.

Quant au codage, un exemple détaillé met en valeur la puissance de la méthode. Il apparaît que le coût de codage est proportionnel au nombre de raies spectrales conservées pour décrire la région. De plus, nous avons constaté qu'à des fins de codage, la correction de la taille de l'échantillon n'apporte rien alors que l'extrapolation produit des améliorations perceptibles du rapport signal à bruit à taux de compression fixé. Nous avons également montré que l'extrapolation du signal peut servir de prédiction.

Perspectives

La phase d'expérimentation a amené une nouvelle idée : pourquoi ne pas se contenter de transmettre un échantillon de texture, de forme rectangulaire, et *synthétiser un signal de texture complet du côté du récepteur*? Bien que cela suppose que le récepteur soit capable d'extrapoler un signal, en procédant de la sorte, le signal envoyé ne contient plus aucune information de contour et, de toute façon, l'émetteur peut transmettre un complément de précision pour le signal extérieur à l'échantillon si cela s'avère nécessaire.

La figure 3.16 montre le résultat de l'extrapolation de deux échantillons de taille différente d'une même texture.

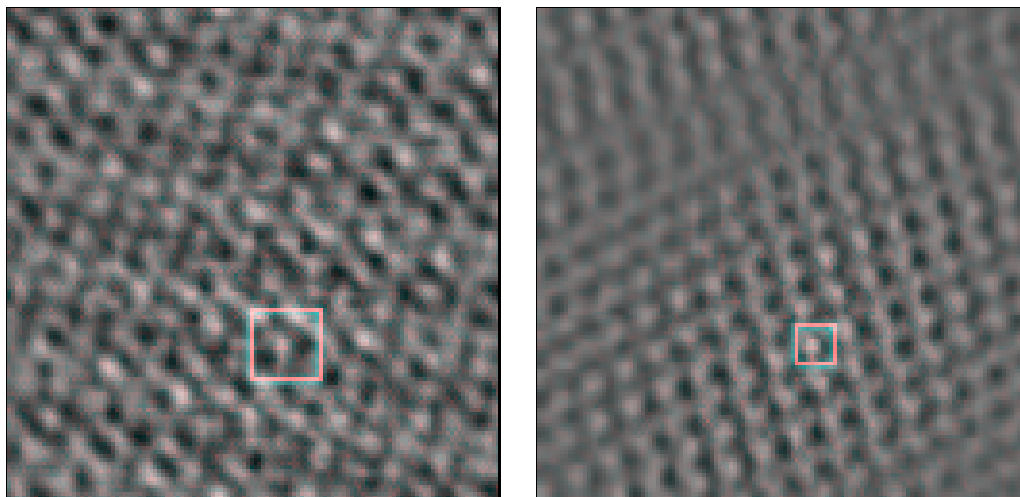


FIG. 3.16 - Deux exemples de synthèse d'une texture à partir d'un échantillon. Dans cette comparaison, l'échantillon de droite est une partie de celui de gauche (tous deux étant inclus dans le morceau de texture de la figure 3.17). On voit donc clairement que la taille de l'échantillon détermine l'aspect de la texture extrapolée.

Dans cet exemple, il apparaît clairement que la détermination de la taille et de la position de l'échantillon conditionne le résultat. L'efficacité de cette idée dépend de l'aptitude à détecter de tels échantillons, à moins qu'on décide de prendre systématiquement le plus grand rectangle inscrit dans la texture.

Le procédé est plus général si l'on définit une famille de formes simples. La figure 3.17 fournit, à droite, un exemple de synthèse de texture à partir d'un échantillon connu dans un disque : le signal de l'intérieur du disque ne présente aucune cohérence spectrale ce qui n'empêche pas que le signal extrapolé convienne pour remplir toute l'image.

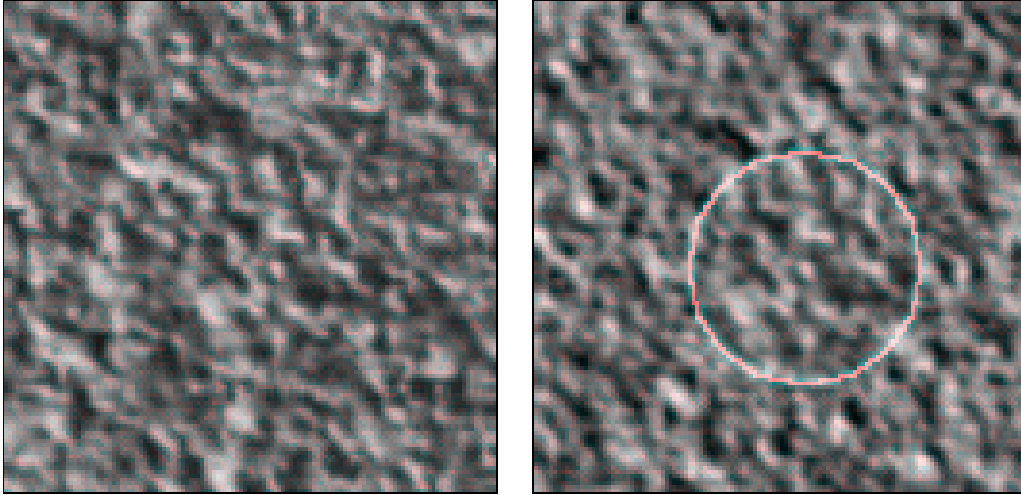


FIG. 3.17 - *Synthèse d'une texture aléatoire par extrapolation d'un échantillon. Le signal de gauche représente la texture originale.*

Chapitre 4

Segmentation d'image

Introduction

Dans les approches classiques de compression ou de rehaussement, la sortie est une approximation ou une amélioration de l'image d'entrée. Une autre branche du traitement d'image s'occupe de l'*analyse d'image* ou *analyse de scène*. Dans ce domaine, l'entrée est toujours une image mais la sortie est une *description* de l'image. La plupart des descriptions nécessitent une *détection* préalable des formes présentes dans l'image, étape appelée *segmentation*. Ce chapitre est consacré à l'*étude d'une technique de segmentation d'image axée sur le concept d'objet*.

Pour situer le problème de la segmentation d'image, nous étudierons une série d'outils de segmentation en précisant leurs avantages et inconvénients dans la section 4.1. Le schéma de segmentation que nous présentons s'inspire des techniques existantes et les complète pour obtenir une segmentation plus universelle, c'est-à-dire applicable à plus d'images.

Comme l'accent est mis sur le codage de ces objets, rien n'impose que les objets détectés correspondent à des objets réels. Et d'ailleurs, comment pourrions-nous remplacer l'expertise d'un être humain ? Dans ce chapitre, nous proposons un *modèle d'image* simple, adapté à la segmentation et au codage, et qui repose sur la notion de texture où il faut comprendre ce terme comme une organisation spatiale homogène.

La suite du chapitre, sections 4.2, 4.3 et 4.4, détaille les trois étapes du processus de segmentation. Elles visent toutes à extraire un type de régions appartenant au modèle. Leur réalisation fait intervenir l'extrapolation de texture, thème du chapitre précédent, quelques outils de filtrage linéaire et morphologique, ainsi que la ligne de partage des eaux.

En procédant par étapes successives, nous espérons allier les avantages de plusieurs procédés de segmentation et étendre le domaine des applications. Quelques résultats, non exhaustifs, confirment l'intérêt de la méthode.

4.1 Présentation du problème

Formellement, la segmentation s'apparente à un opérateur $\mathcal{E}$ agissant sur l'image I et fournissant, par exemple, une image binaire $\mathcal{E}(I)$ qui différencie les points des contours des objets.

Notre propos est de tirer, de cette étape de segmentation, la représentation de forme $\rho(X)$ et la représentation de texture $\rho(T)$ adjointe.

Définition 23 *L'obtention des objets contenus dans une image I se réalise par application d'un opérateur noté $\mathcal{E}(I)$ et appelé opérateur de segmentation. Après application de l'opérateur, nous disposerons des représentations de forme $\rho(X)$ et de texture $\rho(T)$.*

De fait, l'adoption d'une démarche cohérente de codage par objets impose la caractérisation de tous les objets dès l'étape de détection des objets.

4.1.1 Synthèse de la littérature consacrée à la segmentation

La plupart des techniques de segmentation reposent sur la réalisation quantitative d'une propriété particulière. Il s'agit par exemple d'extraire toutes les régions dont le niveau de luminance moyen dépasse telle valeur ou de déceler des cellules ayant une allure prédéterminée.

Le processus de segmentation d'images naturelles ne peut se contenter d'une description de régions aussi sommaire, raison pour laquelle la majorité des techniques de segmentation échouent complètement à produire une segmentation applicable à une large gamme d'images. En réponse, les auteurs ont imaginé des modifications allant de l'approche à plusieurs niveaux de résolution à l'accroissement du jeu de propriétés des régions, en passant par une complexification excessive des procédés primaires. Ainsi, certaines opérations de segmentation s'appliquent directement à toute l'image, d'autres nécessitent des images préalablement partitionnées ou des marqueurs, etc.

Les procédés actuels de segmentation appartiennent à plusieurs grandes familles que nous parcourons ci-après, en indiquant quels éléments interviendront dans notre schéma :

1. *Segmentation par détection de transitions dans le profil de luminance.*

Les techniques de segmentation les plus répandues s'obtiennent par *détection de contours*. L'idée est simple : pour repérer des objets dans une image, il suffit de détecter leurs frontières après avoir accentué les contours des objets par un filtrage passe-haut, tel un gradient morphologique ou linéaire.

La simplicité des techniques de détection de contours pousserait à les adopter pour rechercher les objets. Cependant, trop de situations réelles mettent en défaut les beaux principes de détection de contours. Remarquons tout d'abord qu'un seuil intervient lors de l'extraction des contours. Le seuillage est une opération délicate pour deux raisons : 1° la détermination du seuil doit s'adapter au contenu de l'image, 2° les contours des régions ne seront pas nécessairement fermés. Outre leur inadéquation pour certaines images, les détecteurs de contours présentent une sensibilité excessive au bruit car un filtrage passe-haut amplifie le bruit.

Dans un système de segmentation hybride tel celui que nous avons développé, l'emploi des techniques de détection de contours ne s'envisagera qu'en aval de l'outil de segmentation et dans une situation peu sensible à la valeur du seuil d'extraction.

2. Identification de régions par étude statistique d'ensembles de points.

Lorsqu'une simple détection échoue, il est tentant de donner une certaine cohérence statistique au procédé de segmentation. Une voie d'amélioration des performances consiste à traiter la segmentation comme un problème de décision dans lequel les statistiques des pixels se calculent par mélange de statistiques de régions composantes. De telles démarches de segmentation procèdent d'abord à l'évaluation de paramètres significatifs (moyenne, variance, ...) de différentes régions, puis elles estiment les mêmes paramètres pour des combinaisons de régions et, enfin, effectuent des modifications de structure de régions sur base d'un critère de décision théorique (cf. par exemple LACROIX [45]). Ces méthodes sont abondamment étudiées dans le domaine de la *reconnaissance de formes* (DUBUISSON [18], DUDA *et al.* [19], FU [26]). Sans aucune modification, elles ne produisent pas une segmentation pertinente. Dans la pratique, elles sont doublées d'arguments et de procédés empiriques.

Le reproche à adresser à ces techniques concerne l'inaptitude à inclure la structure spatiale au moyen de paramètres usuels. Ainsi, un procédé de segmentation construit par évaluation de paramètres de l'histogramme des niveaux de gris est totalement insensible à une rotation de l'image, alors que la rotation de certains objets modifie l'interprétation de la scène. Suite à l'importance de l'organisation spatiale des objets, les auteurs ont proposé des critères tenant compte de la disposition des objets. L'article de HSIAO et SAWCHUK [35] détaille à merveille une segmentation usant d'outils statistiques et incluant l'agencement spatial des régions.

Les démarches statistiques de segmentation se ressemblent en plusieurs points¹ :

- Exploitation de toute la connaissance accessible.

L'information concerne éventuellement l'aspect des régions, la disposition des régions, ... Il y a malheureusement fort peu de chance de disposer de toutes les données nécessaires au modèle. C'est donc une caractéristique essentielle des problèmes de décision en segmentation de devoir raisonner avec une connaissance a priori imparfaite et de la compléter par une analyse des signaux étudiés ; le système de décision doit être adaptatif.

- Synthèse d'un schéma de représentation.

Le processus de segmentation s'étudie grâce à l'introduction d'un schéma de représentation qui fixe la qualité, les performances et les limitations. De nombreux auteurs négligent de mentionner le modèle d'image sous-jacent. Or, la validité des processus de segmentation n'est satisfaite qu'à l'intérieur d'un jeu d'hypothèses bien précis.

- Ajout de critères correctifs.

Pour corriger les manquements d'un modèle global, plusieurs critères ont été étudiés. Il s'agit souvent de critères purement géométriques dont le but est d'organiser l'information manipulée au sein du modèle mathématique.

1. On peut trouver des exemples typiques de cette démarche dans PAL *et al.* [67], ou SUK *et al.* [97].

Nous verrons plus loin que la détection des régions actives s'effectue suivant les principes d'une approche statistique.

3. *Description d'image en termes d'éléments appartenant à diverses résolutions.*

Les analyses et descriptions par multirésolution (transformée en ondelettes, décomposition sous-bandes, décomposition pyramidale, ...) partent d'une idée simple, vérifiée par l'expérience, selon laquelle les détails d'une image, obtenus par filtrage, se remarquent à différents niveaux de résolution. Ce type d'approche est monnaie courante en codage d'image parce qu'elle permet une analyse hiérarchique, une transmission progressive et un changement de format.

En dehors des questions de vitesse, il nous semble inutile de recourir à une approche multirésolution car le plus haut niveau de résolution contient toute l'information. Le procédé de segmentation développé dans le cadre du projet Morpheco fonctionne sur base d'une structure hiérarchique morphologique. Comme on pouvait le craindre, le filtrage lié à la représentation introduit des effets indésirables dans le traitement d'images complexes.

4. *Sur-segmentation avec croissance de régions contrôlée.*

Pour compléter la revue des principales techniques de segmentation, il reste à décrire les *méthodes adaptatives de segmentation par découpage spatial*.

La plus ancienne est certainement la décomposition par *arbre quaternaire* ou '*quad-tree*'. Le procédé de segmentation basé sur cette représentation revient à découper progressivement l'image. Au départ, l'image forme une seule région. Si le critère d'homogénéité choisi n'est pas satisfait pour la région, alors elle est découpée en quatre parties égales. Puis, les régions nouvellement formées sont testées à leur tour. Ainsi, au terme de la décomposition, l'image se présente comme une collection de régions aux caractéristiques homogènes. On reprochera le côté arbitraire de la découpe géométrique, car une région qui chevauche deux zones d'un arbre quaternaire sera coupée en deux. Les résultats encourageants ont incité les chercheurs à améliorer le procédé et, si la décomposition n'est plus utilisée telle quelle dans les procédés de segmentation, elle sous-tend encore de nombreuses techniques de codage [96].

Les techniques inspirées de la décomposition quaternaire sont nombreuses et variées : *segmentation par croissance de régions*, partitionnement '*top-down*' ou '*bottom-up*' parfois appelé '*split and merge*', ... Elles s'articulent sur quelques grands axes :

- Choix d'un critère de cohésion.

Son évaluation détermine le découpage spatial.

- Organe de décision.

Le découpage en régions ou le regroupement s'effectue par le biais d'un organe de décision après évaluation du critère d'homogénéité. Souvent, il dépend de la taille de l'image ou de la région traitée.

- Sur-segmentation de l'image.

Par une sur-segmentation de l'image, on acquiert la certitude de n'oublier aucun contour. En contrepartie, la sur-segmentation introduit des artefacts et la présence de faux contours à éliminer par un procédé quelconque. Quand il s'agit

de codage, l'élimination des faux contours se double d'un *lissage des contours*, sujet du chapitre 6, afin de diminuer le coût de codage des contours.

Nous appliquerons également une sur-segmentation avec fusion de régions dans notre schéma.

Malgré toutes les tentatives entreprises, il n'existe malheureusement pas de procédé suffisamment général au délicat problème de la segmentation. Le chassé-croisé entre la segmentation d'image et son interprétation témoigne d'ailleurs de l'incertitude liée à la détermination des frontières des objets.

4.1.2 Vers un modèle d'image

Toute image numérique se présente sous la forme d'une grille de pixels. Traiter correctement ces données nécessite un modèle qui met en relief les caractéristiques des régions. Le modèle est l'outil analytique qui permet de lier un pixel à ses voisins ; il guide l'étape de segmentation, régit l'expression de la représentation de forme et de texture initiales et conduit aux détails de réalisation du codeur.

Les algorithmes de segmentation en vue d'un codage que nous connaissons, négocient d'une même manière toutes les parties de l'image, quel que soit leur contenu énergétique. Du point de vue de la théorie des variables aléatoires, cela suppose que l'image est la réalisation d'une source faiblement stationnaire. Rien ne justifie cette démarche car une image se compose de régions hétérogènes. Pour obtenir une meilleure réduction de débit binaire, il est préférable de réaliser une liaison parfaite entre toutes les étapes, de la segmentation jusqu'au codage, en adaptant le codage d'une région à son contenu. Ceci nous amène à proposer un modèle d'image basé sur des critères spectraux.

Définition d'un modèle d'image pour la segmentation

En substance, notre modèle d'image considère que *toute image ne contient que trois types de régions caractérisées par leur contenu spectral local* (voir la figure 4.1) :

1. les régions à variations lentes.

Le spectre de ces régions accumule la majeure partie de son énergie dans la zone des basses fréquences. Il n'est apparemment pas trop difficile de segmenter ces régions. Seule la présence de dégradés, parties d'image qui ne présentent aucune transition forte, complique la tâche ;

2. les régions à particularités spectrales.

Dans ces régions, quelques fréquences spécifiques impriment une homogénéité de direction ou d'aspect. À partir d'ici, nous réserverons la dénomination de *textures*² à ces régions ;

3. les régions actives.

Aucune particularité ne se dégage dans le spectre de ces régions, ce qui complique leur

2. Le terme *texture* sera parfois employé comme synonyme de l'aspect des régions *T*. Au cas où une confusion serait possible, nous préciserons le sens accordé.

codage. Outre la réelle difficulté de les segmenter, ces régions contiennent tellement de contours que leur considération introduirait un surcoût de codage important. Nous avons préféré réunir ces régions par composantes connexes.

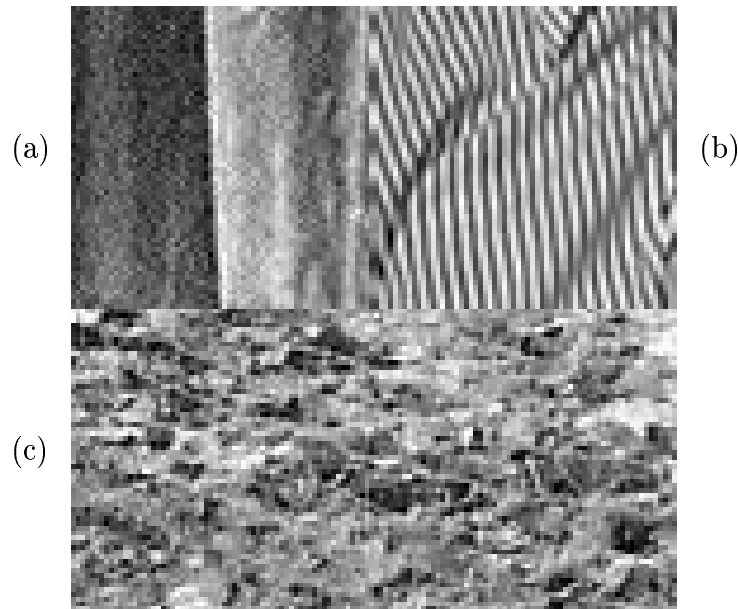


FIG. 4.1 - *Les trois types de régions du modèle d'image : (a) régions à variations lentes, (b) texture et (c) régions actives*

In fine, la segmentation produira ces trois types de régions qui, lors de traitements ultérieurs, subiront un codage adapté à leur nature spectrale.

Une segmentation à des fins de codage? Combiner des techniques de codage et de segmentation est indispensable en vertu de la diversité des régions qui composent une image naturelle. Ainsi, nous avons lié le nombre de raies spectrales sélectionnées pour décrire une région texturée à la surface et à la variance de cette dernière, alors que ce nombre dépend uniquement de la surface s'il s'agit d'une région à variations lentes, les régions actives étant confiées à un codeur DCT. De plus, aucun outil de segmentation ne convient à toute image. Seule l'utilisation d'un mélange judicieux et réfléchi d'outils de segmentation permet de détecter tel objet dans telle zone, suivant sa forme, sa texture, son orientation, etc. À notre connaissance, ce genre de techniques de segmentation hybride n'a pas encore fait l'objet de travaux de recherche.

4.1.3 Description d'un schéma de segmentation

La phase de segmentation est très complexe, à tel point que les auteurs réalisent habituellement une segmentation uniquement sur des images très particulières. Dans l'optique d'un codage, il est primordial de favoriser l'universalité et l'efficacité de la représentation $\rho(I)$ résultant de la segmentation, éventuellement aux dépens de l'interprétation qu'en ferait un observateur humain. Dès lors, *il ne faut pas s'attendre à une correspondance systématique des objets détectés avec des objets physiques.*

La démarche que nous proposons combine l'une ou l'autre solution actuelle, ce qui la différencie des algorithmes classiques, de manière à détecter tour à tour les objets appartenant aux trois grandes familles de régions. Nous avons également ajouté une phase de croissance de germes qui est nouvelle. Signalons que notre schéma de segmentation s'applique à des images de télévision ; il n'est pas bâti pour des images trop particulières comme des images de texte, car ce type d'application requiert l'introduction d'une information spécifique dès le stade de la segmentation.

Voici les *trois principales étapes du processus de segmentation*, reprises à la figure 4.2 :

- la première étape détecte les textures. L'algorithme extrait d'abord des germes de texture, puis il les extrapole et les étend après comparaison des valeurs extrapolées aux valeurs de l'image I ;
- la deuxième détecte les régions actives, et
- la troisième segmente les régions demeurées en suspens, à savoir les régions à variations lentes.

En procédant en trois phases, nous sommes conscient d'avoir alourdi la démarche de segmentation, mais l'efficacité de codage, assurée par la cohérence spectrale de la représentation, compense cette lourdeur.

La suite du chapitre décrit les trois étapes de segmentation en détail. Hormis la détection des régions texturées et leur croissance après extrapolation, les outils de segmentation sont inspirés de la littérature. C'est au niveau du principe –segmenter une image en utilisant des outils adaptés chacun à un style de région– que se situe l'originalité du schéma global.

4.2 Détection des régions texturées

Parvenir à détecter les textures, c'est-à-dire les régions présentant une certaine régularité spatiale, dans une image naturelle n'est pas une tâche facile. La difficulté s'explique aisément : bien qu'ayant doté tout type de régions d'une cohérence spectrale, en pratique la frontière qui sépare les familles de régions est ténue. Faut-il considérer un mur de briques comme un seul objet ou le segmenter davantage ?

4.2.1 Le principe

Des travaux récents, principalement basés sur l'utilisation de filtres de Gabor [5, 12, 17, 39, 50], aboutissent à d'intéressants résultats de segmentation de textures. L'information utile est tantôt la phase des images filtrées, tantôt le module.

Le principal reproche à adresser à toutes ces méthodes est la détermination d'un seuil dont la valeur dépend fortement de l'image de départ. Pour se départir de la sensibilité au seuil, nous avons proposé, dans la référence [50], de lier sa détermination au spectre de l'image par la recherche d'éventuelles nuées dans le spectre. Dans ce travail, nous avons contourné la difficulté en confiant aux filtres de Gabor l'unique tâche de détecter des germes de texture. De cette façon, la valeur du seuil est moins critique mais, en contrepartie, le processus de segmentation des textures est devenu plus complexe.

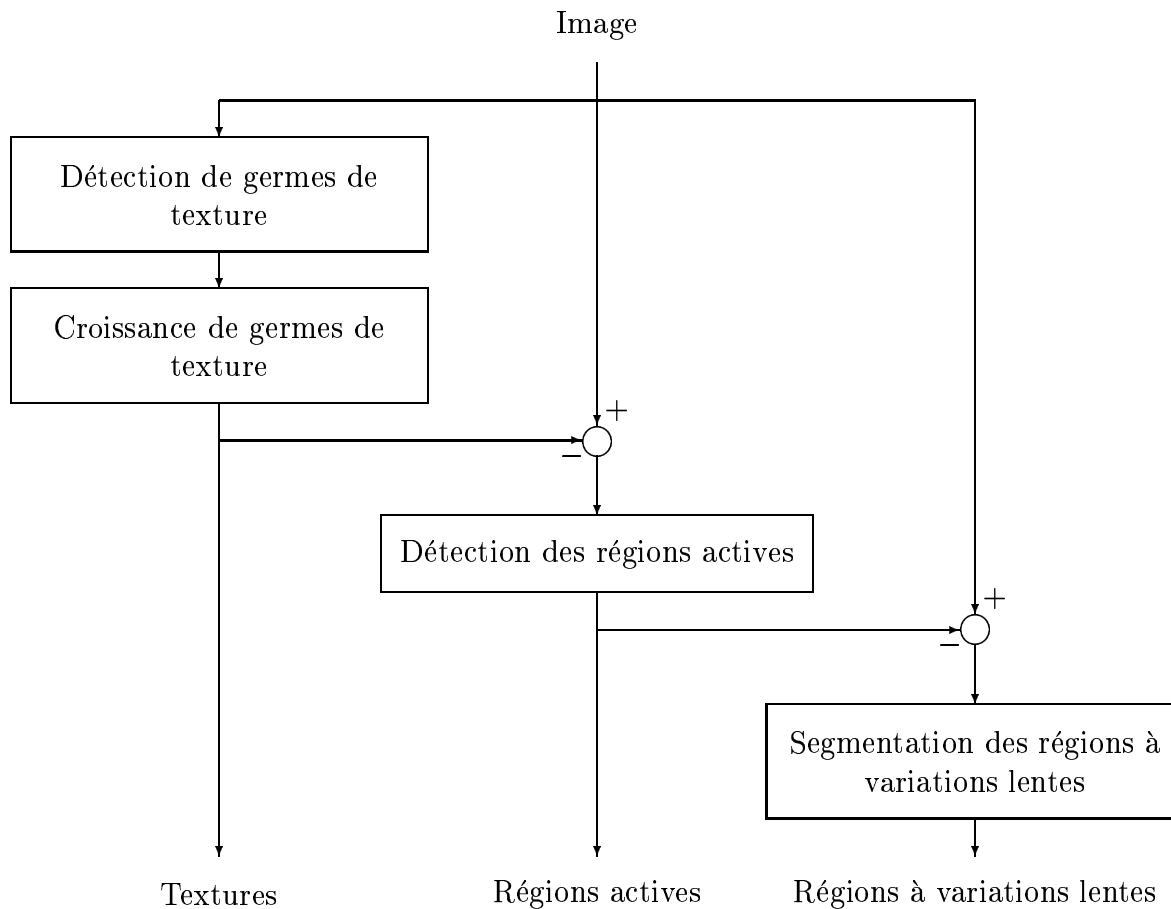


FIG. 4.2 - *Algorithme de détection d'objets en trois phases : (1) Identification de textures par croissance de germes. (2) Détection des régions actives. (3) Segmentation des régions à variations lentes.*

Il s'agit d'un algorithme en trois phases itérées jusqu'à stabilisation du résultat : (1) *extraction de germes de texture par seuillage souple des images filtrées au moyen de quatre filtres de Gabor*, (2) *extrapolation une à une des textures jusqu'à remplissage de l'image ou d'une partie plus restreinte entourant le germe*, et (3) *comparaison des valeurs réelles aux valeurs extrapolées avec croissance éventuelle des germes*.

Le résultat d'une détection de texture par croissance de germe a déjà été fourni au chapitre 3 (figure 3.10, page 62). À cette occasion, nous avons montré une application possible de l'extrapolation de textures.

4.2.2 Expression des filtres de Gabor

La première étape de segmentation s'occupe de la détection des textures. L'étape suivante est l'identification des régions actives. À la base, nous avons cependant utilisé un outil commun, à savoir celui des filtres de Gabor, pour la détermination de ces deux types de régions. C'est pourquoi la présentation de ces filtres précède le détail des étapes de segmentation.

Les filtres de Gabor ont une réponse impulsionnelle de la forme :

$$h(x, y) = g(x', y')e^{2\pi j(Ux+Vy)} \quad (4.1)$$

où

- $(x', y') = (x \cos \phi + y \sin \phi, -x \sin \phi + y \cos \phi)$, c'est-à-dire les coordonnées (x, y) tournées d'un angle ϕ , et
- $g(x', y') = \frac{1}{2\pi\sigma^2}e^{-(x'/\lambda)^2+y'^2/2\sigma^2}$.

La réponse impulsionnelle $h(x, y)$ est donc une fonction complexe sinusoïdale modulée par une gaussienne bidimensionnelle de rapport d'axes λ , de facteur de dilatation σ et où ϕ est l'orientation de l'axe x' par rapport à l'axe x .

La transformée de Fourier correspondante est :

$$\mathcal{H}(u, v) = e^{-2\pi^2\sigma^2[(u'-U')^2\lambda^2+(v'-V')^2]} \quad (4.2)$$

où

- $(u', v') = (u \cos \phi + v \sin \phi, -u \sin \phi + v \cos \phi)$, et
- (U', V') est obtenu par rotation du point (U, V) par le même angle ϕ .

$\mathcal{H}(u, v)$ est une gaussienne passe-bande de rapport d'axes $1/\lambda$, tournée d'un angle ϕ , centrée en $F = \sqrt{U^2 + V^2}$ et d'orientation $\theta = \tan^{-1}(V/U)$. On prend habituellement $\theta = \phi$.

Présentés ainsi, les filtres de Gabor jouissent de quatre degrés de liberté qui permettent d'extraire une partie de l'information spectrale de l'image. Il s'agit du rapport d'axes λ , du facteur d'excentricité σ des gaussiennes dans les deux plans, et de la translation imposée par (U, V) ou (F, ϕ) dans le domaine des fréquences. C'est principalement pour la facilité d'interprétation des paramètres que nous avons choisi les filtres de Gabor, même s'il est vrai que ceux-ci minimisent un certain type d'incertitude introduit par DAUGMAN [12].

Grâce à la forme gaussienne des filtres, les enveloppes des images filtrées apportent une *information spectrale locale* en chaque pixel. De plus, elles renseignent sur le contenu énergétique de l'image dans la direction du filtre choisi. Alors, pour extraire les signaux texturés, on peut mettre plusieurs images filtrées en concurrence. Si, suite à une comparaison des énergies de ces images, une sortie au moins est supérieure à un seuil, il s'agit d'un morceau d'une texture ou d'une région active, la distinction étant opérée à l'aide d'un second seuil (voir infra). Et finalement, les points pour lesquels les énergies sont faibles appartiennent à des régions à variations lentes.

Pour réaliser cette séparation, nous reprenons des bribes d'un schéma introduit par BOVIK [5] et appliqué par après à des images naturelles [49, 50]. Le schéma initial comprend entre autres un filtrage avec quatre filtres passe-haut orientés dans un demi-plan spectral, un filtrage passe-bas et un seuillage. Ces mêmes étapes sont utilisées dans le cadre de l'extraction des textures et illustrées ci-après. Nous reviendrons brièvement sur le sujet lors de la détection des régions actives.

4.2.3 Détection des textures : description, détails pratiques et résultats

La détection des textures s'effectue en trois phases.

- *Étape 1 : initialisation – détection de l'ensemble des germes de texture.*

L'initialisation du processus de segmentation consiste à détecter la présence de germes de texture. Pour y parvenir, nous avons utilisé des filtres de Gabor dont les sorties ont été traitées par un filtre passe-bas, puis seuillées. Le processus complet est illustré à la figure 4.3, où l'indice des filtres h reprend la valeur de leur orientation ϕ .

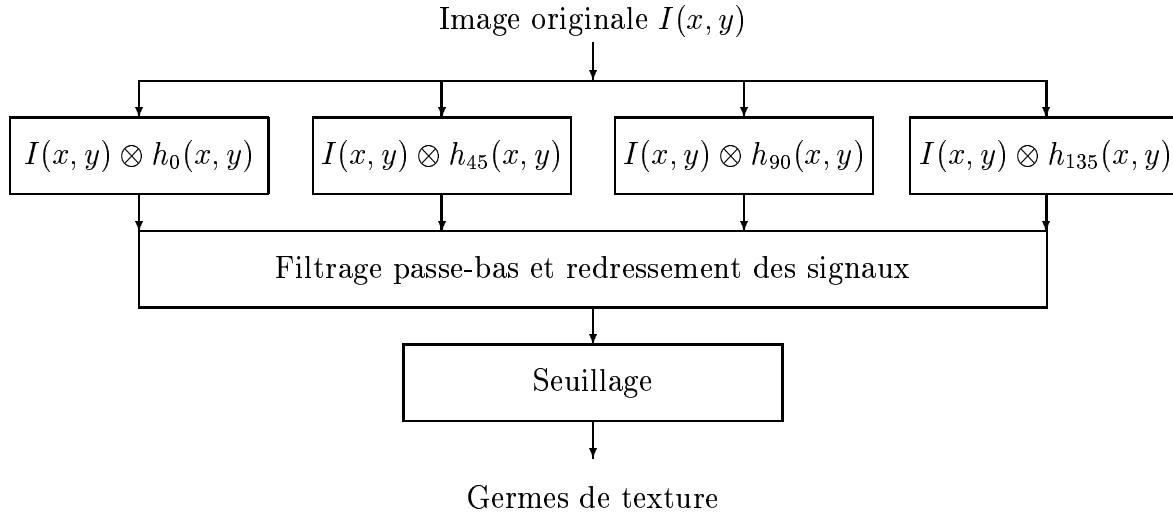


FIG. 4.3 - *Processus de détection des germes de texture. Les indices des filtres représentent leur orientation en degrés.*

Détaillons les composantes du processus d'extraction de germes:

Étape 1.a : filtrage de l'image par quatre filtres orientés hautes fréquences.

Nous devons fixer les quatre paramètres des filtres de Gabor que sont l'orientation ϕ , la bande passante (ce qui revient à fixer σ), la fréquence centrale F et le rapport des axes λ .

Après implémentation, nous avons remarqué que quatre filtres suffisent pour obtenir des résultats satisfaisants; utiliser davantage de filtres complique le seuillage. De plus, la détermination des paramètres des filtres n'est pas très critique. Les valeurs que nous donnons ici concernent le traitement d'image de taille 256x256. Signalons toutefois que la résolution spectrale croît en même temps que la taille de l'image, ce qui interdit pratiquement de traiter des images plus petites.

Orientation ϕ . Le choix de la valeur des angles est aisé. Les filtres sont dirigés dans les directions de 0° , 45° , 90° et 135° . Ils suffisent à caractériser les principales directions du spectre, car l'image étant réelle, son spectre est symétrique. D'autre part, comme le spectre d'une image accumule souvent une partie importante de l'énergie autour des axes horizontaux et verticaux,

le fait d'intercaler des filtres entre ces deux directions améliore la discrimination.

Bande Passante. L'influence de la bande passante sur les résultats est considérable. Il faut à tout prix limiter les recouvrements entre filtres et détecter toutes les orientations marquées avec seulement quatre filtres. Une bande passante d'une octave nous semble un bon compromis.

Fréquence centrale F . Cette fréquence détermine la précision des contours parce que ceux-ci se caractérisent par un riche contenu en hautes fréquences. Plus F est élevée, plus les contours seront nets et la bande passante large. La valeur $F = 120$ a parfaitement convenu.

Rapport d'axes λ . Le rapport entre les axes permet de jouer sur la sélectivité angulaire. A priori, $\lambda = 1$ convient à nos besoins

La figure 4.4 montre la trace d'un filtre de Gabor orienté à 45° dans le plan transformé. Le cercle blanc dessine la bande passante à 3 dB. Il apparaît que les filtres se recouvrent entre eux, et cela pour éviter d'oublier des fréquences qui seraient situées entre deux filtres. De plus, comme les images naturelles comprennent toute une gamme de fréquences différentes, l'effet de recouvrement est à peine perceptible. Un autre détail d'implémentation doit être mentionné : l'effet d'apodisation. La théorie du traitement du signal linéaire nous apprend que l'échantillonnage de la transformée du signal filtré périodise le signal de départ. Autrement dit, le signal qui touche le bord de droite contamine le contenu spectral local du signal situé près du bord gauche de l'image. La solution consiste à ajouter un bord fictif. Ainsi, nous avons pu atténuer les effets néfastes de la périodisation du signal. D'autre part, ayant choisi $F = 120$ pour une image 256×256 , les volumes des filtres diffèrent sensiblement, rompant ainsi l'invariance en rotation. L'expérience montre cependant qu'il est inutile de normaliser les sorties. Nous montrerons des exemples d'images filtrées après avoir introduit le post-filtrage et décrit le seuillage.

Étape 1.b : post-filtrage passe-bas. Les enveloppes des sorties des filtres subissent un second filtrage passe-bas destiné à éliminer les très hautes fréquences, nuisibles à une bonne discrimination par seuil. Nous avons choisi de prendre un Butterworth avec une fréquence de coupure de 60. Un autre filtre conviendrait tout autant. Enfin, le signal est redressé et les sorties sont normalisées de façon à occuper la plage dynamique de 0 à 255.

Étape 1.c : seuillage. Les images filtrées contiennent une information à propos du contenu énergétique local dans une certaine orientation. Malheureusement, un bruit s'y superpose ; le post-filtrage a supprimé une bonne part du bruit haute-fréquence avec, en contrepartie, une légère dégradation subséquente du signal. Ne disposant d'aucune indication quant au type de bruit, le procédé le plus commode pour réduire son influence consiste à seuiller les images. Un seuil inférieur, noté $seuil_1$ indique alors si le niveau énergétique suffit pour que le point appartienne à une région texturée ou à une région active. Ainsi, si une sortie au moins dépasse la valeur de ce seuil alors le signal est actif ou texturé. Cette dernière distinction est rendue possible par l'introduction d'un second seuil, noté $seuil_2$. En chaque point, nous mesurons l'écart entre les deux plus

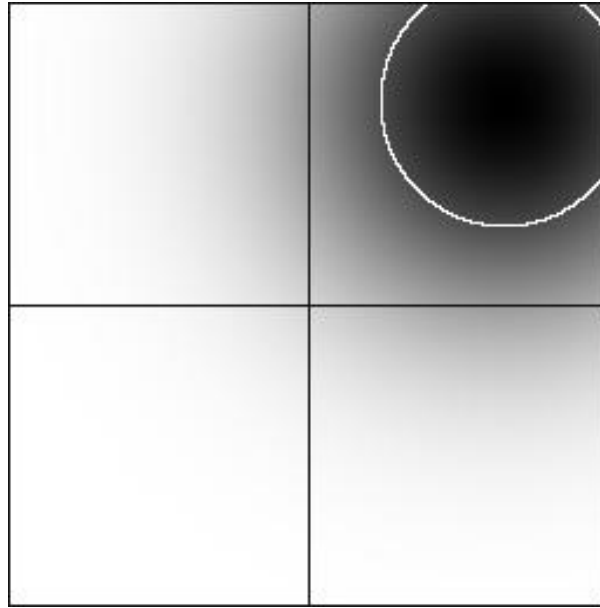


FIG. 4.4 - Trace d'un filtre de Gabor dans le plan transformé. Le cercle blanc représente la bande passante à 3 dB.

grandes valeurs parmi les quatre images filtrées redressées. S'il excède $seuil_2$ alors le point appartient à une texture.

Comme mentionné précédemment, la détermination du seuil est souple et pas trop critique parce que la phase d'extrapolation corrige les manquements de l'extraction. Avec une dynamique de 0 à 255, nous avons choisi les valeurs de seuil suivantes : $seuil_1 = 40$ et $seuil_2 = 20$. Toutes les images ont été traitées avec ces valeurs.

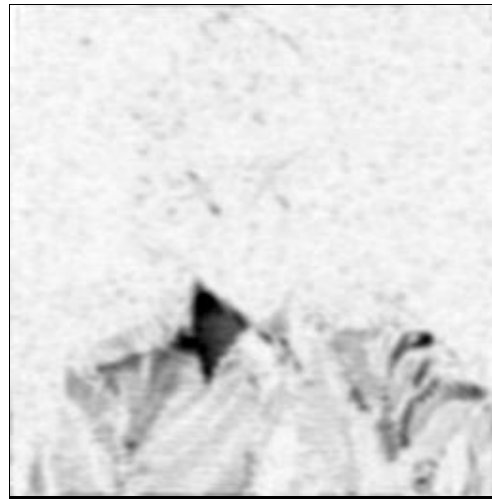
Les images des figures 4.5 et 4.6 illustrent le filtrage et le seuillage. Il s'agit dans l'ordre de l'image originale, de deux versions filtrées, et d'une image qui indique par différentes teintes de gris quelle est la sortie de filtre la plus importante en chaque point ; les zones blanches désignent les régions à variations lentes parce qu'aucune sortie de filtre n'y est supérieure à $seuil_1$. C'est parmi les régions ayant une même couleur et telles que l'écart entre les deux plus grandes sorties de filtre est supérieur à $seuil_2$ que se trouvent les germes des textures.

- *Étape 2 : construction d'un masque d'extension maximale et extrapolation à l'intérieur de celui-ci.*

Le mécanisme de croissance de la texture repose sur un critère d'identification qui compare le signal extrapolé au signal réel. Il est intéressant de doubler cette contrainte d'une autre purement spatiale : un germe est autorisé à recouvrir les régions actives, à déborder très légèrement sur les régions à variations lentes, puisque de toute façon nous corrigeons les frontières par la suite, mais en aucun cas il ne peut englober des points d'un autre grand germe. Ainsi, le col de la chemise de la figure 4.5 (a) est une région qui ne pourra être recouverte par aucune autre. En termes morphologiques, on définit un *masque géodésique*. Ce masque est construit en dilatant l'ensemble des régions actives et les surfaces pour lesquelles le même filtre est maximum. Puis, le germe de départ est reconstruit dans ce masque, noté MG , de manière à *supprimer les*



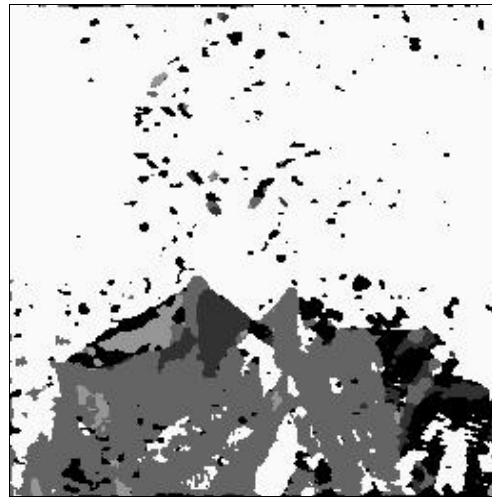
(a) Image originale.



(b) Image filtrée à 45°.



(c) Image filtrée à 90°.

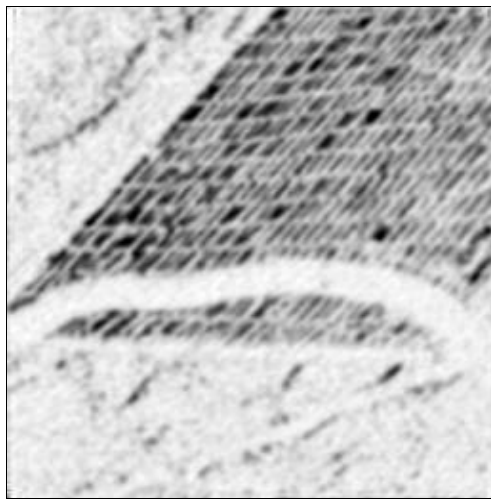
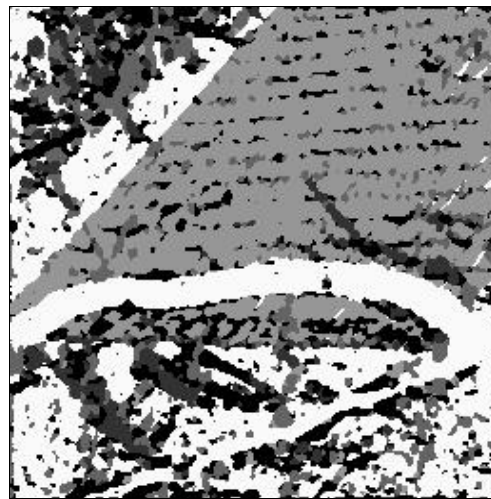


(d) Le filtre maximum en chaque point.

FIG. 4.5 - *Premières étapes de la détection des textures : un premier exemple.*



(a) Image originale.

(b) Image filtrée à 90° .(c) Image filtrée à 135° .

(d) Le filtre maximum en chaque point.

FIG. 4.6 - Premières étapes de la détection des textures : un second exemple.

parties non connexes au germe. D'autres petits traitements (suppression des petites régions, filtrages, ...) nettoient le masque pour éviter l'apparition de petits détails insignifiants.

– *Étape 3: croissance d'un germe.*

Le nœud de la question de la croissance des régions tient au critère adopté: faut-il prendre un critère local ou un critère global? Comment adapter la qualité de l'extrapolation au mécanisme de croissance? Notre critère s'appuie sur la *variance du signal d'erreur*, notée σ , entre le signal de départ $I(x, y)$ et le signal extrapolé, approximant $I(x, y)$, à l'intérieur de la région, c'est-à-dire que l'on considère $w(x, y)f(x, y)$ où $w(x, y)$ est l'indicatrice définie sur D_w et $f(x, y)$ le signal extrapolé. Mais avant de tester la concordance entre le signal réel et le signal d'extrapolation, nous effectuons un filtrage morphologique de l'image originale (fermeture de l'ouverture) par un segment linéaire de longueur 3 dans la direction perpendiculaire à l'orientation de la région. Ainsi sont conservées les transitions, tout en permettant une légère réduction du bruit. Pour la commodité, nous maintenons ci-après la notation $I(x, y)$ pour l'image filtrée.

Soit $f_i(x, y)$ le signal extrapolé à l'étape i . L'objectif de la croissance des régions est d'étendre la fenêtre de sorte que $D_{w_i} \subset D_{w_{i+1}} \subseteq MG$. Il est atteint en mesurant l'écart type sur D_{w_i} par

$$\sigma_i = \sum \sqrt{[f_i(x, y) - I(x, y)]^2} \quad (4.3)$$

puis en accroissant le germe par le biais de la relation suivante:

$$w_{i+1}(x, y) = 1 \quad \text{si} \quad w_i(x, y) = 1 \quad \text{ou} \quad |f_i(x, y) - I(x, y)| \leq \alpha \sigma_i \quad (4.4)$$

La figure 4.7 reprend le schéma complet.

Les essais montrent que $\alpha = 1,5$ est une valeur adéquate. Prendre une autre valeur α , de 1 à 3 par exemple, modifie la vitesse de croissance des germes sans affecter la nature du résultat; bien entendu, un faible coefficient offre plus de sécurité. Pour éviter toute dérive, σ_i est toujours le minimum des variances calculées durant la croissance d'un germe: $\sigma_i = \min\{\sigma_0, \dots, \sigma_{i-1}\}$. Le processus entier est itéré jusqu'à stabilisation: $w_{j+1}(x, y) = w_j(x, y)$ pour tout $(x, y) \in \mathbb{Z}^2$.

Choisir un *critère d'identification en fonction de la variance du signal d'erreur* est la *clef du processus de croissance*. Si la variance est petite, cela signifie que le signal extrapolé $f_i(x, y)$ reconstitue bien la texture à l'intérieur de la région D_{w_i} , comme c'est le cas pour des textures de synthèse. Dès lors, les points ajoutés doivent maintenir cette qualité de restitution. Par exemple, 50 coefficients de Fourier permettent déjà d'obtenir une bonne approximation d'une texture de synthèse (l'écart type est de l'ordre de 10). Une texture issue d'une image quelconque ne présente pas une telle cohérence. Dans ce cas, la variance est plus élevée et l'adjonction de nouveaux points moins sévère. Notons que toutes les approximations ont été obtenues avec 50 coefficients de Fourier –sans doute est-il préférable de lier ce nombre à la taille de la région–, et que le germe est régulièrement reconstruit sous le germe étendu pour supprimer les parties non connexes. En effet, les essais ont indiqué que l'algorithme

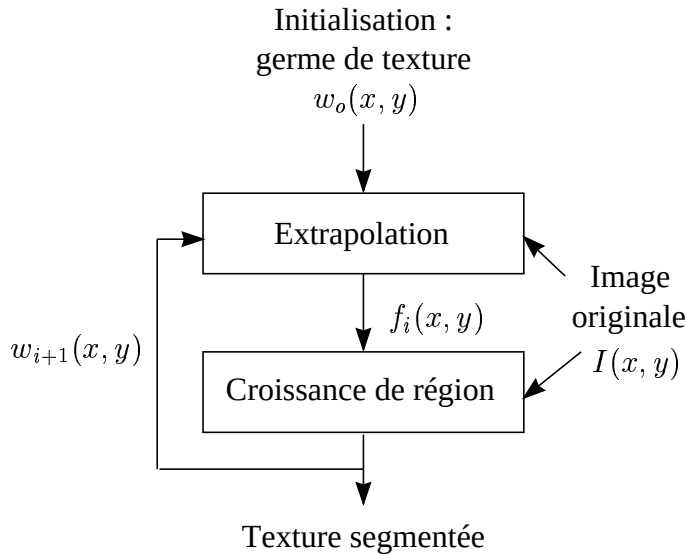


FIG. 4.7 - *Processus itératif de croissance pour la segmentation de textures.*

tente de forcer le raccord de deux grandes régions non connexes en favorisant, dans le processus d'extrapolation, la description de la surface intermédiaire même si sa nature s'écarte des deux textures homogènes.

Les figures 4.8 et 4.9 montrent quelques étapes de la croissance d'une texture d'une image naturelle. La figure 4.8 reprend tour à tour : (a) l'image originale, (b) l'image avec les numéros des images filtrées maximales en chaque point, (c) le germe de texture à étendre (il est entouré d'un trait blanc) et (d) le masque de reconstruction qui fixe l'extension maximale du germe. Quelques étapes intermédiaires du processus de segmentation des textures par croissance de germes sont reprises dans la seconde figure. Y sont successivement rassemblées : (a) l'extrapolation du germe de texture initial, détecté dans le toit, (b) une étape intermédiaire de la croissance du germe, (c) le résultat final de la détection des textures et (d) les deux régions texturées et l'ensemble des régions actives (en gris foncé). Nous allons voir à présent comment ont été obtenues les régions actives de la dernière image.

4.3 Recherche des régions actives

Les régions actives sont les régions à contenu spectral en hautes fréquences important. Elles se déterminent au moyen de $seuil_1$ après la détection des textures. Tout ce qui n'appartient pas à une texture et dépasse la valeur de ce seuil est inséré dans une région active. Il n'est nul besoin de vouloir segmenter davantage les régions actives car elles contiennent quantité de contours et sont d'aspect irrégulier. Il est dès lors préférable de les coder directement avec des outils spectraux au lieu de vouloir les représenter par une série de petits objets dont le coût de codage de la forme serait prohibitif. Peut-être pourrait-on même coder des contours irréguliers de cette manière ?

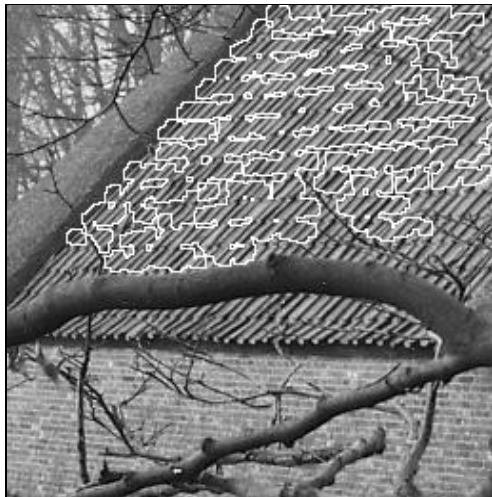
Les régions restantes sont des régions à variations lentes.



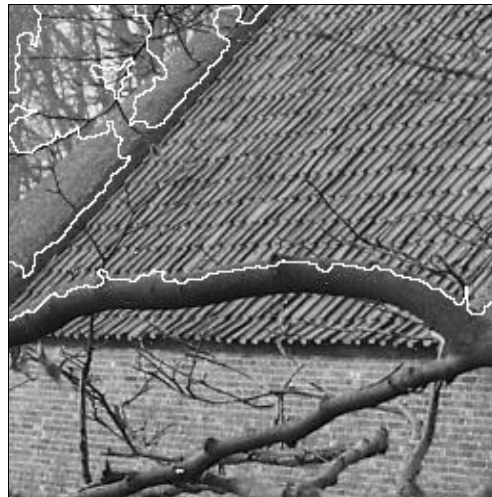
(a) Image originale.



(b) Numéro des filtres maximum (teintes de gris).

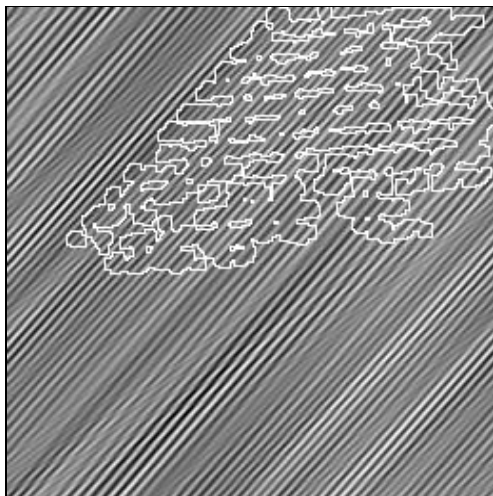


(c) Le premier germe de texture.

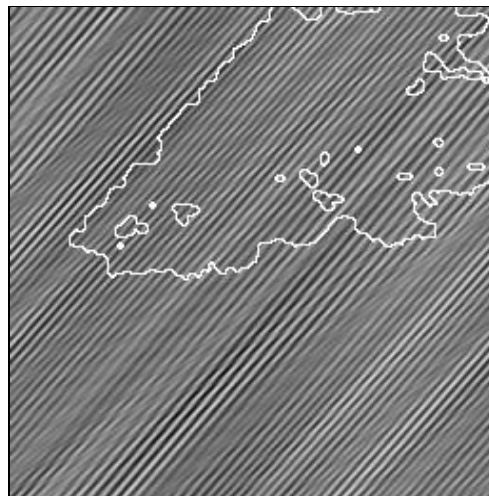


(d) Masque d'extension maximale.

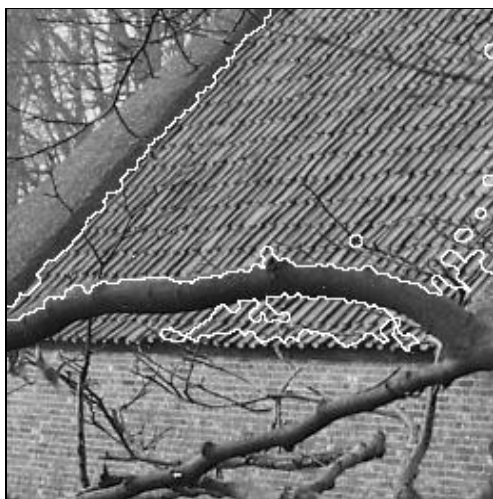
FIG. 4.8 - *Initialisation du processus de croissance d'un germe de texture.*



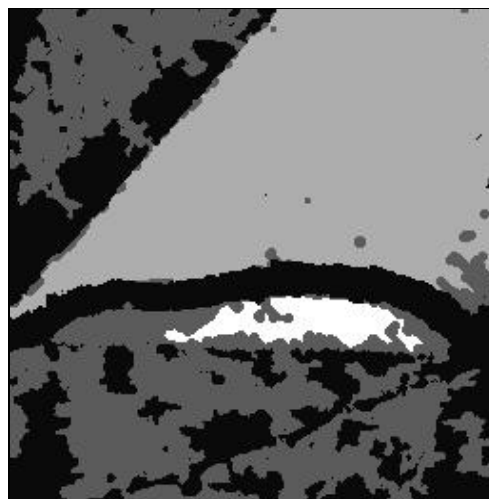
(a) La première extrapolation.



(b) Étape intermédiaire du processus de croissance (signal extrapolé).



(c) Résultat de la détection des textures.



(d) Régions texturées (blanc et gris clair) et régions actives (gris foncé). En noir, les régions à variations lentes.

FIG. 4.9 - Quelques résultats de la détection des textures par croissance de germes.

4.4 Segmentation des régions à variations lentes

Pour segmenter les régions à variations lentes, nous avons initialement songé à appliquer un détecteur de contours et à fermer ensuite les contours extraits [102, pages 50–53]. À vrai dire, il n'est probablement pas nécessaire de développer un nouvel outil de segmentation parce que les régions à variations lentes ne contiennent pas de singularité susceptible de conduire à des problèmes insurmontables.

Dans la méthode proposée, l'outil de base est la ligne de partage des eaux dont il a été question précédemment (cf. section 2.4, page 38). Il est connu que les résultats dépendent fortement des marqueurs dans ce genre de segmentation. Notre algorithme mêle habilement ligne de partage des eaux, marquage des régions texturées ou actives, reconstruction et extrapolation de texture. Voici, décrites dans l'ordre, les différentes étapes qui le composent :

1. *Correction des contours des régions texturées et actives.*

Il est entendu que le travail de segmentation des régions texturées ou actives a déjà été mené à bien. Mais avant d'appliquer un masque qui élimine ces régions, nous corrigeons les contours des régions déjà extraites. Cette opération, rendue nécessaire principalement à cause de la perte de précision qu'occasionne le post-filtrage passe-bas, est obtenue en marquant la surface recouverte par les régions texturées et actives, légèrement érodée, puis en appliquant la ligne de partage des eaux sur le gradient morphologique (dilaté moins érodé) de l'image (voir figure 4.10). La dernière image de la figure 4.10 indique en blanc tout ce qui est considéré finalement comme faisant partie de régions texturées ou actives. Il n'est plus permis de toucher à ces régions par la suite.

2. *Segmentation du gradient morphologique par ligne de partage des eaux.*

Une nouvelle ligne de partage des eaux du gradient morphologique est appliquée après avoir considéré comme marqueurs toutes les zones pour lesquelles le gradient était inférieur à un certain seuil ; nous avons pris un seuil égal à 15 (bien inférieur à celui de l'étape 1) pour une dynamique qui allait de 0 à 255, préférant une sur-segmentation à une sous-segmentation. Nous sélectionnons ensuite la plus grande région dans la surface restant à segmenter. La figure 4.11 montre la découpe en régions effectuée par la ligne de partage des eaux et les trois premières régions détectées par l'algorithme. On peut y voir que la ligne de partage des eaux fonctionne très mal dans les régions texturées. D'autre part, vu le grand nombre de régions qui composent cette image, il faut une étape de fusion de régions.

3. *Extrapolation et fusion des régions jusqu'à stabilisation.*

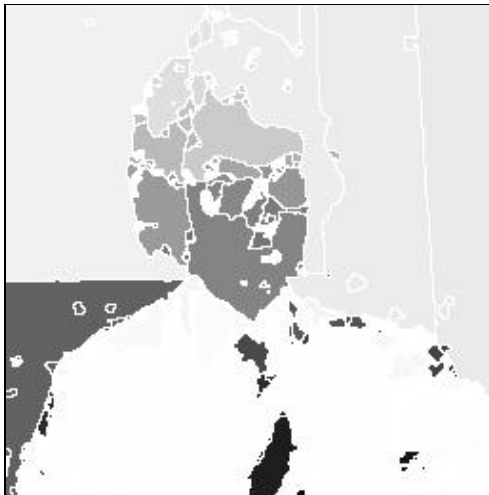
Par un processus similaire à la détection des textures, les régions sont extrapolées à l'ensemble de l'image. Bien sûr, en raison du faible contenu fréquentiel des régions, le nombre de raies spectrales à conserver est petit (5 par exemple). Il convient aussi d'adapter le critère de similitude entre les pixels à cause de la faible variance de la luminance dans ces régions. Nous avons pris un critère de similitude de 3σ . Signalons encore que, si la fenêtre originale est reconstruite dans la fenêtre agrandie pour éviter une augmentation du nombre de connexité, toute une région est annexée dès que le nouveau masque la recouvre à 50%. Le procédé consiste donc à fusionner des régions et non pas à les accroître par ajout progressif de quelques points appartenant aux



(a) Image originale.



(b) Régions texturées (teintes de gris) et actives (taches noires).



(c) Ligne de partage des eaux après marquage.



(d) En clair, les zones à écarter.

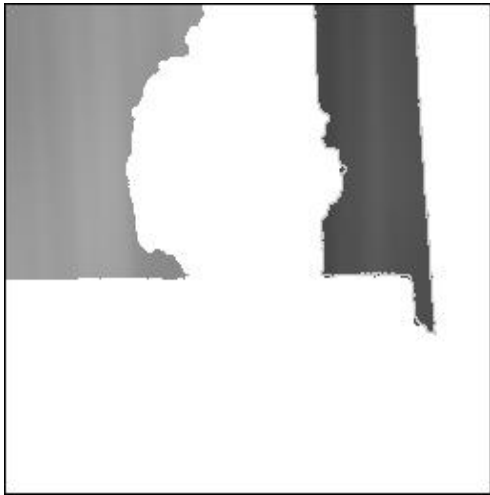
FIG. 4.10 - *Initialisation de la segmentation des régions à variations lentes : correction des contours des régions texturées et des régions actives.*



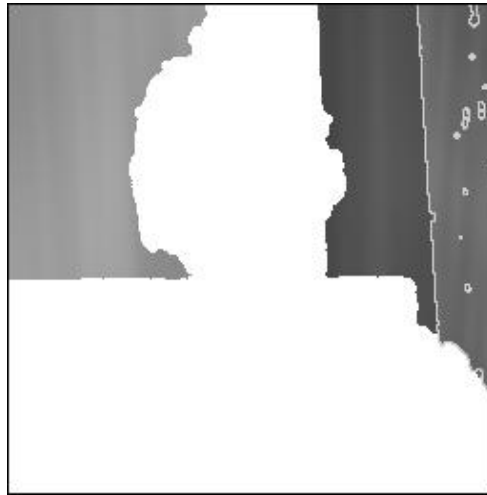
(a) Seconde ligne de partage des eaux.



(b) Première région détectée.



(c) Deuxième région détectée.



(d) Troisième région détectée.

FIG. 4.11 - *Détection et croissance des régions à variations lentes.*

régions voisines. De cette manière, nous bénéficions de la précision des contours de la ligne de partage des eaux. L'image 4.11 (b) montre le résultat final de la croissance de la première région. Le trait noir (respectivement blanc pour les images (c) et (d)) entoure la région tirée de la ligne de partage des eaux. Sur la même image, nous avons reproduit l'approximation de la région finale avec seulement 5 coefficients de la transformée de Fourier! La qualité de reconstitution est remarquable ainsi que la précision des contours. L'image 4.11 (d) illustre mieux encore le rôle pratique de la fusion des régions.

4.5 Étude du mouvement

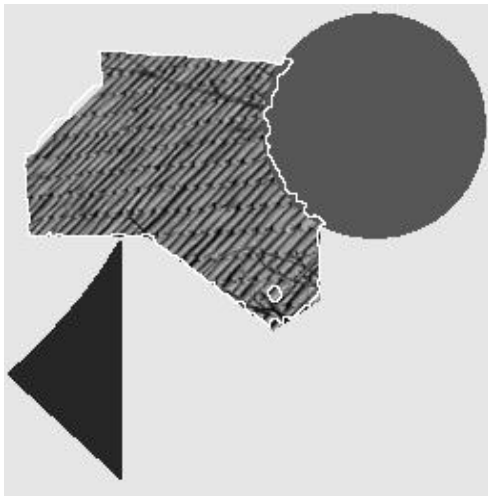
Comme les objets évoluent dans l'espace, ils peuvent se recouvrir. La tâche de la mise en correspondance de régions dans des séquences d'images s'en trouve plus ardue puisqu'un simple critère de forme échoue à établir toutes les correspondances. L'extrapolation des textures permet de résoudre partiellement le problème. Pour comprendre le mécanisme, nous proposons un exemple reporté à la figure 4.12. Soient (a) et (b), deux images successives d'une séquence. La déformation de l'objet texturé annihile toute possibilité de recourir à des outils usuels de mise en correspondance tel le 'block matching'. Non seulement notre algorithme de segmentation détectera les textures –les régions texturées qu'il a détectées sont entourées d'un trait blanc– mais nous pouvons ensuite comparer les signaux extrapolés de ces deux textures ou encore leur spectre. Les images (c) et (d) montrent le spectre des signaux extrapolés respectifs. Il est clair que ces deux spectres sont fortement corrélés, conduisant à une mise en correspondance des régions.

Le procédé de mise en correspondance des signaux texturés peut aussi servir à rassembler des régions non connexes dans une image. En effet, la reconstruction de l'ancien germe dans le germe étendu durant la phase de croissance des textures empêchait tout regroupement de textures similaires non connexes, condition indispensable pour un fonctionnement optimal de la croissance. Pourquoi ne pourrions-nous pas les réunir dans une phase d'analyse supplémentaire, d'autant plus qu'à l'issue de la segmentation nous disposons de chacun des signaux extrapolés? Voilà qui démontre une fois encore la puissance de l'outil d'extrapolation développé dans le chapitre précédent.

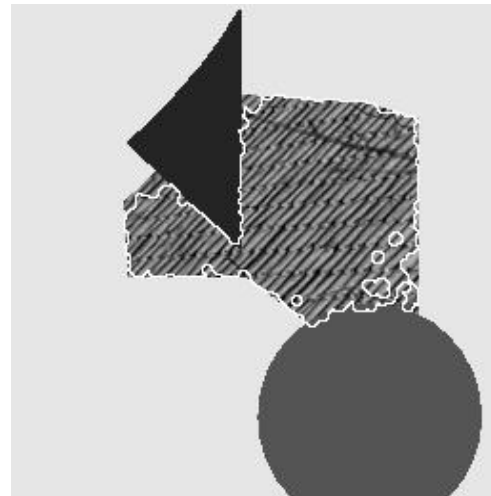
Conclusions du chapitre

Le procédé de segmentation décrit dans ce chapitre s'articule sur un modèle d'image qui classe les régions en trois catégories: les régions texturées, les régions actives et les régions à variations lentes. L'algorithme mène alors une recherche de ces régions en adaptant l'outil d'analyse au contenu spectral de ces régions. La démarche complète est résumée à la figure 4.13.

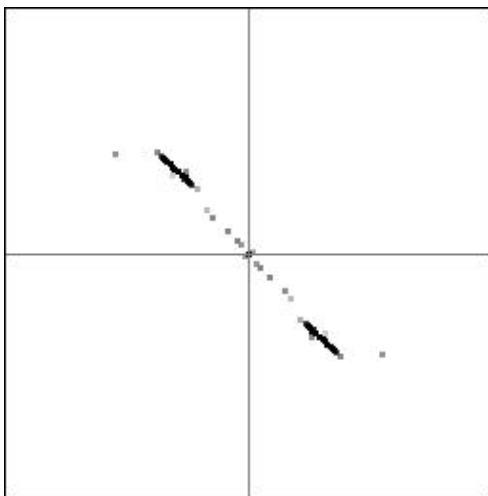
Pour extraire les régions texturées, l'algorithme détecte des germes de texture au moyen de quatre filtres de Gabor dont les sorties sont seuillées. Il extrapole ensuite le contenu des germes, pris un à un, à toute l'image et leur ajoute les points qui concordent dans une certaine mesure avec ce signal étendu. L'étape de croissance est itérée jusqu'à stabilisation. Au terme de la phase de détection des textures, l'algorithme récupère par différence toutes les régions qui n'ont pas été considérées comme des textures et dont le contenu énergétique



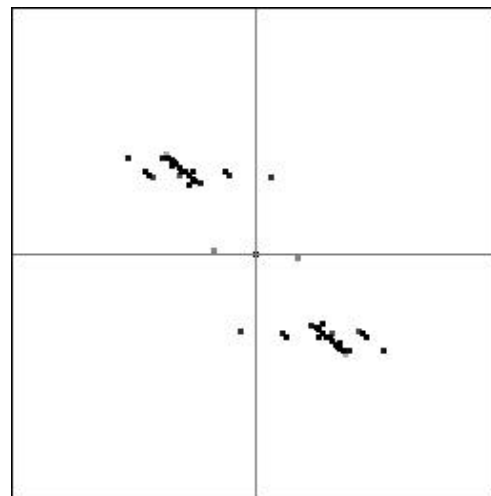
(a) Texture détectée dans l'image 1.



(b) Texture détectée dans l'image 2.



(c) Transformée de Fourier de la texture 1 après extrapolation.



(d) Transformée de Fourier de la texture 2 après extrapolation.

FIG. 4.12 - *Analyse spectrale des textures extrapolées pour la reconnaissance d'objets dans des séquences d'images.*

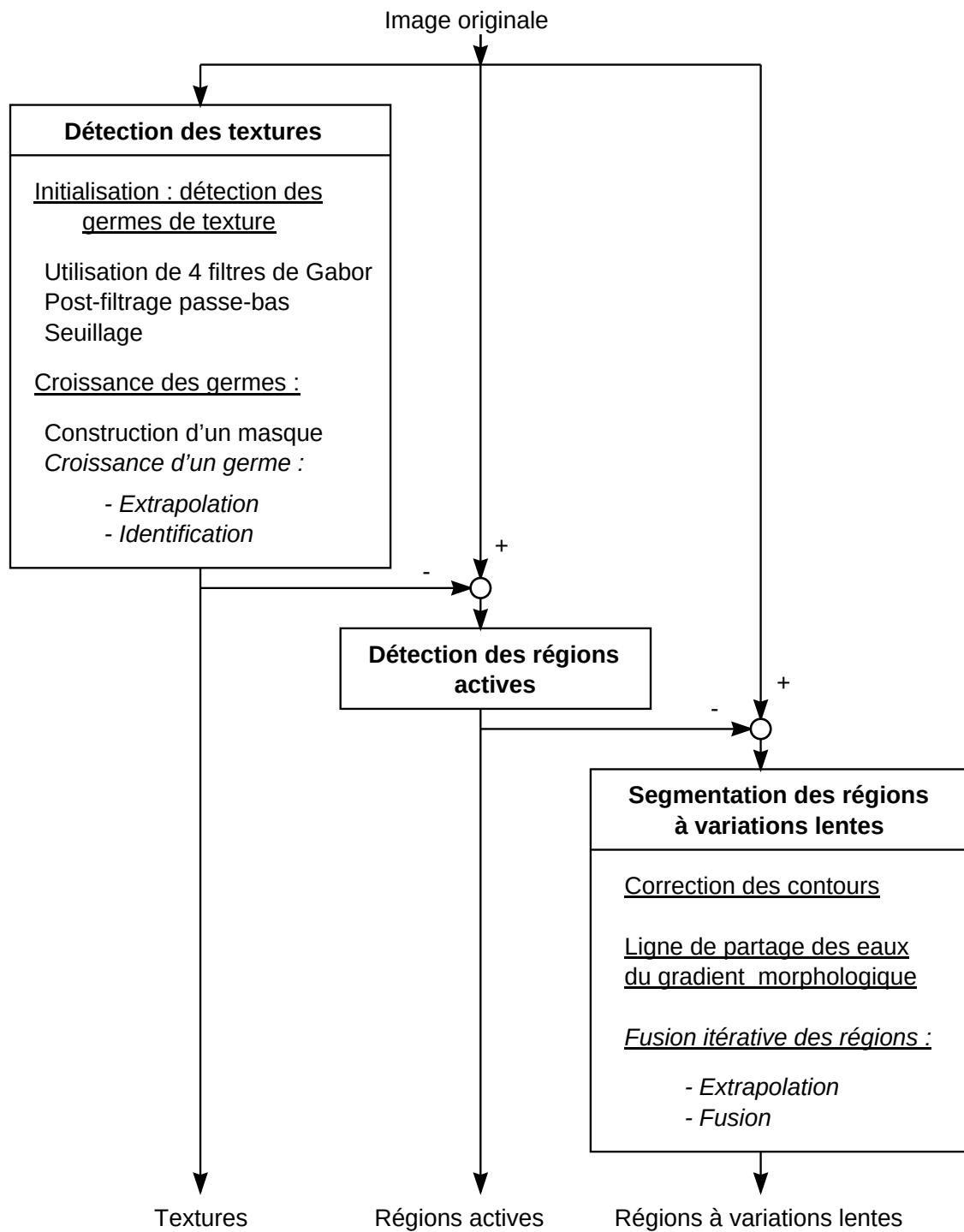


FIG. 4.13 - Synthèse des étapes intervenant dans le procédé de segmentation. Les parties itératives sont indiquées en italique.

dépasse un certain seuil. Ces régions constituent les régions actives. La figure 4.14 (b) représente les régions texturées et actives extraites par l'algorithme. Dans cette image particulière, les taches grises sont les textures et les taches noires désignent les régions actives. La surface blanche regroupe les régions non considérées durant les premières phases de la segmentation. Il ressort de plusieurs essais que ces premières étapes s'avèrent particulièrement efficaces. De plus, nous avons assuré une cohérence entre la segmentation et le codage des textures parce que tous deux s'appuient sur des notions de représentation spectrale de l'image.

De même que la détection des textures s'effectue en plusieurs étapes, la recherche des régions à variations lentes procède à une correction des contours des régions texturées et actives, et à une fusion des régions à variations lentes, issues d'une ligne de partage des eaux peu sévère mais contenant toutes les transitions. La correction des contours des régions texturées et actives s'obtient en érodant la surface de ces régions et en insérant ces marqueurs dans une ligne de partage des eaux du gradient morphologique. Le résultat final de cette étape est repris à la figure 4.14 (c). La sur-segmentation que laissait présager la forme de la ligne de partage des eaux a pu être évitée grâce à la fusion de régions par un procédé similaire à la croissance des régions texturées. La segmentation de l'image complète est reportée en (d).



(a) Image originale.



(b) Régions texturées (teintes de gris) et actives (taches noires).



(c) Segmentation des régions à variations lentes.



(d) Résultat final de la segmentation.

FIG. 4.14 - *Résumé des principales étapes du procédé de segmentation.*

Deuxième partie

**Traitement de la forme des objets :
filtrage et codage des contours**

Introduction

Dans un schéma basé sur la notion d'objet, un opérateur de segmentation produit la représentation de formes $\rho(X)$ assimilée à une image de labels $L(x, y)$, c'est-à-dire de régions composées de pixels ayant un même numéro, et une représentation de textures implicite $\rho(T)$. L'objet de cette partie du document est la présentation d'éléments intervenant dans un schéma de codage des contours.

De toutes les informations contenues dans une image, la forme joue un rôle particulier. C'est en effet par la forme que s'opère, au niveau du cerveau, une première phase de reconnaissance cognitive des objets. Quoi de plus naturel dès lors que se préoccuper de la préservation de la forme dans les algorithmes de traitement d'image ou de codage? Pourtant, assez curieusement, la question du codage de la forme n'a pas été étudiée en profondeur ces dernières années. C'est d'autant plus étonnant que l'on multiplie les tentatives pour obtenir des codeurs d'image orientés par la notion d'objet. Et l'expérience de prouver que le débit associé au codage des contours dépasse souvent le débit propre au codage des textures, allant de 50% à 90% du débit total. On comprend mal alors pourquoi les recherches avancent si timidement dans le domaine de codage des contours alors qu'il s'agit d'un point clef, à résoudre préférentiellement.

Existe-t-il un codeur de contours idéal? Sur le plan conceptuel, les techniques de codage de la littérature s'intègrent toutes dans le bloc-diagramme de la figure 4.15. Elles débutent par un filtrage qui lisse les contours. S'ensuit alors une réduction de représentation qui, après avoir organisé la représentation de contours $\rho(X)$, confie la représentation réduite $\rho_r(X)$ à un codeur entropique. Une éventuelle rétroaction modifie les paramètres de filtrage afin de choisir le degré de déformation des contours en regard du débit. L'idée est la suivante: une méthode de référence facile à mettre en œuvre (par exemple 1,3 bit par point de contour) permet d'évaluer le débit sans filtrage préalable et, en fonction du taux de compression obtenu, la boucle adapte les paramètres de l'opération de filtrage.

La plupart des techniques de codage existantes travaillent à restitution parfaite des contours et ce principalement pour deux raisons:

- le coût du codage n'est pratiquement pas modifié par un filtrage, d'où la quasi-absence d'un filtrage dans les procédés de codage par suivi de contours, et
- il faut trouver un procédé de filtrage ajusté au codage sous peine d'obtenir un schéma global au comportement imprévisible à la moindre modification des paramètres du filtre.

Il est pourtant dommage de se priver de la possibilité d'effectuer un filtrage de contours dans certaines applications alors que, parfois, une légère imprécision sur les contours provient de l'influence d'un bruit.

Notre travail propose *une nouvelle représentation des formes* axée sur une description de la surface des régions, mais surtout, il la combine intimement à *un procédé de filtrage par éléments de surface*. C'est le principal intérêt de la méthode.

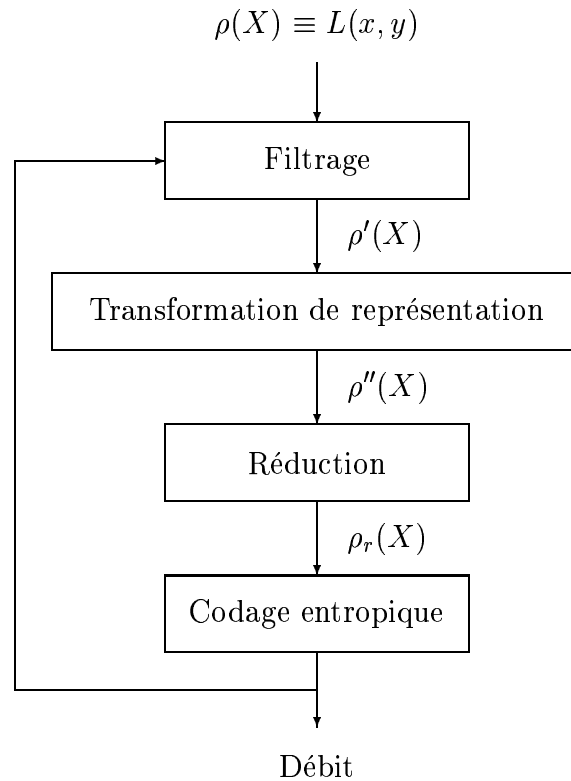


FIG. 4.15 - *Schéma complet d'un codeur de contours.*

Plan

Peut-être la raison majeure qui pousse à ne pas utiliser une technique de filtrage avant le codage des contours est-elle la réelle difficulté de réalisation des opérations qui la composent. Nous avons également été confronté à des problèmes de réalisation en développant notre technique de représentation des contours.

En réponse, le chapitre 5 présente des nouveaux algorithmes de calcul. Leur champ d'application est suffisamment large pour couvrir tous nos besoins. Il apparaît aussi que ces algorithmes appliqués au calcul des opérations classiques de la morphologie mathématique se révèlent plus efficaces que les plus rapides des algorithmes connus. La description algorithmique est doublée d'une réflexion théorique fondamentale sur la réalisation des opérations morphologiques élémentaires.

La description complète des éléments du codeur s'opère dans les deux chapitres suivants. Le chapitre 6 détaille plusieurs procédés de lissage de contours, destinés à lisser les parties peu significatives des formes : pixels isolés, petites régions, ... Il introduit aussi le nouveau concept de polymorphisme dont découle un outil de filtrage et qui peut permettre d'accélérer la recherche de la représentation de la forme optimale.

Le chapitre 7 décrit l'expression générale d'une technique de représentation des contours. Le schéma comprend une étape de construction de la représentation, une phase de réduction de représentation et un codeur entropique, sujet non approfondi dans ce document. Comme perspective, nous proposons aussi une nouvelle *description de régions par squelette propagé*, issue d'un algorithme rapide de calcul de l'ouverture.

Chapitre 5

Algorithmes rapides et réflexions fondamentales sur les opérations élémentaires de la morphologie mathématique

Introduction

La section 5.1 définit tout d'abord quelques opérations morphologiques utilisées par la suite. Nous décrivons ensuite, dans la section 5.2, les principales implémentations connues.

Ce chapitre développe alors *trois manières de réaliser des opérations morphologiques*.

La première famille d'algorithmes, détaillée dans la section 5.3, met en œuvre l'*histogramme local* de l'image. En chaque point du plan, le calcul de l'histogramme est réalisé sur le domaine de définition formé par le translaté de l'élément structurant. La lecture de l'histogramme fournit le minimum ou le maximum local, c'est-à-dire l'information utile pour une érosion ou une dilatation morphologique. Préférer le calcul de l'histogramme local à celui du simple minimum a pour objectif de réduire le nombre d'opérations nécessaires au calcul d'un opérateur en deux points voisins. L'originalité des développements tient tant à la généralisation de la méthode pour des éléments structurants quelconques ou non plans, menée en collaboration avec Hugues TALBOT [98, pages 41–51], qu'à l'exploitation d'un mécanisme sous-jacent qui permet la *réalisation d'opérations composées (ouverture, fermeture de l'ouvert, ...) en un seul balayage de l'image*.

À la section 5.4, nous abordons la seconde famille d'algorithmes dans le cadre plus restreint d'ouvertures d'images binaires et de labels, basée sur un principe de *propagation de l'élément structurant*.

Nous présentons les résultats numériques de mise en œuvre des algorithmes précités pour des images binaires ou en niveaux de gris et des images de labels.

À la lecture de la définition (cf. définition 5, page 29), on peut penser que le calcul de l'ouverture nécessite une érosion suivie d'une dilatation. Les deux familles d'algorithmes présentées s'inscrivent en faux contre cette affirmation. Dans un même ordre d'idée, la section 5.5 propose une *réflexion fondamentale sur la mise en cascade d'opérations morphologiques*. Nous l'illustrerons en traitant le cas d'opérations morphologiques avec un

élément structurant rectiligne.

5.1 Définitions

Les algorithmes basés sur l'histogramme local travaillent tant sur les ensembles que sur les images à teintes de gris ou de labels. Avant d'entreprendre les descriptions algorithmiques, nous présentons les quelques définitions d'opérateurs morphologiques qui s'y rattachent.

Opérations morphologiques sur des images en niveaux de gris. L'extension des outils de la morphologie mathématique à l'étude des images en niveaux de gris s'est réalisée longtemps après la définition des concepts de la théorie. La première étude complète des techniques adaptées à des images en niveaux de gris a été réalisée par SERRA [87].

Pour la morphologie mathématique en niveaux de gris, les opérations de base ne sont plus l'union et l'intersection mais le *supremum* $\vee$ et l'*infimum* $\wedge$. Dans le cas d'images à valeurs dans $\mathbb{Z}$, le supremum et l'infimum se ramènent respectivement au maximum et au minimum :

$$\forall h \in \mathbb{R}^2, \begin{cases} (f \vee g)(h) = \max(f(h), g(h)) \\ (f \wedge g)(h) = \min(f(h), g(h)) \end{cases} \quad (5.1)$$

Nous conserverons l'expression formelle générale parce qu'elle est à la fois plus répandue dans la littérature scientifique et plus commode pour l'étude de propriétés délicates comme la convergence de certains opérateurs.

Définition 24 (Érosion et dilatation) Soit $f(h)$ une fonction définie sur l'ensemble $\mathbb{R}^2$ ($h=(x,y)$) et B un élément structurant plan, c'est-à-dire un sous-ensemble borné de $\mathbb{R}^2$. L'érosion $f \ominus B$ (respectivement la dilatation $f \oplus B$) de la fonction f par B transforme f en une autre fonction définie par :

$$f \ominus B(h) = \bigwedge_{b \in B} f_{-b}(h) = \bigwedge_{b \in B} f(h + b) \quad (5.2)$$

$$f \oplus B(h) = \bigvee_{b \in B} f_b(h) = \bigvee_{b \in B} f(h - b) \quad (5.3)$$

Définition 25 (Ouverture et fermeture) L'ouverture $f \circ B$ et la fermeture $f \bullet B$ résultent de la mise en cascade de l'érosion et de la dilatation :

$$f \circ B = (f \ominus B) \oplus B \quad (5.4)$$

$$f \bullet B = (f \oplus B) \ominus B \quad (5.5)$$

Ouverture d'une image de labels. Pour rappel, un label est une valeur permettant de classer un élément dans un ensemble. Ainsi, par exemple, une image segmentée est constituée de régions ayant chacune un label donné. Aucune relation d'ordre ne lie les valeurs

d'une image de labels, car ils expriment l'appartenance à des régions. Voici la définition de l'ouverture de ce type d'images :

Définition 26 Soit L une image de labels composée de M régions ($L = \cup\{X_j \mid 1 \leq j \leq M\}$)¹. L'ouverture de L par l'élément structurant B , notée $L \circ B$, est l'image de labels, où $X_j \circ B$ est la région de label j . On attribue une valeur arbitraire aux points qui n'appartiennent à aucun ouvert.

5.2 Algorithmes connus

De nombreuses applications de la morphologie mathématique comme le filtrage, la granulométrie, ... nécessitent le calcul d'une opération appelée *ouverture morphologique* [33, 87]. Conformément à son interprétation, l'ouverture² d'un ensemble X par un élément structurant B est l'union de tous les translatés de B inclus dans X . Elle est aussi équivalente à la mise en cascade de deux opérations élémentaires appelées *érosion morphologique* et *dilatation morphologique*; c'est ainsi qu'on réalise l'ouverture dans la pratique.

Par une lecture sommaire de la définition, il semble que le calcul de l'ouverture soit directement proportionnel à la surface de l'élément structurant puisque, pour chaque point, il faut vérifier si tous les éléments du translaté sont inclus dans la figure. Ce n'est bien sûr pas de cette manière, au coût prohibitif, qu'il conviendra d'effectuer le traitement.

Étant donné l'importance de l'ouverture dans les traitements morphologiques et en particulier pour nos applications, il était indispensable de proposer des algorithmes rapides capables de réduire la complexité de l'algorithme issu de la définition. Différentes méthodes d'ouverture d'images binaires sont détaillées dans la littérature (certaines restent applicables à des images à teintes de gris) :

- la *décomposition 'linéaire'* des éléments structurants convexes de grande taille en une suite d'opérations avec des éléments de taille plus petite³. C'est de cette manière qu'a été implémentée l'opération d'ouverture dans le logiciel Xlim⁴ [78]. Lors des essais de comparaison d'efficacité, nous avons choisi cette méthode comme référence. Le temps de calcul de cette méthode évolue linéairement avec la taille de l'élément structurant. Il ne dépend pas du contenu de l'image;
- la *décomposition logarithmique* proposée par PECHT [71] et approfondie par VAN DEN BOOMGAARD [101]. Suggérée en prolongement de l'idée de la décomposition linéaire, la décomposition logarithmique supprime une partie⁵ des calculs redondants

1. X_j représente une région indicée par j et non le translaté de X par j . Cette confusion de notation, inévitable dans les trois derniers chapitres, est contournée en affectant les lettres i, j, k, l par priorité aux indices, $p, q, \dots$ étant réservées aux translations.

2. Pour rappel, nous considérons l'ouverture morphologique partout dans ce document. 'Ouverture' est donc un raccourci pour 'ouverture morphologique'.

3. C'est une mise en œuvre des propriétés 2.7 et 2.10 énoncées au point 2.2.7.

4. Xlim est un logiciel de traitement d'image développé initialement par Jean-François Rivest. Il sert actuellement de plate-forme commune pour le développement et l'intégration de programmes au Centre de morphologie mathématique. Ce logiciel a également été sélectionné pour les travaux de l'extension du projet européen RACE Morpheco.

5. La décomposition logarithmique de VAN DEN BOOMGAARD n'est pas optimale.

qui subsistent dans un calcul en chaîne. Soit à effectuer par exemple une dilatation par un carré 4×4 . Le carré s'obtient par deux dilatations successives d'un carré 2×2 par lui-même, mais on peut également l'obtenir par la dilatation d'un carré 2×2 par un carré 3×3 ne contenant que les quatre coins. Il en résulte une dépendance logarithmique du temps de calcul en fonction de la taille de l'élément structurant. Cette méthode, tout comme la précédente, ne conviendra que si l'élément structurant se décompose en éléments plus simples ;

- la *méthode des lacets* introduite par SCHMITT [85]. Elle consiste à former des lacets contenant les points de contour des différentes particules connexes de l'image, puis à les éroder et à les dilater par de simples règles de lecture lors du parcours des lacets, à condition que l'élément structurant ait une forme qui s'y prête. La complexité de cet algorithme est non plus proportionnelle à la surface de l'élément structurant mais au périmètre de l'ensemble X , ce qui constitue un avantage pour certaines applications. Par contre, cet algorithme ne fonctionne que dans le cadre très restreint des images binaires bidimensionnelles en trame hexagonale. Par analogie, VINCENT a imaginé un algorithme qui permet de dilater un ensemble par un élément structurant quelconque [111] ;
- les *algorithmes à base de file d'attente* [30, 93, 110]. Bien que les files aient été employées auparavant en traitement d'image, VINCENT a repris l'utilisation de files d'attente en morphologie mathématique. Le principe consiste à placer les pixels utiles dans une file que l'on examine et réduit au fur et à mesure. MEYER a introduit l'usage de files d'attente hiérarchiques [63] ;
- le récent *algorithme récursif* de VAN HERK [107]. Cette méthode réalise une érosion ou une dilatation numérique par un élément structurant rectiligne, horizontal ou vertical, au prix de seulement trois comparaisons par point. BREEN et SOILLE [7] l'ont généralisée à des directions quelconques.

La liste précédente n'est pas complète. Il faut par exemple encore citer la méthode de LAÿ [47, 48] qui calcule une érosion par seuillage de la fonction distance, les algorithmes adaptés aux réalisations matérielles, etc.

Quoique la décomposition logarithmique atténue la dépendance de la complexité avec la taille de l'élément structurant, la majeure partie des algorithmes d'ouverture dépendent de la taille de l'élément structurant : plus la taille est élevée, plus le temps de calcul augmente. C'est un sérieux inconvénient pour des traitements tels les filtres alternés séquentiels qui modifient une image par une famille d'ouvertures et de fermetures de taille croissante.

5.3 Érosion et ouverture d'images numériques et de labels par utilisation de l'histogramme local

Le premier souci des informaticiens désireux d'implémenter les opérations de morphologie mathématique était de trouver des algorithmes rapides pour des opérations compliquées, impossibles à réaliser par matériel. Rarement, les chercheurs se sont-ils intéressés à la réalisation des opérations élémentaires comme l'érosion ou l'ouverture. Pourtant, la

solution simple qui consiste à remplacer l'érosion par un carré de taille n par une succession d'érosions avec le carré de taille unitaire est loin d'être la meilleure. Jusqu'il y a peu, rien n'avait été proposé pour des éléments structurants quelconques, la recherche du minimum sur tout le domaine recouvert par l'élément, par application littérale de la définition, conduisant à des temps de calcul élevés. MEYER [64] a exposé les bases d'un algorithme axé sur les files d'attente, algorithme qu'a amélioré GRATIN dans sa thèse [30]. Comme nous le verrons dans un exemple, la complexité de l'algorithme reste de l'ordre de grandeur de celui qui revient à considérer l'un après l'autre chaque point de l'élément structurant mais le coefficient multiplicatif est plus petit, ce qui en fait une méthode nettement plus rapide.

L'idée exploitée par cet algorithme est la minimisation des opérations lors du déplacement d'un point à un autre. Cette voie a conduit HUANG *et al.* [36], CHAUDHURI [9] ou encore GIL et WERMAN [28] à calculer des filtres de rang et des filtres médians par glissement d'une fenêtre d'analyse. Néanmoins, ces auteurs se sont contentés d'utiliser un élément structurant carré et plan alors que le principe s'applique à des cas beaucoup plus généraux.

Obtenir le minimum d'un ensemble de valeurs, opération exigée lors d'une érosion, requiert une succession de tests. On ne pourra jamais descendre en dessous du chiffre d'un test au moins par valeur mais seulement s'approcher de cette limite théorique. Si seules certaines valeurs sont modifiées en passant d'un ensemble à un autre, on peut adopter deux attitudes différentes : (1) reprendre la série de tests systématiques, la solution de facilité, ou (2) analyser uniquement les valeurs sortantes et les valeurs entrantes avec un outil adéquat. Derrière cette remarque se cache le concept d'*histogramme local*, sorte de classement des valeurs définies sur le voisinage local d'un point.

Pour une érosion, il faut déterminer le minimum de l'ensemble constitué des points de l'image définis sur le translaté de l'élément structurant. Dans le cas d'un élément structurant carré, le passage d'un point à son voisin de droite signifie : ajouter les valeurs de la colonne de droite au domaine de définition et supprimer celles de la colonne de gauche. La figure 5.1 illustre le glissement d'une fenêtre et l'histogramme local défini sur le carré 4x4 entouré d'un trait continu. Malgré le déplacement, 12 valeurs continuent d'appartenir telles quelles à la fenêtre et donc à l'histogramme ; elles créent une espèce d'inertie, d'autant plus importante que l'élément structurant est grand, à la fois sur le jeu de valeurs mais surtout sur la valeur du minimum. C'est ce double avantage que nous avons voulu exploiter dans la première famille d'algorithmes.

Des réalisations adaptées à l'érosion et à l'ouverture. Nous allons détailler à présent notre algorithme pour une érosion et une ouverture. Pourquoi distinguer l'ouverture puisque, somme toute, elle résulte de la mise en cascade d'une érosion et d'une dilatation ? Simplement parce que le procédé d'ouverture que nous proposons peut se réaliser en un seul balayage de l'image, c'est-à-dire sans transiter par les opérations intermédiaires d'érosion et de dilatation.

Après la description des algorithmes, nous fournissons des résultats numériques étayés de commentaires.

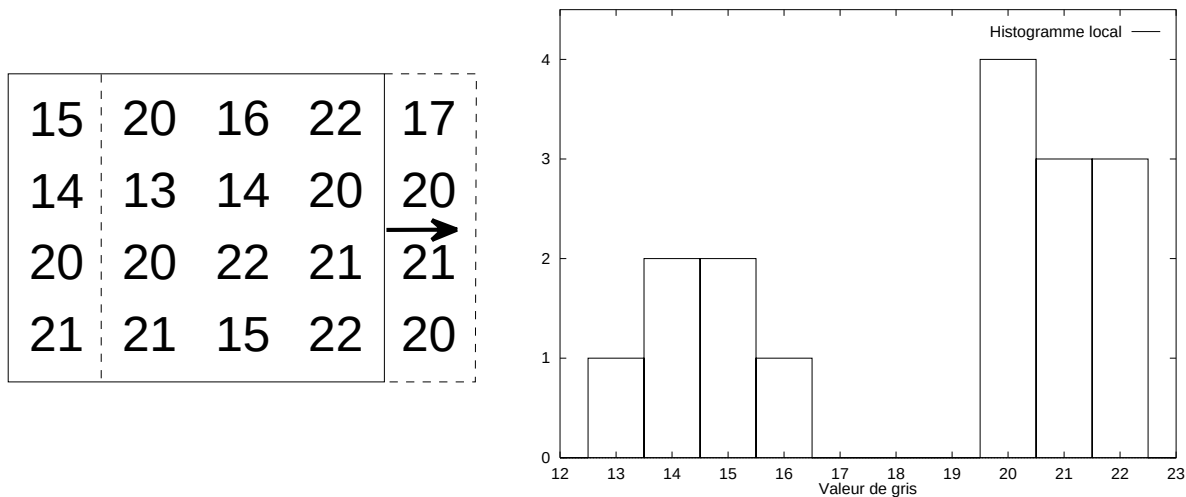


FIG. 5.1 - Glissement d'une fenêtre et histogramme local associé à la fenêtre initiale.

5.3.1 Calcul de l'érosion par histogramme local avec un élément structurant bidimensionnel de forme quelconque

On peut décomposer l'algorithme d'érosion par histogramme local en trois phases : une phase d'*initialisation* destinée à fournir un premier histogramme local, le *déplacement* répété de la fenêtre d'analyse et la *détermination du minimum* de chaque histogramme suivie d'une allocation subséquente de cette valeur dans l'image de sortie. Il est déjà intéressant d'annoncer que les deux premières phases des algorithmes utilisant l'histogramme décrits dans ce chapitre sont rigoureusement identiques, qu'il s'agisse d'une érosion ou d'une ouverture. Une réalisation matérielle pourra profiter de cette similitude et par surcroît, la troisième phase, seul élément différenciateur, n'est rien de plus qu'une interprétation de l'histogramme.

Pour en revenir à l'algorithme même, la phase d'initialisation procède à la construction d'un histogramme local complet, par exemple celui situé dans le coin supérieur gauche. Pour les points suivants, la fenêtre est déplacée de manière à minimiser le nombre total de changements à effectuer sur l'histogramme.

Pour tout déplacement, l'algorithme effectue trois ou quatre opérations suivant l'allure de l'histogramme :

1. Les nouvelles valeurs de gris sont insérées dans l'histogramme.
2. Tout en complétant l'histogramme, l'algorithme vérifie si une valeur ajoutée n'est pas inférieure au minimum précédent et remplace éventuellement le minimum précédent par la nouvelle valeur.
3. Les valeurs qui n'appartiennent plus au nouveau domaine de définition sont extraites de l'histogramme.
4. Parcours de l'histogramme jusqu'à une valeur de fréquence non nulle. Cette dernière étape n'est nécessaire que si aucune valeur ajoutée n'est inférieure ou égale au mi-

nimum précédent. De toute façon, c'est la fréquence du minimum qui indique s'il convient de parcourir l'histogramme ; il est inutile d'analyser les valeurs supprimées.

La question du sens de balayage de l'image. L'avantage de l'algorithme provient de la réduction du nombre de comparaisons nécessaires au calcul de chaque histogramme local. Il est dès lors souhaitable de parcourir l'image en évitant de devoir construire un histogramme complet autre que celui de la phase d'initialisation. Pour y parvenir, la solution imaginée consiste à balayer l'image dans la direction qui maximise le recouvrement.

Quid d'un élément structurant de forme arbitraire ? Il arrive que l'expérimentateur soit confronté au besoin de réaliser une opération morphologique où intervient un élément structurant de forme peu usuelle, s'il doit identifier un anneau par exemple. Le problème est encore plus aigu si l'élément structurant ne se décompose pas en éléments plus simples. Les algorithmes par histogramme développent une *approche en un passage*. La seule information dont il faut disposer est la position des points à ajouter et à retirer de l'histogramme lors d'un déplacement dans une direction quelconque de la trame. C'est pourquoi *avec l'histogramme, il y a moyen de réaliser une opération morphologique par n'importe quel élément structurant et ce dans toutes les trames*. Par parallélisme avec les files d'attente de VINCENT [110], on place les positions relatives des points à modifier –après une analyse de l'élément structurant qui ne présente aucune difficulté particulière– dans des listes qu'il suffit de lire lors d'un déplacement. Seul le nombre de points de frontière détermine la complexité de l'algorithme. En effet, ce sont ces derniers qu'il convient d'ajouter ou de retirer lors du glissement de la fenêtre. Dès lors, il est plus intéressant de traiter l'érosion par un segment vertical dans la direction verticale puisqu'il n'y a qu'un point avant et un point arrière, alors qu'un parcours horizontal nécessiterait le déplacement de tous les points de la fenêtre. TALBOT [98] a montré que le coût algorithmique d'une érosion par un segment était indépendant de sa longueur et ce pour toute direction. Nous avons pu voir qu'en fait, grâce à une bonne gestion des bords, la courbe du traitement par histogramme est légèrement décroissante à l'origine et présente un plateau pour des longueurs croissantes (cf. figure 5.20, page 135).

5.3.2 Réalisation de l'ouverture en un seul balayage

À condition de remplacer le minimum par le maximum, l'érosion devient une dilatation. Alors pour obtenir l'ouverture, il suffirait de mettre en cascade les deux algorithmes d'érosion et de dilatation par histogramme local.

Pourtant à partir de la valeur du minimum, il y a moyen de réaliser l'*ouverture en une seule passe*. La seule différence par rapport à l'algorithme d'érosion se situe dans l'allocation des valeurs après le calcul de l'histogramme : l'érosion affecte la valeur du minimum local en un seul point pour chaque domaine de définition alors que, lors d'une ouverture en une passe, toutes les valeurs du domaine peuvent être revues. Cela revient en quelque sorte à effectuer la dilatation durant l'érosion, sans prendre pour autant un élément structurant deux fois plus grand. L'algorithme est le suivant :

1. Tous les points compris dans la fenêtre et n'ayant pas été recouverts précédemment reçoivent la valeur du minimum local.

2. Concernant les autres points de la fenêtre, on attribue la valeur maximale entre la valeur retenue auparavant et le minimum local. Si la valeur du minimum a baissé, ce test n'est pas nécessaire pour tous les points appartenant à l'intersection de la fenêtre avec la fenêtre avant glissement.

Cet algorithme se simplifie s'il s'agit de réaliser l'ouverture par un élément structurant unidimensionnel, car on peut suivre l'évolution du minimum local. Ainsi, si le minimum local augmente lors du glissement de la fenêtre, tous les points de la fenêtre sont réaffectés. Dans le cas contraire, il suffit d'affecter la valeur du minimum au seul point ajouté. Ce mécanisme est illustré à la figure 5.2.

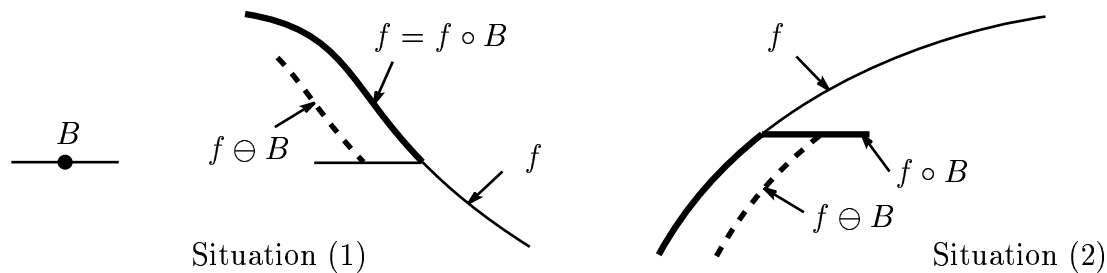


FIG. 5.2 - Illustration de deux situations distinctes rencontrées lors de la réalisation de l'ouverture par un segment en un seul passage. Soit à effectuer l'ouverture de la fonction f par l'élément structurant unidimensionnel B , dessiné à gauche. La rampe descendante de la situation (1) correspond à un minimum local décroissant, représenté par la courbe d'érosion $f \ominus B$. On peut voir que le calcul de l'ouvert n'affecte chaque fois que le point situé complètement à droite du segment, tandis que tous les points du plateau sont réaffectés pour le calcul de l'ouvert dans la situation (2).

Un passage unique évite le double calcul de l'histogramme local bien qu'en échange, le nombre d'affectations soit accru. Remarquons que la distinction entre l'érosion et l'ouverture se fait au niveau de l'affectation, autrement dit la phase de recherche du minimum est commune. Formellement, on peut le percevoir comme l'expression de la relation $f \ominus B = (f \circ B) \ominus B$.

5.3.3 Résultats

Pour estimer l'efficacité des algorithmes, nous avons mené une série de tests systématiques sur quelques images. Les conditions expérimentales, les résultats et les commentaires sont présentés dans cette section.

Description des conditions expérimentales générales. Les valeurs reprises dans les graphiques ont toutes été obtenues sur une station de travail NeXT monochrome 25 MHz (processeur 68040). Pour les comparaisons, nous avons intégré les nouveaux algorithmes dans le logiciel Xlim [78, 106]. La commande `gproff` du système d'exploitation Unix a fourni le temps CPU consommé par chaque essai ; les temps fournis ci-après n'incluent pas la consommation de temps pour les entrées-sorties. Étant entendu qu'il s'agit d'une

réalisation par logiciel, ce ne sont pas les valeurs absolues mais les écarts entre les courbes qui sont commentés. Pour cette première série de tests, l'image d'entrée est **Photograph** de taille 256x256 définie sur 8 bits, sauf mention explicite d'une autre image. Par ailleurs, la figure 5.3 reprend les images en niveaux de gris, utilisées dans ce chapitre.



Photograph (256x256).



Farm1 (1440x1152) dont la partie centrale constitue **Farm2** (748x552).

FIG. 5.3 - Images à teintes de gris utilisées dans les essais.

Étude de l'érosion par histogramme local : comparaison avec les méthodes de la littérature

Un premier graphique, tiré de TALBOT [98], cite les résultats de plusieurs algorithmes d'érosion par un carré, pour une image de taille 512x512 représentant des fibres. La figure 5.4 compare l'algorithme d'érosion par histogramme avec la méthode de GRATIN [30], ainsi qu'avec l'algorithme découlant d'une application immédiate de la définition, c'est-à-dire par recherche systématique du minimum sur tout l'élément structurant. Enfin, la décomposition linéaire [3x3], proposée dans Xlim et choisie parce que cette implémentation est courante, sert de point de référence bien qu'elle ne permette pas de réaliser l'érosion par un élément structurant de forme arbitraire. Ce graphique montre clairement que notre méthode, de complexité en $O(\sqrt{N})$ où N est l'aire de l'élément structurant, est nettement plus rapide que la méthode de GRATIN, de complexité en $O(N)$. Pour être plus précis, la complexité est linéaire en n où n représente le nombre de points de contours qu'il faut adapter à chaque glissement de l'histogramme. Ainsi, un losange et un carré qui circonscrit ce losange donnent des temps d'érosion identiques. En contrepartie, il existe des formes –celles de petite surface avec beaucoup de points de contour– pour lesquelles l'utilisation de notre algorithme est défavorable. Pourtant, la différence de complexité écarte définitivement l'algorithme de GRATIN pour de grands éléments structurants, indépendamment de la forme de l'élément. Notre méthode jouit aussi de l'avantage d'une simplicité de réalisation et est insensible au volume des extrema (cf. TALBOT [98]).

La figure 5.5 résulte d'un essai sur **Photograph**. Elle compare les temps de calcul de (1) l'opération d'érosion par décomposition linéaire d'un carré 3x3, avec (2) l'algorithme d'érosion par histogramme local par un carré, ainsi que (3) l'algorithme par histogramme

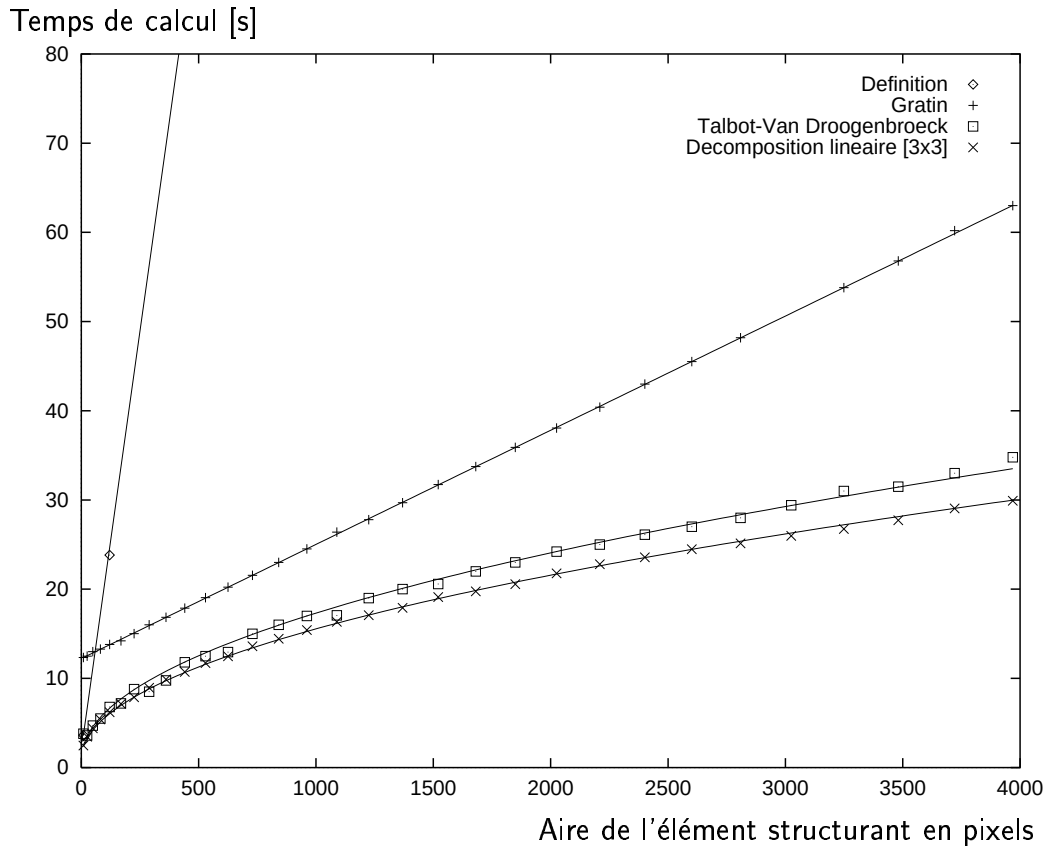


FIG. 5.4 - Comparaison d'algorithmes d'érosion (extrait de [98]).

après avoir décomposé le carré sous la forme de la dilatation d'un segment horizontal par un segment vertical⁶. Les essais montrent que cette dernière méthode devient intéressante à partir d'une largeur de carré de 9 pixels, mais il ne faut pas perdre de vue que seuls les temps de calcul de mise en œuvre de l'histogramme (courbe (2) de la figure 5.5) restent approximativement valables pour des éléments structurants de même taille, non décomposables.

La courbe de l'algorithme par histogramme suit une évolution linéaire ; c'est avant tout la hauteur de la colonne ajoutée qui fixe la complexité de l'algorithme, alors que les petites oscillations proviennent de l'analyse de l'histogramme. Si ces deux tracés sont des droites, leur pente diffère : pour de grands éléments structurants, le gain obtenu par l'histogramme devient intéressant, mettant ainsi en lumière l'effet d'inertie du minimum. Le traitement de l'histogramme alourdit malheureusement le calcul de l'érosion avec des petits éléments structurants. Dans la pratique, on pourra passer d'un algorithme à un autre suivant la taille de l'élément structurant.

Il est intéressant de remarquer que les positions relatives des courbes de décomposition linéaire et de l'algorithme par histogramme diffèrent en passant de la figure 5.4 à la figure 5.5. Bien que le code de l'algorithme par histogramme ayant produit la courbe de la figure 5.4 soit une généralisation de l'autre afin de permettre le traitement d'éléments structurants tridimensionnels, cet effet s'explique sans doute par une dépendance de l'al-

6. De fait, si $B = H \oplus V$, $f \oplus B = (f \oplus H) \oplus V$. Sachant que le temps de calcul de l'érosion par un segment ne dépend pas de la taille de l'élément dans une implémentation par histogramme [98], nous avons extrapolé les résultats à partir d'une seule expérience.

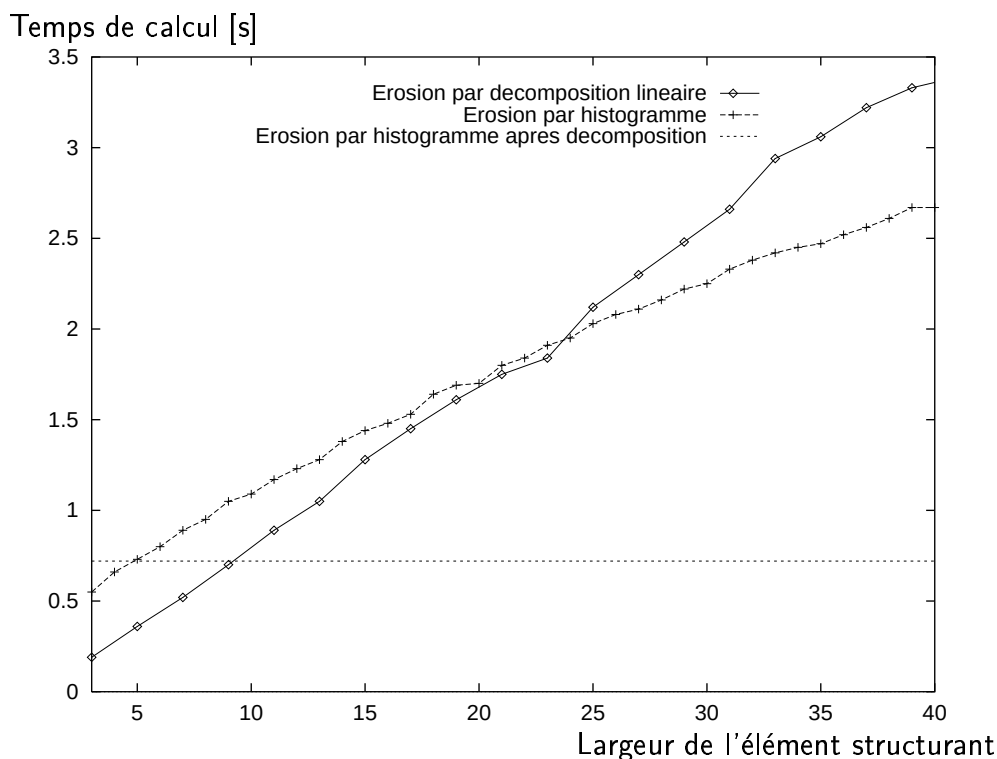


FIG. 5.5 - Comparaison d'algorithmes d'érosion morphologique.

gorithme en fonction du contenu de l'image.

Étude de l'ouverture par histogramme local en une seule passe

Les temps d'exécution d'algorithmes d'ouverture sont reportés à la figure 5.6. Puisque le calcul de l'histogramme est l'élément prépondérant de l'algorithme, les remarques formulées quant à la forme de la courbe d'une érosion par histogramme restent d'application. Pour cette même raison, il apparaît que le calcul en une seule passe améliore sensiblement les performances d'un algorithme qui appliquerait successivement une érosion et une dilata-tion. Pour s'en convaincre, il suffit de comparer les courbes de la figure 5.5 avec les valeurs indiquées à la figure 5.6. Doubler les valeurs de la figure 5.5 ne modifie pas la position du point d'intersection. À la figure 5.6 pourtant, le point d'intersection se situe aux alentours de 11 et non pas vers 22.

Tout semble indiquer qu'au vu de l'allure de la courbe de l'ouverture par histogramme, il est préférable de substituer cet algorithme à celui d'un algorithme linéaire, surtout qu'il traite tout élément structurant de cette taille pour un coût algorithmique semblable. C'est une nouvelle exploitation de l'inertie du minimum qui apporte ce résultat.

Ouverture d'images de labels

À défaut d'outil adapté, l'ouverture d'une image de labels peut s'obtenir en binarisant tour à tour chacune des régions puis en procédant à l'ouverture de chacune d'elles, ou alors en appliquant directement la définition. Aucune de ces deux solutions ne convient à une réalisation concrète en raison de leur coût élevé. L'histogramme offre à nouveau une solution

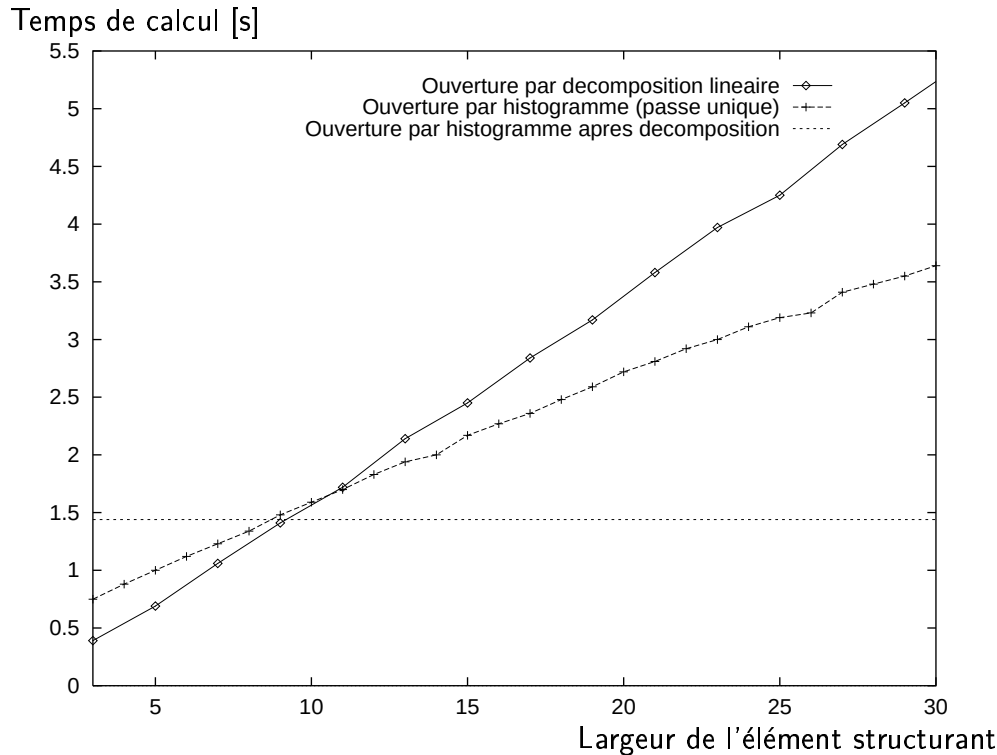


FIG. 5.6 - Comparaison d'algorithmes d'ouverture morphologique.

élégante au problème de l'ouverture d'une image de labels. Il suffit d'exiger que la fréquence du minimum soit égale à l'aire de l'élément structurant à la lecture de l'histogramme. En cas de réponse positive, tous les pixels couverts par le translaté de l'élément structurant reçoivent le label. Nous analyserons les performances de l'algorithme en même temps que celles de la famille d'algorithmes d'ouverture par propagation d'éléments structuraux (voir figures 5.14 et 5.15).

5.3.4 Discussion

Efficacité, simplicité et universalité. Du point de vue de l'*efficacité*, il apparaît que les algorithmes par histogramme local présentent une complexité qui croît linéairement avec la taille de l'élément structurant bidimensionnel; la complexité est même indépendante de la taille si l'élément structurant se décompose en segments. La comparaison des figures 5.5 et 5.6 montrent que, si le calcul de l'érosion par histogramme n'introduit aucune amélioration réelle quand B est quelconque, il n'en va pas de même pour le calcul de l'ouverture parce que celui-ci s'effectue en une seule passe.

La *simplicité* de la réalisation se déduit des éléments constitutifs de l'algorithme : balayage de l'image, constitution d'un histogramme complet, lecture de l'histogramme, ... Toutes ces parties se prêtent aisément à une réalisation par logiciel. De plus, les algorithmes d'érosion et d'ouverture implémentent *deux phases similaires* : initialisation et modification de l'histogramme. La différence intervient au niveau de l'interprétation de l'histogramme.

Opérations avec un élément structurant non plan. Jusqu'ici, nous avons supposé que l'élément structurant était plan. Cela impliquait que seuls les points de la frontière

avaient une incidence sur le calcul de l'histogramme. Il en va tout autrement lorsque l'élément structurant n'est pas plan puisque les valeurs de l'élément représentent des poids par lesquels doivent être pondérées les valeurs de la fonction. C'est donc par l'analyse du profil tridimensionnel de l'élément structurant que sont déterminés les changements à effectuer lors du glissement de l'histogramme ; tout le travail s'effectue dans l'espace et non plus uniquement dans le plan. Quoi qu'il en soit, l'implémentation de cet algorithme n'est pas plus ardue mais la complexité dépendra aussi de ce profil. Les essais que nous avons réalisés montrent que cette complexité évolue linéairement avec le nombre de points à la frontière du support de l'élément structurant 3D et le nombre de 'marches d'escalier' dans la troisième dimension. Cette conclusion signifie que si l'élément présente peu de variations, le coût algorithmique ne s'en fera pas sentir. L'algorithme est performant aussi pour le traitement des éléments structurants non plans.

Opérations partielles. Il est connu que les opérations morphologiques présentent une grande sensibilité au bruit : le résultat d'un filtrage dépend parfois d'un seul pixel. Les filtres de rang atténuent partiellement cette sensibilité, car ils classent les valeurs contenues dans une fenêtre et sélectionnent la valeur qui occupe une position déterminée [58]. Pour filtrer un léger bruit, il suffit alors de ne pas prendre le minimum, comme l'exigerait une érosion, mais la valeur classée après. Nous avons analysé ces opérations, reprises sous le vocable d'*opérations partielles*, pour le lissage des contours et le filtrage d'images à teintes de gris [103]. Le chapitre 6 détaille ces développements.

Pour obtenir le filtre de rang m , appelé érosion partielle de seuil m dans nos travaux en raison d'une approche similaire à celle des opérations morphologiques usuelles, on parcourt l'histogramme en partant du minimum jusqu'à la valeur dont la fréquence cumulée atteint ou dépasse $m + 1$. Le supplément de temps de calcul est faible et, à valeur de seuil constante, il diminue lorsque la taille de l'élément structurant croît. La figure 5.7 rassemble les courbes des temps de calcul de plusieurs érosions. La courbe qui file vers le haut provient d'une implémentation littérale de la définition d'érosion partielle. Plus remarquable est la proximité des 2^e et 3^e courbes : l'érosion partielle ne coûte pas beaucoup plus cher qu'une érosion morphologique et elle diminue la sensibilité au bruit. En outre, on peut aussi concevoir de calculer simultanément les érodés partiels avec plusieurs valeurs de seuil différentes. Ceci constitue la clef de voûte de notre nouveau concept de *polymorphisme*, développé dans le chapitre suivant.

5.4 Ouverture d'images binaires et de labels par propagation de l'élément structurant

Les principaux procédés rapides, capables de réaliser les opérations morphologiques élémentaires mettent l'accent sur l'élément structurant, que ce soit par une décomposition linéaire, une décomposition logarithmique, ou par la méthode des lacets ; le point de vue n'est que rarement placé du côté de l'ensemble à traiter. Or dans certaines situations, l'érosion d'une tache binaire de faible dimension par exemple, il serait plus avantageux de tenir compte d'une information propre au signal pour éviter toute une série de calculs inutiles. Le fil conducteur de l'algorithme décrit ci-après correspond à une démarche *adaptive* : une phase d'analyse détecte les traits caractéristiques du signal qui initialisent un

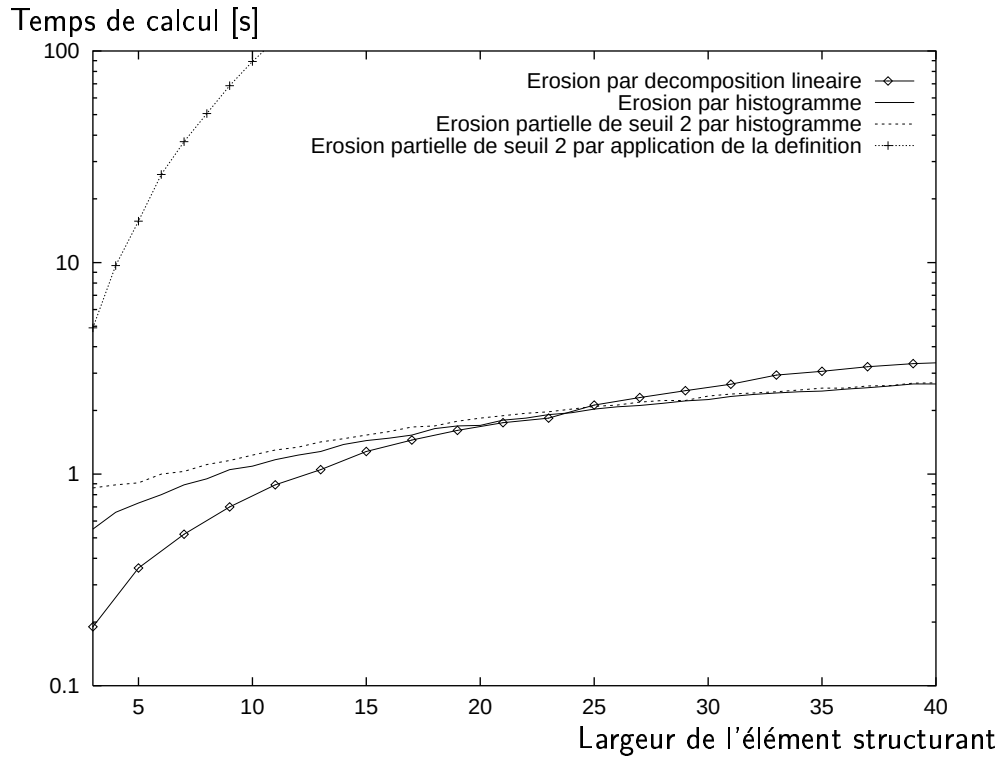


FIG. 5.7 - Étude du coût de l'érosion partielle (échelle semi-logarithmique).

processus de propagation contraint par la forme et la taille de l'élément structurant.

5.4.1 Le principe

Suite à la propriété d'anti-extensivité, l'ouverture de X par B est contenue dans X . Qui plus est, tout ouvert non vide contient inévitablement des points de la frontière intérieure de l'ensemble de départ. Ces points qui limitent l'extension spatiale de l'ouvert sont les *points d'ancrage*. Ils caractérisent l'ouvert d'une autre manière que les points centraux.

Nous avons imaginé de réaliser une ouverture en insérant l'élément structurant à l'intérieur de X puis en le déplaçant dans toutes les directions, à la manière d'un mécanisme de propagation d'onde. L'avantage appréciable qu'offre la propagation est l'utilisation d'une information concernant le 'passé'. Bien exploitée, elle permet l'économie de nombreux tests et conduit à approcher la limite théorique inférieure d'un seul test par point, surtout si X est un homothétique de B ou si le signal est traité par un filtre alterné séquentiel. L'algorithme repose sur la propriété suivante :

Propriété 27

$$\text{Si } B_p \subseteq X \circ B \text{ et } B_q \setminus B_p \subseteq X \text{ alors } B_q \subseteq X \circ B. \quad (5.6)$$

Deux éléments charpentent notre algorithme d'ouverture : la détection d'éléments structuraux B_p servant de germes à l'intérieur de X et la phase de propagation définie pour un jeu de directions q tel que $B_q \setminus B_p$ soit petit. Pour la facilité, nous décrivons les détails algorithmiques de l'ouverture d'une image binaire en 8-adjacence. Sur le plan du principe, rien

n'interdit des développements similaires en 4 ou 6-adjacence, ou même pour des éléments structurants quelconques.

5.4.2 Détails de l'algorithme

Détection d'un homothétique ancré de l'élément structurant

La première phase de l'algorithme est la recherche d'un élément structurant inclus dans X . Comme mentionné précédemment, l'ouvert touche la frontière de X en un point au moins. C'est donc parmi les points de la frontière intérieure 4-connexe de X que se trouvent les points d'ancrage potentiels. La figure 5.8 montre à droite les points d'ancrage candidats pour une figure X ; ils sont entourés d'une boîte.

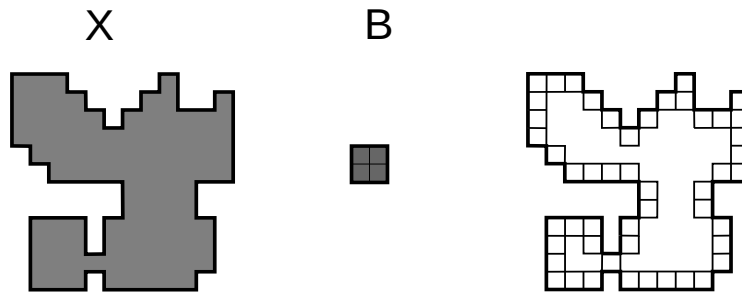


FIG. 5.8 - Un ensemble X , un élément structurant B et les points d'ancrage possibles.

Quelques observations permettent de *simplifier la recherche des points d'ancrage*. Primo, il est inutile de vouloir construire au départ une liste contenant tous les points de la frontière puisque la phase de propagation recouvrira une partie d'entre eux. Un simple balayage suffit ; à n'importe quel moment, seuls les points de contour non recouverts précédemment sont candidats. Secundo, un point n'est un véritable point d'ancrage que si la distance entre ses voisins les plus éloignés de gauche et de droite et si la hauteur entre ses voisins les plus éloignés du haut et du bas sont au moins égales à la taille de l'élément structurant. Cette condition nécessaire élimine d'emblée un grand nombre de points. Pour les points qui satisfont cette condition, nous essayons de placer le plus grand élément structurant dans l'ensemble X qui ait les dimensions requises. Signalons enfin que, suivant la manière dont s'effectuent la détection d'un homothétique et la propagation, on peut ne considérer que les points d'ancrage situés sur la frontière 'ouest' de X . En effet, en prenant $B_p \subseteq X$ le plus à gauche possible, B_p touche la frontière gauche de X , c'est-à-dire que $\exists x : x \in B_p \subseteq X$, avec un point $y \notin X$ comme voisin gauche de x . En pratique, limiter les points d'ancrage à ceux de la frontière 'ouest' amène un gain de temps de calcul qui peut atteindre 25%.

L'élément est ensuite propagé dans l'ensemble.

Propagation de l'ensemble homothétique ancré

La gestion d'un processus de propagation dans un ensemble est une tâche ardue. Il faut envisager différentes configurations, optimiser la propagation et éviter de passer sur des zones déjà traitées. Le risque d'instabilité due à des boucles est grand dans la mesure où

l'algorithme de propagation utilise des appels récursifs à des fonctions et que le nombre de fronts de propagation possibles croît rapidement. Si l'on veut supprimer les appels récursifs, il faut gérer soi-même la pile des appels de fonction, ce qui complique encore la mise en œuvre.

La difficulté principale du processus de propagation est celle de transmettre des informations concernant l'évolution spatiale du front de propagation. Parfois, cette information est implicite. Par exemple, étant donné Y un homothétique de l'élément structurant inclus dans X , si les points de droite de Y appartiennent tous à X alors ils appartiennent à l'ouvert. Souvent, la question est délicate comme dans un changement de direction. Au lieu d'opter pour une description complète de la phase de propagation qui nous mènerait dans des discussions relatives à un langage de programmation, nous dégageons quelques axes directeurs du processus. Nous illustrons ensuite l'algorithme complet par un exemple.

On peut décomposer la propagation en deux étapes :

– *Initialisation du processus de propagation.*

Puisque l'homothétique détecté est le plus grand ensemble qui s'insère dans X au point d'ancrage considéré, il contient obligatoirement un point de contour intérieur sur chacun des côtés. En d'autres termes, il est impossible d'ajouter une colonne ou une ligne entière à cet ensemble. Le front de propagation ne peut donc que glisser entre ces points. Parfois, la largeur n'est pas suffisante pour remplir la condition de taille et la propagation s'arrête. Autrement dit, un processus de propagation est initialisé dans certaines directions en favorisant le traitement des directions diagonales puisqu'elles équivalent à des propagations horizontale et verticale simultanées. La figure 5.9 reprend les huit directions de propagation autorisées pour les fronts, hachurés en gris clair.

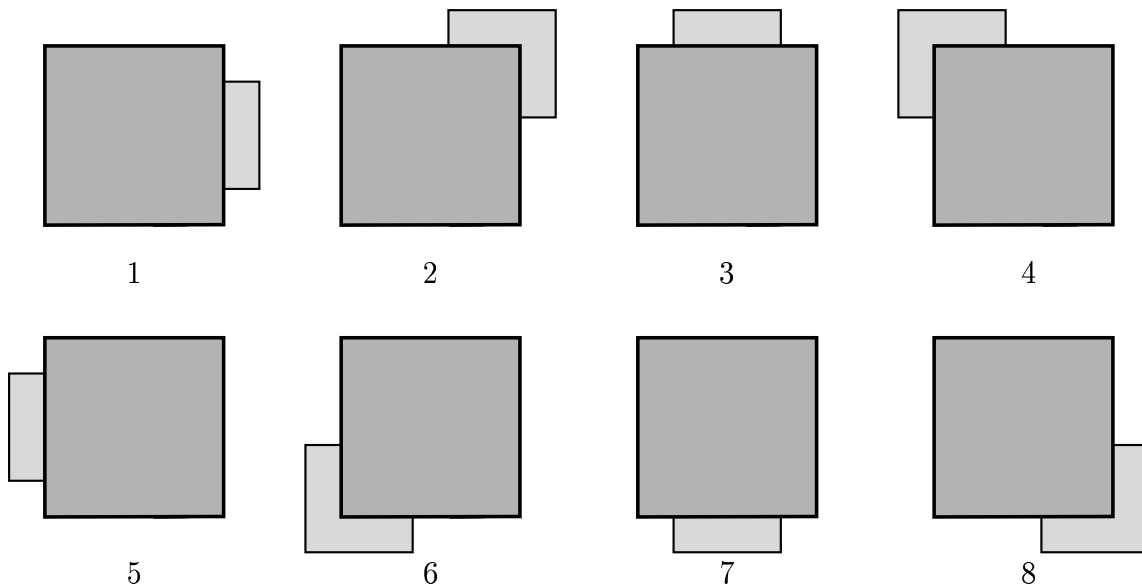


FIG. 5.9 - Exemple d'un jeu de directions de propagation. Ces directions sont numérotées de 1 à 8.

– *Propagation dans une direction avec des changements possibles de direction.*

Cette phase constitue véritablement le nœud de la réalisation par logiciel. Comme

une propagation en diagonale se décompose toujours en une composante horizontale et verticale, tout arrêt de propagation dans une direction se traduit par des essais de propagation dans les deux directions conjuguées. Par contre, l'arrêt d'une propagation horizontale ne signifie pas que le processus de propagation soit terminé. En effet, il est fort possible que, lors d'une propagation, on ait découvert qu'il était possible de propager dans la direction perpendiculaire à partir de la trace que laisse le front. Illustrons le mécanisme sur un exemple simple : soit X la lettre 'O' schématisée par quatre segments perpendiculaires et B un pixel unique. Le balayage de l'image s'interrompt dès que l'algorithme a atteint le coin supérieur gauche de la lettre. L'ensemble homothétique associé à ce point d'ancrage est le trait vertical gauche. La propagation est initialisée vers la droite, en haut et en bas, c'est-à-dire dans les deux traits horizontaux. Pourtant à la fin de l'une d'elles, le front d'onde changera de direction pour remplir le trait vertical de droite. La seconde propagation vers la droite doit détecter que la propagation dans le trait vertical de droite n'apporte rien de neuf. Mieux, une seule propagation vers la droite se poursuit en bouclant, remplit le second trait horizontal et annihile la tentative initiale de la seconde propagation horizontale. Cette exploitation maximale de toutes les directions possibles de propagation est la condition *sine qua non* pour le succès de l'algorithme. Mal programmée, la phase de propagation entraîne des oublis de recouvrement de zones intérieures et occasionne des incréments non négligeables du temps de calcul.

5.4.3 Exemple complet d'ouverture par propagation de l'homothétique d'un élément structurant

La figure 5.10 reprend plusieurs phases de l'ouverture de l'ensemble X , dessiné à la figure 5.8, par un carré 2×2 . À chaque instant, les surfaces en gris foncé représentent les parties déjà recouvertes alors que le trait en gris clair indique le front d'onde ; nous l'avons représenté avec une hauteur d'un demi-pixel pour accentuer le mouvement de propagation. L'appel récursif des fonctions est géré de manière à poursuivre la propagation dans la direction initiale, sauf si une propagation dans une direction diagonale devient possible.

Voici commentées dans l'ordre, les différentes images de la figure 5.10 associées à des phases intermédiaires du processus de propagation. À chaque étape, nous mentionnons l'état de la pile contenant les fronts de propagation candidats, notés chacun par leur direction suivant les conventions de la figure 5.9 et la phase d'apparition du front en italique et entre parenthèses.

Initialisation : détection d'un homothétique ancré.

Cette première phase initialise le processus de propagation. Il apparaît qu'une propagation en diagonale est licite. Même si d'autres propagations (horizontale ou verticale) sont possibles, nous commençons par la direction diagonale (direction 8).

Phase 1 : début de la propagation en diagonale.

Tous les points recouverts par le front d'onde sont ajoutés à l'ouvert. C'est pourquoi, le front s'élargit éventuellement durant la phase de propagation en diagonale tout en demeurant confiné dans le demi-plan que forment le point le plus à gauche et le plus en haut.

État de la pile : vide.

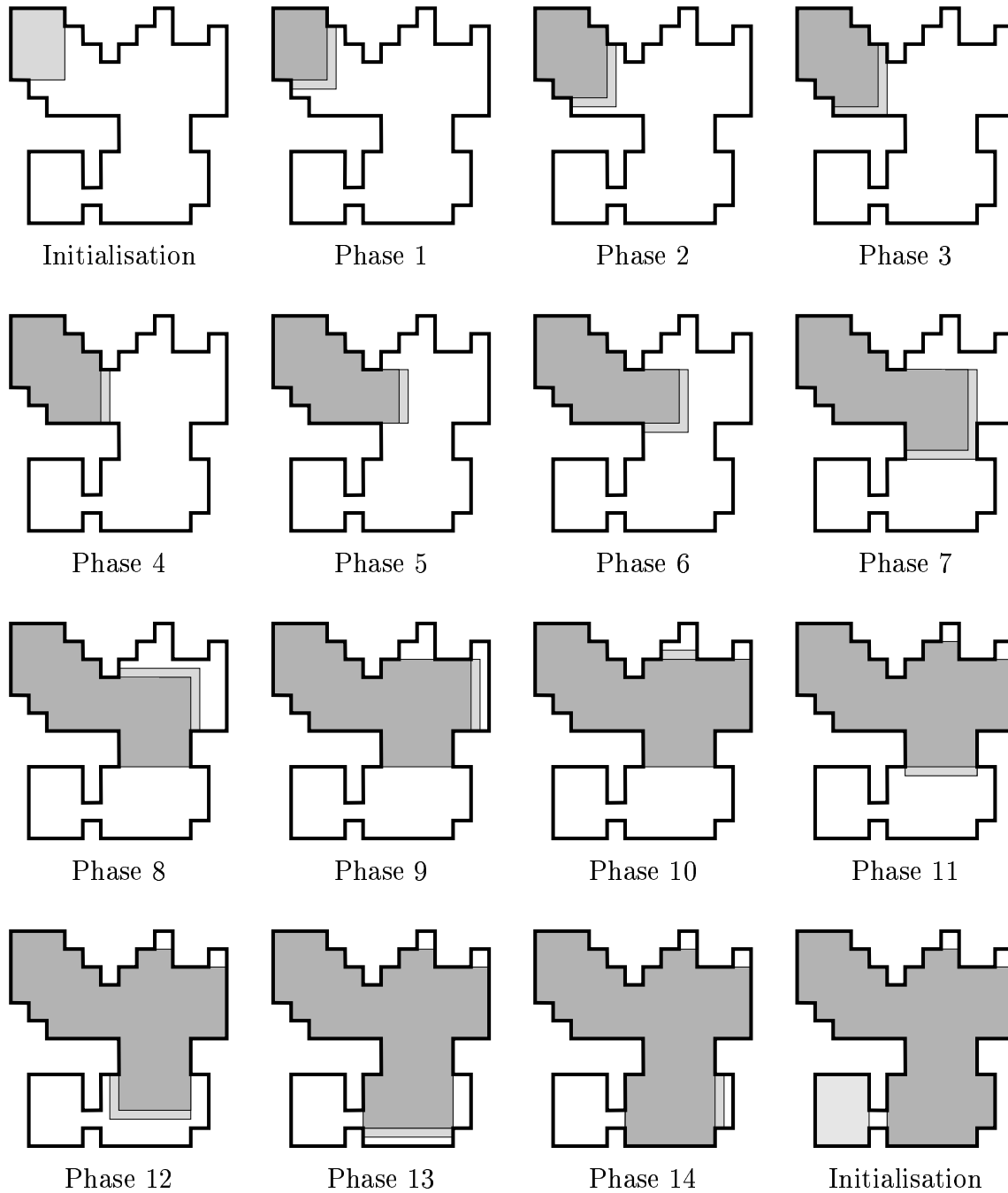


FIG. 5.10 - Exemple d'ouverture par propagation d'un élément structurant. Pour une meilleure compréhension, l'accroissement du front d'onde ne vaut que la moitié d'un pixel.

Phase 2: poursuite de la propagation en diagonale avec un léger rétrécissement du front d'onde.

Pour respecter la géométrie de X , les composantes horizontale et verticale du front ont été réduites.

État de la pile: vide.

Phase 3: dernière propagation en diagonale.

Au-delà de ce front, il faudra décomposer le front en une composante horizontale et verticale.

État de la pile: vide.

Phase 4: propagation horizontale.

Comme la figure ne peut s'étendre verticalement, la propagation se poursuit dans la direction horizontale a priori jusqu'à l'arrêt.

État de la pile: vide.

Phase 5: poursuite de la propagation horizontale.

Le front se propage horizontalement et l'algorithme comptabilise, à chaque déplacement, le nombre de points laissés à droite et à gauche du front. Ces valeurs permettront le changement de direction en temps opportun.

État de la pile: vide.

Phase 6: changement de direction.

Le front a détecté deux points de X à sa droite et à sa gauche. C'est donc une indication pour un éventuel changement de direction vers la droite ou vers la gauche. Comme ici le nombre de points appartenant à X laissés à droite est égal à la largeur de B et que la propagation peut se poursuivre horizontalement, le front s'accroît en diagonale. La direction et la position du front de la diagonale montante (direction 2) sont placés dans une pile. Par une exploitation de l'appel récursif des fonctions, la gestion de la pile est confiée à l'ordinateur. Nous pourrions gérer la pile de sorte à choisir le front le plus large de la pile à chaque étape, mais il est probable que le gain apporté soit complètement négligeable en échange d'une complexité algorithmique accrue.

État de la pile: $2(6)$.

Phase 7: propagation en diagonale.

La propagation en diagonale s'achève et le front se découpe en deux composantes, toutes deux ajoutées à la pile des fronts en privilégiant la propagation diagonale.

État de la pile: $2(7)$, $2(6)$, $7(7)$.

Phase 8: analyse de la pile et propagation.

On retire le premier front de la pile, c'est-à-dire celui de la propagation diagonale. La propagation s'ensuit.

État de la pile: $2(6)$, $7(7)$.

Phase 9: fin de la propagation diagonale avec apparition de nouveaux fronts et analyse de la pile des fronts.

Deux nouveaux fronts apparaissent au terme de la phase de propagation en diagonale. Ils sont empilés en plaçant d'abord le plus étroit et en respectant à nouveau la priorité

accordée à la direction diagonale.

État de la pile : 2(6), 1(9), 3(9), 7(7).

L'algorithme choisit ensuite le front 2(6). Mais tous les points ayant été traités par des propagations antérieures, ce front n'est plus propagé. L'algorithme dépile et étend alors un autre front d'onde.

État de la pile : 3(9), 7(7).

Phase 10 : propagation d'un front de la pile.

Le front précédent s'est propagé vers la droite mais comme aucun nouveau front n'a été créé, il faut dépiler les coordonnées d'un front et poursuivre la propagation vers le haut.

État de la pile : 7(7).

Phase 11 : propagation d'un front de la pile.

Même mécanisme qu'à l'étape 10. Le processus de propagation reste actif tant que la pile n'est pas vide et que le front ne se déplace plus.

État de la pile : vide.

Phase 12 : changement de direction.

À la fin de cette propagation en diagonale, on se retrouve dans une situation comparable à celle de la phase 7. La propagation continue avec le front le plus large, l'autre est empilé.

État de la pile : 1(12).

Phase 13 : poursuite de la propagation vers le bas.

La propagation diagonale étant terminée, un nouveau front se déplace vers le bas.

État de la pile : 1(12).

Phase 14 : propagation d'un front de la pile.

Le dernier changement de direction ne permettra plus de réactiver le processus de propagation parce qu'aucun front nouveau n'est détecté. L'algorithme cherche donc un autre homothétique de B .

État de la pile : vide.

Initialisation.

Un nouvel élément ancré réinitialise le processus de propagation. Dans le cas présent, la propagation n'ajoute aucun élément à l'ouvert de X .

5.4.4 Résultats expérimentaux

Les conditions expérimentales (insertion dans Xlim, ordinateur de travail, ...) sont inchangées. Par contre, le champ d'application est quelque peu différent parce que nous discernons à cette fois le traitement d'images binaires et d'images de labels de difficulté variable.

Ouverture binaire

Les figures 5.11 et 5.12 reproduisent sous forme de graphique les temps de l'ouverture par propagation d'élément structurant pour les images d'entrée respectives que sont **Farm2**

(image de taille 748x552) binarisée par seuillage à 137 et **Photograph** binarisée par seuillage à 128. La première image contient de nombreux points de contours; il s'agit dès lors de conditions assez défavorables à l'utilisation d'un algorithme par propagation. Comme témoin de la difficulté, le nombre d'éléments structurants ayant servi de germe⁷, lors de l'ouverture par un carré 5x5 s'élevait à 15301. La seconde image convient davantage au nouvel algorithme parce que les formes sont plus convexes; 25 germes ont suffi pour mener à bien l'ouverture par le même carré.

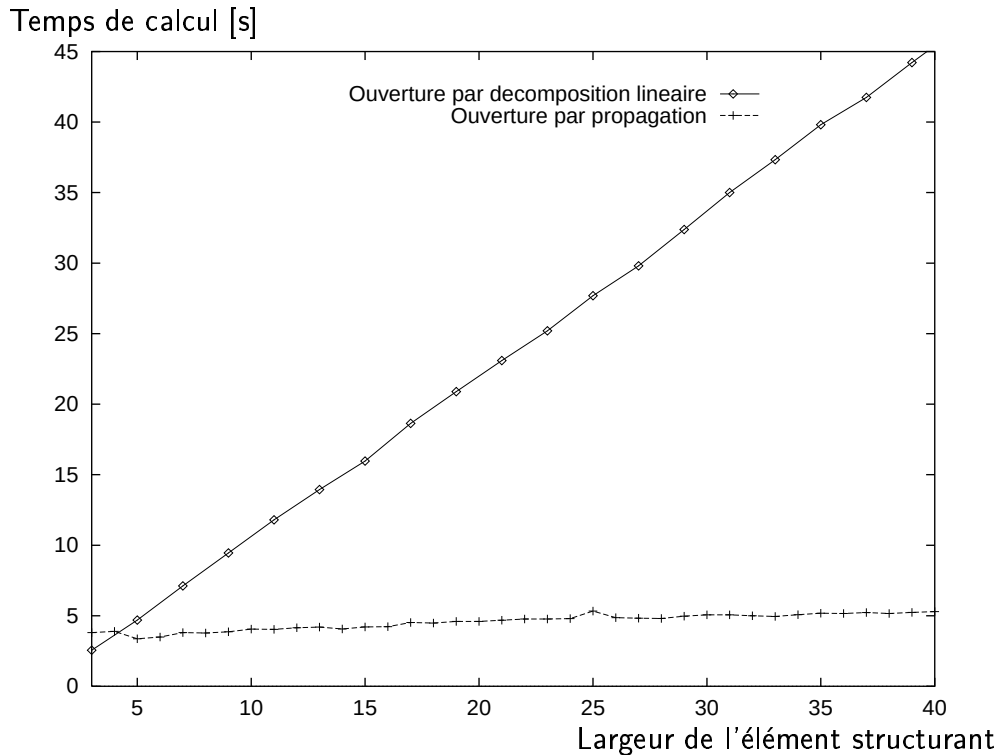


FIG. 5.11 - Temps de calcul de deux algorithmes d'ouverture morphologique par un carré sur une image binaire complexe et de grande taille.

Les temps de calcul requis pour un algorithme qui agit par propagation d'élément structurant sont sans commune mesure avec les temps requis pour d'autres algorithmes. Hormis le cas extrême des très petites tailles de B et d'une image compliquée, le traitement par propagation accélère la réalisation de l'ouverture binaire. Le temps de calcul est même toujours inférieur à celui de l'algorithme par décomposition linéaire de Xlim pour des images binaires simples. D'autre part, le coût plafonne à une valeur constante ce qui en fait un *algorithme de complexité indépendante de la taille de l'élément structurant*. Cette indépendance provient de la manière par laquelle le front s'étend. Par exemple, l'ajout d'une ligne a requis un test par point ajouté sans qu'intervienne nullement la taille du front. Le nombre total de tests est proche de cette limite théorique inférieure.

Les essais rapportés ci-dessus indiquent que le temps de calcul dépend très peu du contenu de l'image, qu'elle soit composée de régions connexes ou non. L'explication de ce phénomène est une équivalence, au niveau du temps de calcul, entre la recherche d'un homothétique et une phase de propagation qui remplirait la surface de cet homothétique.

7. Nous appelons 'germe' un élément structurant homothétique à B ancré en un point de la frontière de X .

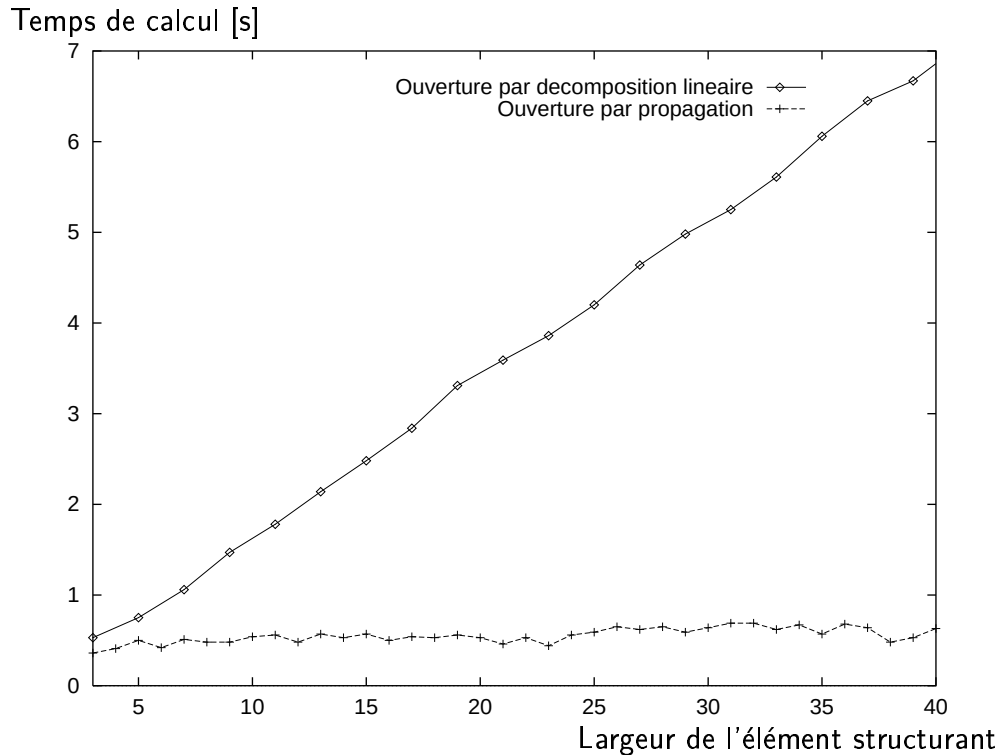


FIG. 5.12 - Temps de calcul de deux algorithmes d'ouverture morphologique par un carré sur une image binaire simple.

Ouverture d'une image de labels

Dans la foulée de l'essai précédent, nous avons appliqué l'algorithme à des images de labels. En traitant une région comme une surface ayant un label particulier, un algorithme par propagation d'élément cherche l'emplacement d'un homothétique de l'élément structurant de label homogène. Le label est alors la valeur propagée. Les figures 5.14 et 5.15 rapportent les temps mesurés pour l'ouverture d'une image de labels simple, composée de 35 régions (figure 5.14) et d'une image compliquée (figure 5.15). Ces images de référence sont illustrées à la figure 5.13. La figure 5.14 fournit aussi le temps utilisé pour l'ouverture d'une image binaire de même taille –rappelons que cet algorithme ne dépend pas du contenu de l'image– et le temps que demande l'algorithme qui binarise tour à tour chacune des 35 régions. Afin de compléter la comparaison, nous avons également indiqué le coût d'une implémentation par histogramme local.

En comparant les courbes, on peut affirmer que les nouveaux algorithmes effectuent le travail sur toute l'image de labels pour le coût de l'ouverture d'une image binaire. Pour une taille élevée de l'élément structurant, le coût de l'ouverture par propagation est même largement inférieur au traitement d'une seule région par un algorithme classique. En outre, les courbes continuent d'être indépendantes de la taille de l'élément structurant et montrent que l'ouverture par histogramme est une alternative simple, quoique légèrement moins performante, pour l'ouverture d'images de labels.

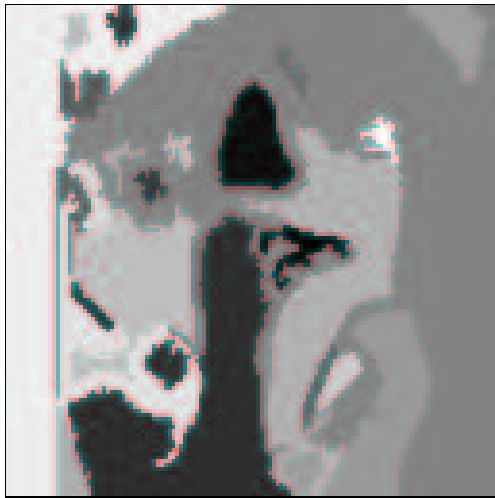


Image de labels simple (128x128).

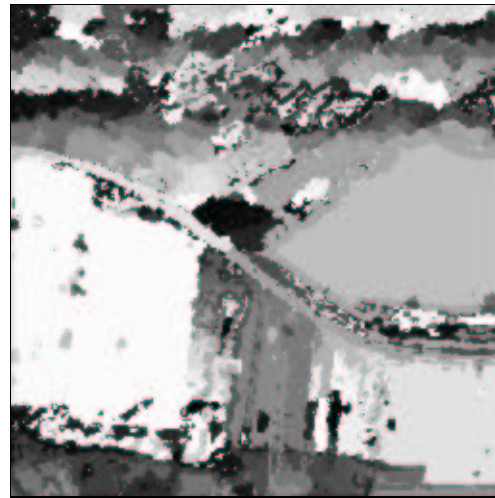


Image de labels complexe (256x256).

FIG. 5.13 - Images de labels utilisées dans les essais.

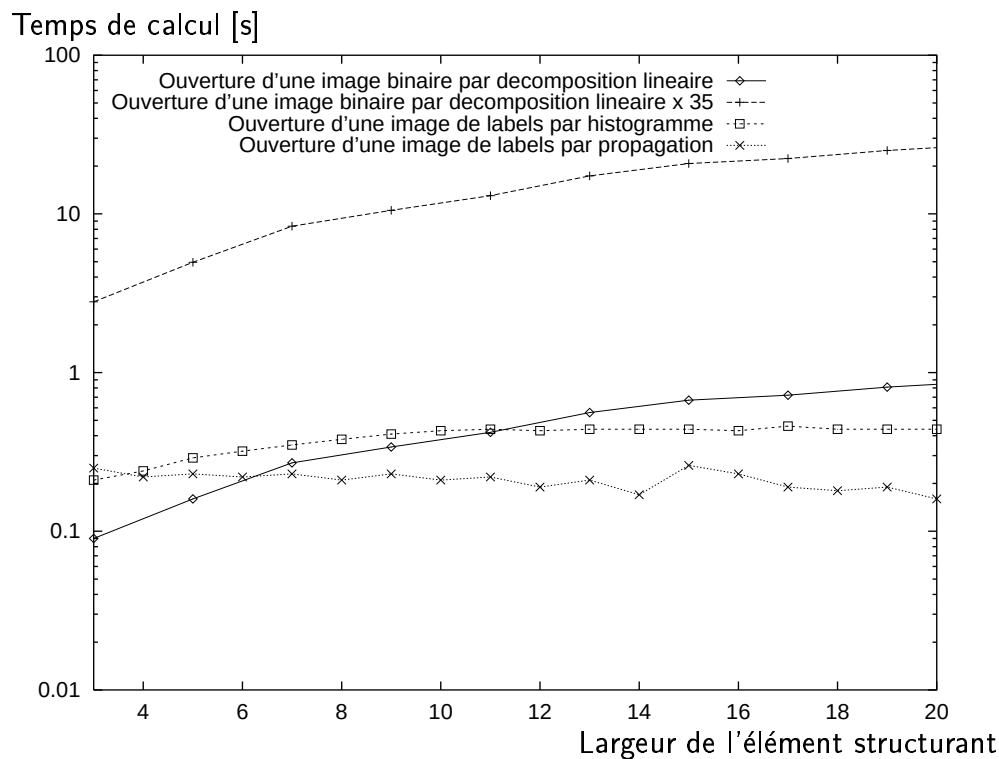


FIG. 5.14 - Temps algorithmiques de plusieurs procédés d'ouverture d'une image de labels simple par un carré (échelle semi-logarithmique).

5.5 À propos de l'implémentation d'opérateurs morphologiques composés

Il apparaît comme inéluctable, par simple lecture des définitions, que l'ouverture et la fermeture soient obtenues par la mise en série d'une érosion et d'une dilatation. Non seulement, comme nous l'avons montré dans les deux sections précédentes, cette remarque

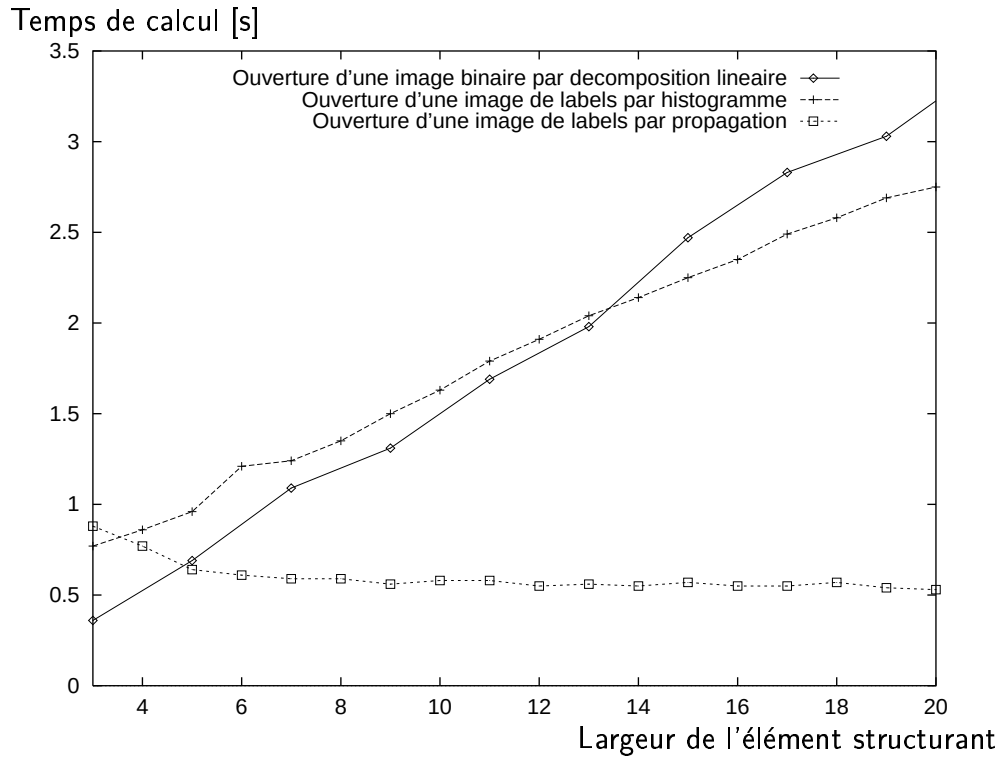


FIG. 5.15 - Temps algorithmiques de plusieurs procédés d'ouverture d'une image de labels complexe par un carré.

est fausse mais, avec des éléments structurants linéaires, il est même possible d'obtenir des opérations plus complexes encore (fermeture de l'ouvert, ...) pour un surcoût inférieur à 40% par rapport au calcul d'une simple érosion. D'où vient une telle économie? Simplement de l'utilisation du fait que la fonction est traitée dans une étape antérieure. Autrement dit, une fonction érodée ou ouverte possède des caractéristiques telles qu'elles simplifient les opérations futures.

Cette section propose une réflexion fondamentale sur la réalisation d'opérations morphologiques composées. Nous établissons tout d'abord une relation générale qui lie, sous la forme d'une correspondance, les fonctions traitées par érosion ou dilatation, et les éléments structurants associés. Des propriétés intéressantes découlent de particularisations de la propriété générale. Après les avoir énoncées, nous examinons la contrainte sur les éléments structurants et fournissons une condition suffisante pour qu'elle soit respectée. La dernière partie de cette section illustre ces principes théoriques par le biais d'implémentations d'opérations morphologiques avec un segment rectiligne.

5.5.1 Développements théoriques

Le théorème suivant lie les éléments structurants qui interviennent dans des opérations d'érosion et de dilatation.

Théorème 28 Pour toute fonction f ,

$$(f \ominus A) \oplus B \leq (f \oplus C) \ominus D \quad (5.7)$$

si et seulement si

$$A \oplus C \supseteq B \oplus D \quad (5.8)$$

Démonstration

Nous développons le théorème dans un cadre algébrique (cf. les travaux d'HEIJMANS et de RONSE [34]). La démonstration s'effectue en deux parties qui valident l'équivalence.

- (1) $\forall f, (f \ominus A) \oplus B \leq (f \oplus C) \ominus D \Rightarrow B \oplus D \subseteq A \oplus C$:
 Soit o le neutre pour l'addition. $A = o \oplus A \subseteq A$ et donc $o \subseteq A \ominus A$ par adjonction (relation 2.11, page 32). En remplaçant f par A dans l'inégalité 5.7, on a $B = o \oplus B \subseteq (A \ominus A) \oplus B \subseteq (A \oplus C) \ominus D$. Enfin, par adjonction, on obtient $B \oplus D \subseteq A \oplus C$.
- (2) $\forall f, (f \ominus A) \oplus B \leq (f \oplus C) \ominus D \Leftarrow B \oplus D \subseteq A \oplus C$:
 $B \oplus D \subseteq A \oplus C$ donne $(f \ominus A) \oplus B \oplus D \leq (f \ominus A) \oplus A \oplus C$. Or, $(f \ominus A) \oplus A \oplus C = (f \circ A) \oplus C \leq f \oplus C$. Ainsi, $(f \ominus A) \oplus B \oplus D \leq f \oplus C$, ou encore $(f \ominus A) \oplus B \leq (f \oplus C) \ominus D$ par adjonction.

Grâce à la généralité du cadre de travail, le principe de la démonstration reste valable pour des éléments structurants non plans . ■

Corollaire 29 Pour toute fonction f

$$f \circ C \leq (f \oplus A) \ominus B \quad (5.9)$$

$$(f \ominus A) \oplus B \leq f \bullet C \quad (5.10)$$

si et seulement si

$$A \oplus C \supseteq B \oplus C \quad (5.11)$$

Démonstration

La relation 5.10 est l'énoncé du théorème 28 dans le cas particulier $C = D$. Pour obtenir l'inégalité 5.9, il suffit d'inverser les ensembles dans les membres de la relation 5.8 et d'adapter 5.7 en prenant $C = D$. ■

Les exemples de filtres morphologiques abondent dans la littérature. Parmi les plus utilisés, il y a les *filtres alternés séquentiels* [88, chapitre 10], introduits par STERNBERG [95], et qui consistent à alterner des ouvertures et des fermetures de taille croissante. Le nombre d'opérations intervenant dans la réalisation d'un tel filtre est malheureusement très élevé et, dans la mesure du possible, il faut le réduire. SERRA [88] a développé une approximation pour certains filtres alternés séquentiels, dans laquelle on élimine une opération sur deux. Nous proposons plus loin une simplification du même ordre de grandeur, mais qui a le mérite de fournir le résultat correct. Ce résultat s'appuie sur le théorème suivant.

Théorème 30 Si $A \oplus C \supseteq B \oplus C$ et $A \subseteq B$ (ce qui implique $A \oplus C = B \oplus C$) alors

$$[(f \circ C) \oplus A] \ominus B = (f \circ C) \bullet B \quad (5.12)$$

$$[(f \bullet C) \ominus A] \oplus B = (f \bullet C) \circ B \quad (5.13)$$

Démonstration

Relation 5.12:

Si $A \subseteq B$ alors $(f \circ C) \oplus A \leq (f \circ C) \oplus B$ et, par croissance de l'érosion, $[(f \circ C) \oplus A] \ominus B \leq (f \circ C) \bullet B$.

D'autre part, la relation 5.9 précise que $(f \oplus A) \ominus B \geq f \circ C$. Étant donné que ce résultat est valable pour toute fonction f , il l'est aussi pour $f \circ C$: $[(f \circ C) \oplus A] \ominus B \geq (f \circ C) \circ C = f \circ C$. Mais comme la fermeture est un opérateur croissant: $[(f \circ C) \oplus A] \ominus B \bullet B \geq (f \circ C) \bullet B$. En plus, pour toute fonction g , $(g \ominus B) \bullet B = g \ominus B$ –il s'agit de la relation 2.12 (page 32) généralisée à une fonction–, et donc $[(f \circ C) \oplus A] \ominus B \geq (f \circ C) \bullet B$. La relation 5.12 est ainsi démontrée.

Relation 5.13: c'est le dual par complémentation de la relation précédente. ■

Comment choisir les éléments structurants?

Il va de soi que le théorème précédent ne présente de l'intérêt que si les conditions $A \oplus C \supseteq B \oplus C$ et $A \subseteq B$ sont réalisables. D'autre part, on voudrait disposer de la relation $[(f \circ B) \oplus A] \ominus B = (f \circ B) \bullet B$ pour réaliser la fermeture de l'ouvert à moindre coût, c'est-à-dire en prenant un élément A nettement plus petit que B . La question mérite que nous traitions deux situations: la première impose $B = C$, au contraire de la seconde.

Situation 1 : $B = C$

Quand $B = C$, les hypothèses du théorème 30 deviennent $A \oplus B \supseteq B \oplus B$ et $A \subseteq B$. Il en résulte une égalité entre les ensembles $A \oplus B$ et $B \oplus B$, reprise par la propriété suivante.

Propriété 31 (Condition de réduction) *Si $A \subseteq B$ alors*

$$A \oplus B \supseteq B \oplus B \Leftrightarrow A \oplus B = B \oplus B \quad (5.14)$$

L'égalité

$$A \oplus B = B \oplus B \quad (5.15)$$

est appelée la condition de réduction⁸.

Le détermination d'un ensemble A d'aire minimale à partir de B s'apparente au problème de la décomposition logarithmique. PECHT [71] a établi que, pour tout ensemble B de $\mathbb{Z}^2$, il existe un entier m tel que, si $m' \geq m$,

$$\underbrace{B \oplus B \dots \oplus B}_{m' \text{ dilatations}} \oplus B = \underbrace{B \oplus B \dots \oplus B}_{m' \text{ dilatations}} \oplus E(B) \quad (5.16)$$

où $E(B)$ représente le plus petit sous-ensemble de B ayant la même enveloppe convexe. Déterminer la valeur de m pose malheureusement quelques difficultés lorsque l'élément structurant n'est ni symétrique, ni convexe. Dans [88], MATHERON et SERRA ont résolu l'équation $A \oplus \check{A} = B \oplus B$ sachant que $B = \check{B}$. Ces études s'appuient sur la symétrie intrinsèque de l'élément structurant. Dans le contexte de minimisation de la surface de A , la théorie semble fort démunie lorsqu'aucune symétrie n'est décelée dans B .

8. L'équivalence plus générale est: $A \oplus B \supseteq B \oplus B$ et $A \subseteq B \bullet B \Leftrightarrow A \oplus B = B \oplus B$.

La première ligne de la figure 5.16 illustre la condition de réduction pour un élément structurant de $\mathbb{R}^2$, symétrique à une translation près. Si A se compose des quatre extrémités des diagonales –ces points sont dessinés en noir– alors $A \oplus B = B \oplus B$ et ce, quelle que soit la taille du losange.

La seconde partie de la figure 5.16 montre des éléments structurants A' et B' qui respectent la condition de réduction dans $\mathbb{Z}^2$, bien que B' ne soit pas symétrique.

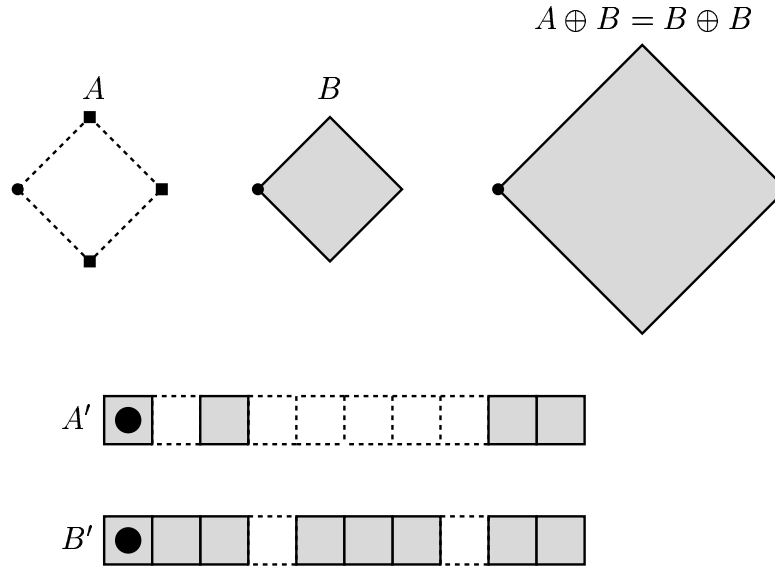


FIG. 5.16 - Deux paires d'ensembles $(A, B \subseteq \mathbb{R}^2$ et $A', B' \subseteq \mathbb{Z}^2$) qui vérifient la condition de réduction. Les disques noirs représentent les origines des différents élément structurants.

Ces deux exemples mettent en jeu des ensembles A contenus dans la frontière de B , notée ∂B . En toute rigueur, il n'est pas nécessaire de prendre tous les points de la frontière de B pour parvenir à la condition de réduction, parce que certains points de A sont redondants. En revanche, il est probable qu'un ensemble minimal A contienne uniquement des points de la frontière. En fait, choisir A égal à la frontière de B permet toujours de satisfaire la condition de réduction. Pour le démontrer, nous établissons tout d'abord deux lemmes.

Lemme 32 Soit $S = [a, b]$ un segment de $\mathbb{R}^2$ et $\partial S = \{a, b\}$ ses extrémités. On a

$$\partial S \oplus S = S \oplus S \quad (5.17)$$

Démonstration

$$\partial S \oplus S = [2a, a+b] \cup [a+b, 2b] = [2a, 2b]. \text{ D'autre part } S \oplus S = [2a, 2a+2(b-a)] = [2a, 2b].$$

■

Il existe un résultat similaire pour un ensemble constitué d'une union de segments non connectés, appartenant à une même droite.

Lemme 33 Si $S = \bigcup_{i \in I} [a_i, b_i]$ est une collection de segments de $\mathbb{R}^2$ orientés dans une même direction, avec $\forall i, j : [a_i, b_i] \cap [a_j, b_j] = \emptyset$ et $\partial S = \{a_i, b_i | i \in I\}$ alors

$$\partial S \oplus S = S \oplus S \quad (5.18)$$

Démonstration

Par croissance de la dilatation: $\partial S \subseteq S \Rightarrow \partial S \oplus S \subseteq S \oplus S$.

L'inclusion réciproque est plus délicate. Prenons $p \in [a_k, b_k]$. En vertu du lemme 32, $p \oplus [a_k, b_k] \subseteq \{a_k, b_k\} \oplus [a_k, b_k] \subseteq \partial S \oplus S$. Que se passe-t-il pour $p \oplus [a_l, b_l]$ lorsque $p \notin [a_l, b_l]$? Comme $p \in [a_k, b_k]$, $p \oplus [a_l, b_l] \subseteq [a_k, b_k] \oplus [a_l, b_l] = [a_k + a_l, b_k + b_l]$. De plus, la dilatation distribuant l'union,

$$\begin{aligned} \partial S \oplus S &= (\{a_k, b_k, a_l, b_l, \dots\}) \oplus ([a_k, b_k] \cup [a_l, b_l] \cup \dots) \\ &= (\{a_k, b_k\} \oplus [a_l, b_l]) \cup (\{a_l, b_l\} \oplus [a_k, b_k]) \cup \dots \\ &= ([a_k + a_l, a_k + b_l] \cup [b_k + a_l, b_k + b_l]) \cup ([a_l + a_k, a_l + b_k] \cup [b_l + a_k, b_l + b_k]) \cup \dots \\ &= ([a_k + a_l, b_k + b_l]) \cup \dots \end{aligned}$$

Dès lors, $p \oplus [a_l, b_l] \subseteq [a_k, b_k] \oplus [a_l, b_l] \subseteq \partial S \oplus S$.

En conclusion, $(p \oplus [a_k, b_k]) \cup (p \oplus [a_l, b_l]) \subseteq \partial S \oplus S$. Ceci signifie que $p \oplus S \subseteq \partial S \oplus S$, résultat valable pour tout $p \in S$. ■

Des formes B quelconques causent plus de soucis quand il s'agit de trouver un ensemble A minimal. Dans tous les cas pourtant, la frontière de B suffit comme le certifie le théorème suivant.

Théorème 34 *Soit ∂B la frontière du fermé B de $\mathbb{R}^2$. Alors*

$$\partial B \oplus B = B \oplus B \quad (5.19)$$

Démonstration

$\partial B \oplus B \subseteq B \oplus B$:

Ceci est évident parce que, comme B est fermé, on a $\partial B \subseteq B \Rightarrow \partial B \oplus B \subseteq B \oplus B$.

$\partial B \oplus B \supseteq B \oplus B$:

Soit D_α la collection des droites passant par le point $p \in B$ (α est par exemple un angle, cf. figure 5.17). Clairement, $\bigcup_\alpha (D_\alpha \cap B) = B$. Reste à savoir si l'ensemble $p \oplus B$ est inclus dans $\partial B \oplus B$.

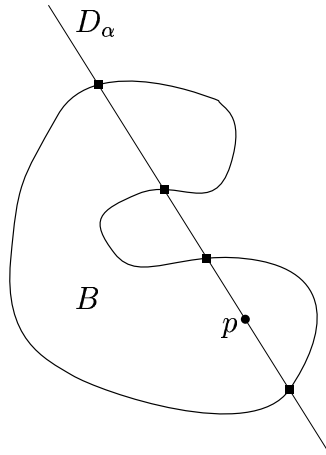


FIG. 5.17 - D_α intersecte ∂B à plusieurs endroits.

Comme $D_\alpha \cap B$ est une réunion de segments disjoints, le lemme 33 prouve que $\forall D_\alpha$, $p \oplus (D_\alpha \cap B) \subseteq \partial(D_\alpha \cap B) \oplus (D_\alpha \cap B)$, lui-même inclus dans $\partial B \oplus B$. De plus, on a les équivalences suivantes: $\bigcup_\alpha [p \oplus (D_\alpha \cap B)] \subseteq \partial B \oplus B \Leftrightarrow p \oplus \bigcup_\alpha (D_\alpha \cap B) \subseteq \partial B \oplus B \Leftrightarrow$

$p \oplus B \subseteq \partial B \oplus B$. Puisque cette inclusion est valable $\forall p$, quelle que soit sa position dans B , $B \oplus B \subseteq \partial B \oplus B$. ■

Dans [111], VINCENT fournit une version discrète (dans $\mathbb{Z}^2$), proche de ce théorème : si B est connexe et contient l'origine, $X \oplus B = X \cup (\partial X \oplus B)$ où ∂X est la frontière intérieure de X .

L'important théorème 34 donne une indication pour choisir un ensemble A à partir de B , sans minimiser pour autant sa surface. Si on souhaite réduire la surface de A , il est assez logique de sélectionner au moins les points situés aux extrémités des segments droits ; c'est à tout le moins ce que réclame le lemme 33.

Prenons le cas du polygone convexe dessiné dans la partie gauche de la figure 5.18. Dans cet exemple, il n'est pas possible de restreindre ∂B à l'ensemble des coins, comme le confirme la partie droite de la figure 5.18 : le triangle gris foncé ne sera pas recouvert complètement par $\{a, b, c\} \oplus B$. Par contre, si $A = [a, b] \cup \{c\}$ alors la condition de réduction est remplie.

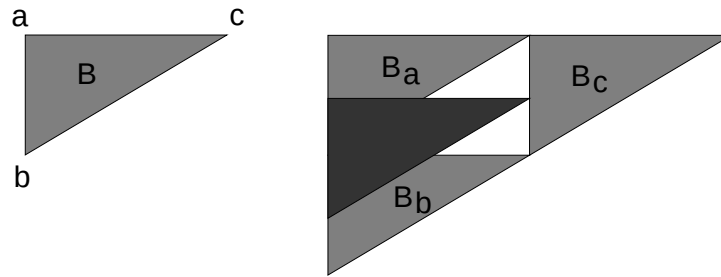


FIG. 5.18 - Ici, $\partial B \oplus B = ([a, b] \cup \{c\}) \oplus B$.

Le corollaire suivant permet parfois de simplifier cette recherche.

Corollaire 35 *Si B admet la décomposition $B = H \oplus V$, alors $B \oplus B = \partial H \oplus \partial V \oplus B$; la condition de réduction est respectée si $A = \partial H \oplus \partial V$.*

Démonstration

En vertu de la commutativité de la dilatation et du théorème 34, $\partial H \oplus \partial V \oplus H \oplus V = \partial H \oplus H \oplus \partial V \oplus V = H \oplus H \oplus V \oplus V = B \oplus B$. ■

Il suffit donc de dilater un rectangle B par ses quatre extrémités ou un hexagone par ses six sommets et son centre⁹ pour obtenir $B \oplus B$.

Situation 2 : $B \neq C$

Remarquons que les conditions $A \oplus C \supseteq B \oplus C$ et $A \subseteq B$ n'imposent nullement que l'ensemble C soit inclus dans B ou qu'il le contienne ; la figure 5.19 le prouve dans $\mathbb{Z}^2$. Le nombre de combinaisons $A - B - C$ ainsi offertes est réellement infini.

En guise d'illustration, intéressons-nous au cas des filtres alternés séquentiels linéaires.

Théorème 36 (Filtres alternés séquentiels linéaires) *Soient deux familles A_i, B_i d'éléments structurants définis dans $\mathbb{Z}$ par $A_i = \{0, i\}$ et $B_i = \{0, \dots, i\}$. Les opérateurs*

9. Dans ce cas, le centre est superflu mais le corollaire ne permet pas de le savoir.

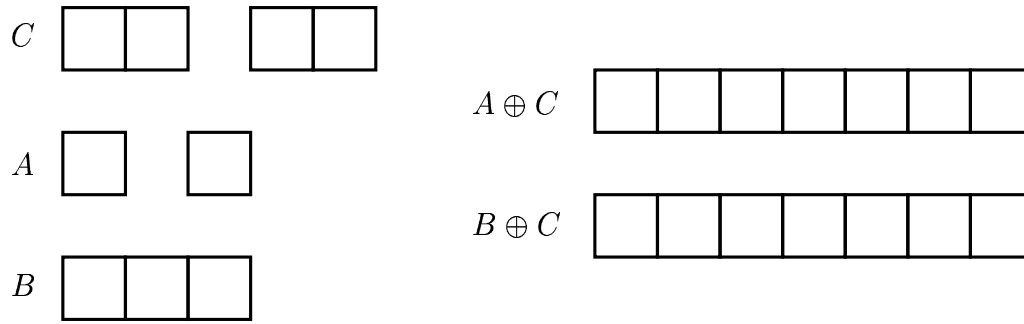


FIG. 5.19 - Pour montrer que B et C ne sont liés par aucune relation d'inclusion, même si $A \oplus C \supseteq B \oplus C$ et $A \subseteq B$.

$\eta_i \zeta_i \eta_{i-1} \zeta_{i-1} \dots \eta_1 \zeta_1$ et $\zeta_i \eta_i \zeta_{i-1} \eta_{i-1} \dots \zeta_1 \eta_1$ sont respectivement égaux aux filtres alternés séquentiels $\gamma_i \phi_i \gamma_{i-1} \phi_{i-1} \dots \gamma_1 \phi_1$ et $\phi_i \gamma_i \phi_{i-1} \gamma_{i-1} \dots \phi_1 \gamma_1$, où nous avons utilisé les notations $\gamma_i(f) = f \circ B_i$, $\phi_i(f) = f \bullet B_i$, $\eta_i(f) = (f \ominus A_i) \oplus B_i$, et $\zeta_i(f) = (f \oplus A_i) \ominus B_i$.

Démonstration

Si on prend $i = 1$, $A_1 = B_1$, d'où $(f \ominus A_1) \oplus B_1 = f \circ B_1$ et $(f \oplus A_1) \ominus B_1 = f \bullet B_1$. La définition des familles A_i et B_i entraîne que $\forall i: A_i \subseteq B_i$ et $A_i \oplus B_i \supseteq B_i \oplus B_i$. D'autre part, il est évident que $A_{i+1} \oplus B_i \supseteq B_{i+1} \oplus B_i$. Par récurrence (pour $i = 2, 3, \dots$) et en exploitant en alternance les relations 5.12 et 5.13, on démontre les deux relations. ■

Le théorème affirme qu'une opération sur deux dans la réalisation de filtres alternés séquentiels par un segment est remplacée de manière équivalente par une opération où intervient seulement le doublet contenant ses extrémités. Cette dernière consommant nettement moins de ressources, le gain avoisine un facteur 2. À cela s'ajoute le bénéfice d'une simplification du processus, susceptible d'une répercussion au niveau d'une réalisation matérielle.

5.5.2 Une illustration : implémentation des opérations morphologiques linéaires

Au cours de ce chapitre, nous avons proposé plusieurs solutions pour abaisser le nombre de calculs nécessaires à la réalisation d'opérations morphologiques. Ces améliorations ont été intégrées dans le logiciel Xlim pour des opérations avec un élément structurant linéaire et les résultats des essais sur Farm1 sont rapportés à la figure 5.20. De bas en haut, cette figure présente quatre courbes :

- *érosion linéaire* : cette courbe est une simple application de l'algorithme de calcul par histogramme local. La décroissance du temps de calcul s'explique par une gestion efficace des bords ;
- *ouverture linéaire* : cette ouverture est implémentée en une seule passe car on affecte la valeur du minimum à tous les points de la fenêtre lorsque cette valeur augmente ;
- *érosion et ouverture linéaires* : à la différence de la courbe précédente, cette simulation conserve aussi le résultat de l'érosion durant le calcul de l'ouverture. L'écart par rapport à la courbe précédente traduit donc uniquement la sauvegarde des données en

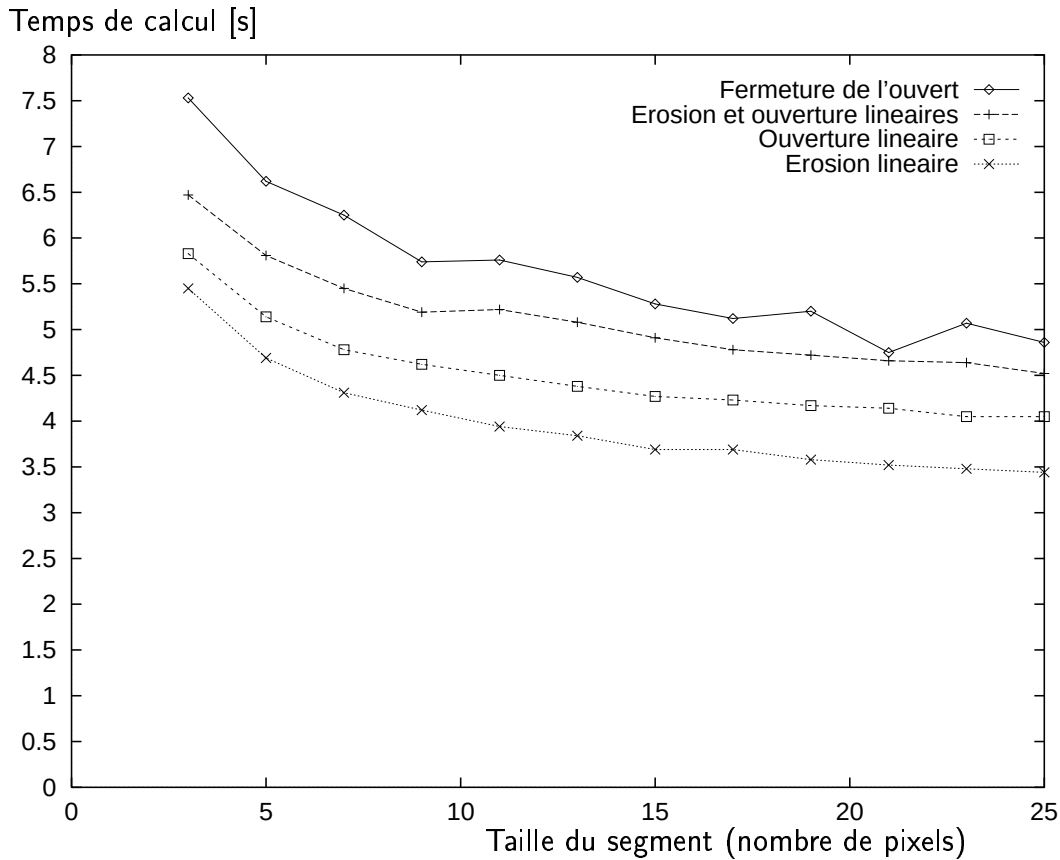


FIG. 5.20 - Temps de calcul pour différentes opérations morphologiques par un segment sur une grande image.

mémoire. Une comparaison similaire entre les deux premières courbes nous apprend alors que le calcul fictif du dilaté est réalisé pour le coût du stockage du résultat, comme si aucun calcul n'avait été nécessaire ;

- *fermeture de l'ouvert*: c'est sans conteste le résultat le plus remarquable de la série. La simulation fait usage de la propriété $[(f \circ S) \oplus \partial S] \ominus S = (f \circ S) \bullet S$, en ajoutant quelques raffinements d'ordre algorithmique car, pendant que s'opère le calcul de l'ouvert sur la fenêtre définie par l'histogramme, les valeurs du fermé de l'ouvert sont déterminées sur un même domaine voisin déjà balayé, intégrant de la sorte *toutes les opérations dans un balayage unique de l'image*. À nouveau, on peut utiliser l'information du minimum local pour juger de l'à-propos du bouchage des trous laissés entre les pics de l'ouvert. Ainsi, tant que le minimum descend, il ne sert à rien de modifier le fermé. Ce n'est que lorsque le minimum augmente, et par-là l'ouvert, qu'il convient de corriger certaines valeurs. Pour dernière amélioration, il y a la possibilité d'arrêter la modification du fermé lorsque l'algorithme détecte une longue rampe de luminance montante. La combinaison de ces éléments algorithmiques produit la courbe de la figure. Avec des algorithmes classiques, cette courbe avoisinerait le quadruple du coût de l'érosion linéaire. Alors que l'algorithme rapide produit l'ouvert à partir de l'érodé pour à peine un surcoût de 20%, le calcul de la fermeture éclair multiplie la courbe d'érosion par un facteur proche de 1,4 ! Quant aux solutions matérielles, un

essai sur une machine dédiée (Quantimet 570) a montré que le calcul du fermé de l'ouvert par logiciel était plus rapide que celui du Quantimet à partir d'une taille de 10 pixels.

Dans cet exemple, nous nous sommes limité à un élément structurant linéaire mais le principe est très général : *utiliser l'information d'un filtrage précédent pour minimiser le nombre d'opérations ultérieures.*

Conclusions du chapitre

Ce chapitre a présenté deux familles d'algorithmes efficaces pour la réalisation d'ouvertures morphologiques.

La première famille d'algorithmes utilise le concept d'histogramme local. Ce dernier est défini sur une fenêtre dont le déplacement est organisé de manière à minimiser le nombre total d'opérations nécessaires au calcul de tous les histogrammes. Si l'étape de constitution des différents histogrammes ne diffère en rien entre l'érosion, l'ouverture d'une image en niveaux de gris ou de labels, la lecture et l'interprétation de l'histogramme caractérisent les opérations. Les essais indiquent que le temps de calcul de ce type d'algorithme suit une évolution linéaire avec la taille de la frontière de l'élément structurant dans la direction du déplacement, proche de la courbe d'un algorithme qui travaille par décomposition linéaire. D'autre part, l'utilisation de l'histogramme apporte des solutions intéressantes pour l'implémentation des opérations partielles et des ouvertures d'images de labels ; il permet également de traiter un élément structurant de forme quelconque.

La seconde famille d'algorithmes, totalement nouvelle, analyse le signal préalablement à une phase de propagation. Certains points de la frontière intérieure constituent les points d'ancrage de formes homothétiques à celle de l'élément structurant. Cette forme s'étend ensuite par un processus de propagation complexe. Si les algorithmes présentent de réelles difficultés de programmation, les temps de calcul d'ouverture d'images binaires sont très faibles et, qui plus est, totalement indépendants de la taille de l'élément structurant. Ces avantages sont également perceptibles lors d'ouvertures d'images de labels.

Ce chapitre a aussi le mérite de clarifier la liaison qui existe entre les opérations morphologiques d'un point de vue algorithmique. Ainsi, il apparaît que l'érosion et l'ouverture ne représentent qu'une interprétation nuancée d'une même situation. Par exemple, l'algorithme d'ouverture par propagation peut fournir l'érodé si l'évolution du front s'accompagne d'une écriture adéquate du résultat. Nous avons également démontré d'intéressants résultats théoriques liés au nouveau principe d'une *utilisation de l'information des traitements antérieurs lors de la réalisation d'opérations complexes*. Comme illustration, prenons un élément structurant carré B et A l'ensemble des coins de B . Dans ce cas, $[(f \circ B) \oplus A] \ominus B = (f \circ B) \bullet B$. En combinaison avec les résultats théoriques, nous avons aussi montré que $\partial B \oplus B = B \oplus B$ où ∂B est la frontière de B . Quelques résultats d'implémentation avec des éléments structurants linéaires ont confirmé l'intérêt de ces développements.

Au départ, la recherche d'algorithmes rapides était une condition nécessaire pour réaliser notre technique de filtrage des contours. Mais en approfondissant la question, il est apparu que nos algorithmes pouvaient s'appliquer à plusieurs problèmes que nous rencontrons dans ce travail : ouverture d'un ensemble de régions, ouverture rapide par

un segment, érosion ou dilatation partielle, ... L'algorithme par propagation nous a aussi amené à concevoir une nouvelle représentation de la forme que l'on pourrait appeler une *description de forme par squelette propagé*, décrite plus loin. Tant la généralité des méthodes développées que leur vitesse d'exécution nous font penser que l'étude menée dans ce chapitre est importante.

Chapitre 6

Morphologie à seuillage et polymorphisme : deux approches au filtrage des contours

Introduction

La représentation de la forme d'une région que nous décrivons au chapitre 7 s'effectue en plaçant une série d'éléments structurants, choisis dans une famille $\mathcal{B}$, à l'intérieur de la région. Si par malheur, la famille d'éléments structurants est mal adaptée à la forme d'une région, il risque d'en falloir beaucoup pour la remplir. C'est par exemple le cas lorsqu'on souhaite représenter un cercle en disposant uniquement de carrés ; dans l'espace euclidien, le nombre de carrés nécessaires est même infini. Comme elle doit permettre une reconstruction parfaite de la forme, la palette $\mathcal{B}$ d'éléments structurants contient nécessairement un pixel isolé. L'importance accordée au pixel isolé est similaire à celle des autres éléments structurants, ce qui semble paradoxal au vu du rôle du pixel dans une description.

Comme nous l'avons suggéré précédemment, rien n'interdit de recourir à un *filtrage purement géométrique* pour lisser avantageusement les contours, d'autant qu'une méthode de codage orientée par la notion d'objet code à la fois les contours et les textures et que, dès lors, il est possible de corriger l'erreur introduite par un effet de lissage : il suffit de coder les textures avec une plus grande précision. Ce qui pousse à proposer ce traitement géométrique, c'est la constatation faite que le codage des contours est souvent plus coûteux que le codage des textures. À ce rythme, accroître le débit binaire propre aux textures n'est pas fondamentalement pénalisant, alors que réduire le débit binaire des contours est particulièrement intéressant. De plus, dans une optique où reproduire l'image de départ n'est pas une fin en soi mais où seule une description synthétique est souhaitée, il n'est pas indispensable de décrire tous les objets avec une précision absolue.

Plan du chapitre

Ce chapitre développe deux techniques de lissage des contours, basées sur la volonté commune de traiter un ensemble et son complémentaire d'une même façon. L'approfondissement de ce sujet indique cependant que la démarche sous-jacente dépasse le cadre

du traitement des contours, ce qui explique pourquoi, à plusieurs reprises, l'analyse est étendue aux images à teintes de gris.

Dans un premier temps, nous proposons de *lisser les contours* en comparant la forme des objets d'une image avec des éléments structurants *sans qu'il y ait l'exigence de correspondance parfaite*, c'est-à-dire qu'il subsiste une légère imprécision, mais qui est maîtrisable. De la sorte, nous espérons être à même d'atténuer l'influence du bruit et de stabiliser le résultat des opérateurs morphologiques. Tous les développements ultérieurs découlent du concept central d'*érosion partielle*¹, définie dans la section 6.1. Succinctement, cette opération effectue une corrélation suivie d'un seuillage. Outre l'atténuation du bruit, avantage-clef de l'opération, elle conduit par prolongement à définir des opérateurs qui traitent un signal et son complémentaire d'une manière identique (opérateurs auto-duaux). Par dualité avec l'érosion, nous proposons une définition de la *dilatation partielle*. Si ce concept fixe le vocabulaire et prolonge l'analogie avec la morphologie mathématique, il n'est en toute rigueur pas indispensable parce que contenu dans celui d'érosion partielle. Et enfin, toujours par similitude avec les opérations morphologiques usuelles, nous définissons la *pseudo-ouverture* et la *pseudo-fermeture*. À l'inverse de développements récents en ce domaine, fondés sur la logique floue, c'est l'*aspect géométrique* qui y tient le haut du pavé ; les opérations partielles se conçoivent comme des interactions entre objets et non en termes de lois de probabilité.

Avec tous ces outils élémentaires, il devient possible d'imaginer une importante quantité de schémas mêlant ces opérateurs. Nous en étudierons quelques-uns dans la section 6.2. Les opérations partielles conduisent aussi à proposer une série de procédés de lissage des contours d'une image de labels. Ces procédés dépendent d'un seuil unique et d'une famille d'éléments structurants adaptée à l'application. Des techniques similaires sont proposées pour des images en niveaux de gris.

La section 6.3 introduit le concept nouveau de *polymorphisme d'un point*, sorte d'extension de l'ouverture morphologique. En effet, il indique pour chaque point le nombre d'éléments structurants différents qui le comprennent et sont inclus dans X . L'étude de ses propriétés montre qu'il fournit une mesure du remplissage d'une région par un élément structurant et qu'il facilite le choix des éléments structurants pour une représentation de formes, avantage dont on peut tirer profit durant une phase d'optimisation comme indiqué dans le prochain chapitre. Enfin et surtout, le polymorphisme servira de cœur à un autre procédé de lissage des contours.

6.1 Morphologie binaire à seuillage

L'approche qui consiste à utiliser un seuil pour caractériser des opérations de morphologie n'est pas neuve ; on la retrouve dans les notions de *morphologie 'à seuillage'* –les filtres de rang appartiennent à cette famille–, ou à la base de la *morphologie 'floue'*, très en vogue aux États-Unis en ce moment [91, 92]. Dans sa thèse, VAN DEN BOOMGAARD [101] détaille un formalisme pour les opérations de la morphologie mathématique, à la fois pour des images binaires et des images à niveaux de gris, calqué sur des résultats de logique empruntés au domaine de la reconnaissance de forme. Par le biais d'analogies, il unifie plusieurs types de filtrage.

1. Ce terme est emprunté à SIMON [89] pour représenter le *filtre de rang* décrit dans la littérature scientifique.

Le point faible de la plupart de ces développements est la quasi-absence d'interprétation géométrique. Idéalement, il faut veiller à conserver la notion de forme dans l'étude d'opérations à seuil. Les recherches actuelles sur le sujet négligent un peu cet aspect parce que les opérations à seuil introduisent des imprécisions de position guère aisées à modéliser. L'étude menée dans ce chapitre s'inscrit en faux contre ce mouvement d'écartement de la morphologie mathématique ; les opérations partielles relèvent avant tout d'une *conception initialement géométrique* et non probabiliste des opérations à seuillage. Par ailleurs, le formalisme utilisé s'apparente fortement aux notations de la morphologie mathématique.

Pour comprendre l'intérêt des opérations à seuil, un retour à l'objectif que nous poursuivons s'impose. Pour filtrer des ensembles binaires, le morphologue recourt classiquement à des opérations d'ouverture ou de fermeture. Or, il se fait que le résultat du filtrage dépend parfois d'un simple pixel. À titre d'illustration, prenons le cas d'un ensemble X représentant un carré amputé d'un seul pixel. Si l'élément structurant est un carré complet ayant la taille de l'ensemble X , le résultat de l'ouverture est l'ensemble vide ! Dans la mesure où l'application autorise une certaine imprécision, ne serait-ce qu'en raison des imprécisions générées par les appareils de mesure, ou simplement parce que les résultats d'une segmentation ne sont précis qu'à quelques pixels près, le résultat est décevant. On rétorquera, à raison, qu'il est plus judicieux de filtrer l'image par une fermeture dans ce cas. D'accord, mais comment procéder alors pour déterminer a priori l'opérateur à utiliser ? Et puis de toute façon, la même difficulté survient si X est une figure trouée par un carré ayant la taille de l'élément structurant à un pixel près. L'idéal est de développer un opérateur intermédiaire entre les deux opérations élémentaires que sont le filtrage par ouverture et le filtrage par fermeture. D'autre part, nous voulons éluder les questions relatives à la position des pixels manquants ; un pixel manquant en plein centre ou sur un bord doit avoir une même conséquence puisque c'est davantage le degré de l'imprécision – en fait, le nombre de pixels ‘perturbateurs’ – qui importe et non sa position dans l'objet, sans pour autant gommer le caractère géométrique de l'opération. Enfin, comme dernière exigence, il y a le besoin de créer un outil capable de mesurer le degré de filtrage avec *un seul paramètre* comme c'est le cas pour les opérations classiques de filtrage. De la sorte, si l'élément structurant est très grand et par là-même présente une grande sensibilité au bruit, nous autorisons une plus grande imprécision, proportionnelle à sa surface.

Les chercheurs ont développé toute une série de filtres morphologiques aux propriétés fort diverses [88]. Il y a notamment le *centre morphologique* qui conduit à des filtres auto-duaux, ou alors le *filtre médian*. Des techniques algébriques aboutissent également à élargir la famille des filtres. Pour citer un exemple, le supremum d'ouvertures est une ouverture, c'est-à-dire une opération croissante, anti-extensive et idempotente. Au moment où ces travaux prenaient corps, d'autres recherches menées par MARAGOS, SCHAFER [58], RONSE [79], ... exploraient la voie des filtres de rang. Il est apparu que ces opérateurs ne jouissaient pas de toutes les propriétés algébriques habituelles ; ainsi, le filtre de rang n'est ni anti-extensif, ni extensif. Pourquoi, dès lors, avons-nous poursuivi dans cette voie ? La raison est simple. Dans une démarche de lissage des contours, les propriétés relatives à l'extensivité ne nous intéressent pas, car un objet doit pouvoir s'étendre ou rétrécir localement. D'autre part, il est difficile de lier l'usage de filtres auto-duaux existants à notre procédé de représentation de contours par éléments de surface. Au contraire, la notion de pseudo-ouverture et celle associée de polymorphisme permettent de tisser une étroite liaison.

Cette section débute par la définition de l'érosion et de la dilatation partielles. Nous établissons ensuite l'importante relation de dualité avant de terminer cette section en combinant les opérations pour construire des opérations appelées pseudo-ouverture et pseudo-fermeture. Pour une étude complète des propriétés et l'extension à la notion d'opération partielle volumique, le lecteur est invité à consulter l'annexe B. Des utilisations de ces opérations seront détaillées dans la section suivante par le biais de trois applications.

6.1.1 Érosion partielle

La morphologie mathématique définit l'érosion d'un ensemble X par un élément structurant B , notée $X \ominus B$, comme l'ensemble des points p tels que l'élément structurant translaté par p est entièrement inclus dans l'ensemble X . Par une interprétation moins formelle, l'érosion effectue en quelque sorte une mesure de corrélation entre X et B pour chacun des points de l'espace [62] où la corrélation est parfaite. L'opération d'érosion partielle définie ci-après prolonge de manière analytique cette interprétation, tout en l'assouplissant. En soi, l'opération n'est qu'une réécriture de la notion de *filtre*² d'ordre ou de *filtre de rang* (cf. [58]). L'opération partielle est définie dans $\mathbb{Z}^2$.

Définition 37 Soit $\mathcal{A}(\cdot)$ l'opérateur qui calcule l'aire³ d'un ensemble. L'érosion partielle de seuil m entier ($0 \leq m \leq \mathcal{A}(B)$) d'un ensemble X par un élément structurant plan B est l'ensemble

$$X \overset{m}{\ominus} B = \{p \in \mathbb{Z}^2 \mid \mathcal{A}(X \cap B_p) \geq \mathcal{A}(B) - m\} = \{p \in \mathbb{Z}^2 \mid \mathcal{A}(B_p \setminus X) \leq m\} \quad (6.1)$$

Que signifie cette définition? Simplement que le translaté de B par n'importe quel point de l'ensemble érodé intersecte l'ensemble de départ avec une précision fixée par le seuil. Il apparaît en particulier que l'érodé partiel de seuil nul est l'érodé habituel défini par la morphologie mathématique. Autrement dit, l'érodé partiel est une extension de la notion d'érosion morphologique, ce qui justifie l'appellation de la nouvelle opération.

Propriété 38

$$X \overset{0}{\ominus} B = X \ominus B \quad (6.2)$$

Démonstration

$p \in X \overset{0}{\ominus} B \Leftrightarrow \mathcal{A}(B_p \setminus X) = 0$, c'est-à-dire que $B_p \subseteq X$. ■

Remarquons que l'érosion partielle ne fait qu'ajouter des points à l'érodé morphologique. C'est aussi un signe que l'érodé perd son caractère de réduction de la taille en fonction de la forme de l'élément structurant. Le cas de $m = \mathcal{A}(B)$ est critique. En effet, la définition en elle-même n'a plus guère de sens puisque l'aire est toujours positive. Autrement dit, l'érodé partiel remplit l'espace $\mathbb{Z}^2$ tout entier; l'opérateur a agrandi la forme au lieu de la rétrécir. Comme nous le verrons plus loin, dans son comportement, l'érosion

2. En toute rigueur et suivant la terminologie morphologique, il ne s'agit pas véritablement d'un filtre car il ne jouit pas de la propriété d'idempotence.

3. Dans $\mathbb{Z}^2$, l'aire est égale au *cardinal* de l'ensemble. Définir l'érosion partielle dans $\mathbb{R}^2$ engendre de réelles difficultés dues aux ensembles de mesure nulle ou aux ensembles non mesurables. On peut éviter ces difficultés en travaillant dans $\mathbb{Z}^n$, où tout ensemble est mesurable.

adopte le comportement d'une dilatation à partir d'une valeur de seuil proche de la moitié de l'aire de l'élément structurant B .

La notion d'érosion partielle offre l'avantage d'être paramétrique. Par un réglage du seuil, paramètre d'interprétation aisée, l'opérateur ajuste la précision. La description d'algorithmes de lissage des contours illustrera pleinement le concept. En pratique, la détermination du seuil en valeur absolue pose parfois problème. Qu'à cela ne tienne, il suffira de choisir alors un seuil directement proportionnel à l'aire de l'élément structurant. Voici la définition qui en résulte.

Définition 39 *L'érosion partielle de seuil relatif α , notée $X \overset{\alpha}{\ominus} B$, se définit par*

$$X \overset{\alpha}{\ominus} B = \{p \in \mathbb{Z}^2 \mid \mathcal{A}(B_p \setminus X) \leq \alpha \mathcal{A}(B)\} \quad (6.3)$$

C'est ce type de démarche qu'il conviendra d'adopter lorsque l'image est traitée avec une palette d'éléments structurants ayant des tailles fort distinctes.

Extension de la définition d'érosion partielle. Le cadre restrictif des ensembles de $\mathbb{Z}^2$ ne doit pas masquer la généralité des propos tenus dans cette section. Ainsi, l'érosion partielle se définit aisément pour les images en niveaux de gris de $\mathbb{Z}^n$ dans $\mathbb{Z}$ avec des éléments structurants non plans, comme nous l'a mentionné RONSE (communication privée).

Définition 40 *Soient f une fonction et g une autre fonction définie sur le support D . L'érosion partielle de seuil m de f par g est la fonction définie par*

$$(f \overset{m}{\ominus} g)(h) = \bigvee \{v \mid \mathcal{A}(\{q \in D \mid v > f(h+q) - g(q)\}) \leq m\} \quad (6.4)$$

Cette égalité signifie que pour tout point h , on regarde les valeurs v telles que le translaté de g par h , augmenté de v , se trouve en dessous de f sauf sur un ensemble de mesure inférieure à m , l'enveloppe supérieure de (h, v) donnant l'érodé partiel.

6.1.2 Dilatation partielle

Revenons à $\mathbb{Z}^2$. En-deçà d'un certain seuil, l'opération d'érosion partielle réduit l'aire de l'ensemble analysé. Pour les mêmes valeurs de seuil, il existe une opération duale, appelée *dilatation partielle*, définie également par SALEMBIER [84] et SIMON [90] sous des formes un peu différentes.

Définition 41 *Dans l'espace $\mathbb{Z}^2$, la dilatation partielle de seuil m entier ($0 \leq m \leq \mathcal{A}(B)$) d'un ensemble X par un élément structurant B est l'ensemble*

$$X \overset{m}{\oplus} B = \{p \in \mathbb{Z}^2 \mid \mathcal{A}(X \cap \check{B}_p) > m\} \quad (6.5)$$

où l'écriture $\check{B}_p$ est un raccourci pour $(\check{B})_p$.

Avoir introduit le symétrique de B dans la définition n'a pour seule raison que de préparer le terrain pour la pseudo-ouverture et de conserver l'analogie avec le formalisme existant, développé au chapitre 2.

Interprétation. La définition du dilaté partiel impose que le nombre de points communs entre X et le translaté du symétrique de B soit strictement supérieur au seuil m .

Conséquemment, le dilaté de seuil $\mathcal{A}(B)$ est toujours vide : $X \overset{\mathcal{A}(B)}{\oplus} B = \emptyset$.

Le dilaté morphologique est moins contraignant que le dilaté partiel puisqu'il exige uniquement que $\tilde{B}_p$ ait une intersection non nulle avec X . La dilatation partielle exige quant à elle non seulement que l'intersection ne soit pas vide mais également que son aire soit supérieure à un certain seuil. Tout comme pour l'érosion, il apparaît que le dilaté de seuil nul n'est rien d'autre que l'érodé morphologique :

$$X \overset{0}{\oplus} B = X \oplus B \quad (6.6)$$

Nous avons développé des *algorithmes rapides, basés sur l'histogramme local*, pour le calcul des opérations d'érosion et de dilatation partielles ; ces algorithmes sont par ailleurs très semblables à ceux utilisés pour les opérations morphologiques usuelles. Nous renvoyons à la page 117 pour une discussion plus détaillée de la mise en œuvre par logiciel.

La dilatation partielle étend moins la figure que la dilatation morphologique. D'autre part, l'érosion partielle réduit moins X que l'érosion morphologique. Nous allons voir que l'érosion et la dilatation partielles se rejoignent à mesure que le seuil se rapproche de la moitié de l'aire de l'élément structurant.

6.1.3 Correspondance entre les opérations d'érosion partielle et de dilatation partielle

Les opérations d'érosion et de dilatation partielles sont duales.

Propriété 42 (Dualité des opérations partielles)

$$(X \overset{m}{\ominus} B)^c = X^c \overset{m}{\oplus} \check{B} \quad (6.7)$$

Démonstration

De la définition de dilatation partielle, il résulte que $p \in X^c \overset{m}{\oplus} \check{B} \Leftrightarrow \mathcal{A}(X^c \cap B_p) > m$. Or $\mathcal{A}(X^c \cap B_p) = \mathcal{A}(B_p \setminus X)$, et donc d'après la définition de l'érosion partielle, $p \in X^c \overset{m}{\oplus} \check{B} \Leftrightarrow p \in (X \overset{m}{\ominus} B)^c$. ■

Plus le seuil croît, plus l'érosion partielle perd son analogie intuitive avec l'érosion morphologique. Il en va de même pour la dilatation partielle. Le lien entre les deux opérations est résumé dans la proposition suivante, valable dans le cas discret.

Propriété 43

$$\text{Pour } m \text{ entier, } X \overset{m}{\ominus} B = X \overset{\mathcal{A}(B)-m-1}{\oplus} \check{B} \quad (6.8)$$

Dès lors, tant que $m < \frac{\mathcal{A}(B)-1}{2}$, l'opération d'érosion partielle a plutôt un caractère d'érosion. Pour $m = \frac{\mathcal{A}(B)-1}{2}$, ce n'est ni une érosion ni une dilatation ; il s'agit alors d'un opérateur appelé *filtre médian* si $\mathcal{A}(B)$ est impair. Au-delà de cette valeur limite, il y a inversion du caractère de l'opération. Cette remarque est importante pour la suite car,

par combinaison d'érosions et de dilatations, nous construisons toute une panoplie de nouvelles opérations dont l'interprétation dépend très fortement de la valeur du seuil. Il est à noter que VAN DEN BOOMGAARD [101] ne définit pas explicitement la dilatation partielle, jugeant préférable de la traiter comme une érosion de seuil complémentaire alors que SALEMBIER [84] la dissocie complètement.

L'usage du filtre médian est assez répandu en traitement d'image suite à la propriété suivante :

Corollaire 44 *Le filtre médian (c'est-à-dire que $m = \frac{\mathcal{A}(B)-1}{2}$ et $\mathcal{A}(B)$ est impair) est auto-dual*

$$X \overset{m}{\ominus} B = (X^c \overset{m}{\ominus} B)^c \quad (6.9)$$

$$X \overset{m}{\oplus} B = (X^c \overset{m}{\oplus} B)^c \quad (6.10)$$

Démonstration

Nous n'établirons que la première relation, la seconde s'obtenant par un cheminement identique. Par définition du seuil d'un filtre médian, $m = \mathcal{A}(B) - m - 1$, ce qui signifie que, en vertu de la propriété 43, $X \overset{m}{\ominus} B = X \overset{m}{\oplus} \check{B}$. D'autre part, le principe de dualité exposé précédemment remplace le second terme par $(X^c \overset{m}{\ominus} B)^c$, ce qui démontre la relation. ■

Ainsi, un ensemble et son complémentaire subissent le même traitement, l'opérateur conservant cette propriété pour des images en niveaux de gris. L'inconvénient de cet opérateur est sa non-idempotence. L'itération de l'opérateur conduit parfois à des oscillations mais, plus la valeur s'écarte de cette médiane, plus les difficultés de convergence par itération disparaissent. Ce n'est nullement un problème pour nous, car tous les seuils utilisés dans les filtres de lissage des contours (cf. section 6.2.2) sont éloignés de la valeur de la médiane.

L'analyse du comportement des opérateurs partiels sous l'intersection et l'union fait l'objet de l'annexe B. Nous y avons mené une étude complète pour relever les dissemblances avec les opérations morphologiques correspondantes.

En morphologie mathématique, les opérateurs d'érosion et de dilatation servent de maillons dans la synthèse de filtres. Ceux que nous avons construits dérivent d'une démarche similaire.

6.1.4 Les opérateurs composés : pseudo-ouverture et pseudo-fermeture

Définitions

La pseudo-ouverture résulte de la mise en cascade d'une érosion et d'une dilatation. Il n'est dès lors plus souhaitable de les définir en toute généralité au moyen d'un seuil unique, quoique dans certaines applications un seul seuil suffise. Dans l'immense famille des opérations de pseudo-ouverture et de pseudo-fermeture, nous aurons à effectuer un choix pour des applications concrètes. Voici les définitions de ces deux opérateurs :

Définition 45 *La pseudo-ouverture et la pseudo-fermeture de seuils (m, n) d'un ensemble X par l'élément structurant B sont définies respectivement par*

$$X \overset{m, n}{\circ} B = (X \overset{m}{\ominus} B) \overset{n}{\oplus} B \quad (6.11)$$

et

$$X \overset{m,n}{\bullet} B = (X \overset{m}{\oplus} B) \overset{n}{\ominus} B \quad (6.12)$$

Si n est choisi égal à m , on notera les opérations par $X \overset{m}{\circ} B$ et $X \overset{m}{\bullet} B$.

Propriétés. Tout comme l'étaient les opérations élémentaires, les opérations de pseudo-ouverture et de pseudo-fermeture sont complémentaires. À témoin, la proposition suivante :

Propriété 46 (Dualité de la pseudo-ouverture et de la pseudo-fermeture)

$$(X \overset{m,n}{\circ} B)^c = X^c \overset{m,n}{\bullet} \check{B} \quad (6.13)$$

Démonstration

En utilisant l'équation 6.7 de dualité démontrée précédemment: $(X \overset{m,n}{\circ} B)^c = [(X \overset{m}{\ominus} B) \overset{n}{\oplus} B]^c = (X \overset{m}{\ominus} B)^c \overset{n}{\ominus} \check{B} = (X^c \overset{m}{\oplus} \check{B}) \overset{n}{\ominus} \check{B} = X^c \overset{m,n}{\bullet} \check{B}$. ■

Le comportement de la pseudo-ouverture et de la pseudo-fermeture n'est pas à proprement parler celui des opérations morphologiques d'ouverture et de fermeture. Si les opérations de pseudo-ouverture et de pseudo-fermeture sont croissantes, elles ne sont ni extensives, ni anti-extensives, et encore moins idempotentes. C'est l'analogie de construction qui a guidé le choix du vocabulaire et non la similitude de propriétés.

On peut distinguer plusieurs types de filtres, au sens élargi du terme 'filtre'. Parmi ceux-ci, signalons $X \overset{m,n}{\circ} B$, $X \overset{m}{\circ} B$ et $X \overset{m,0}{\circ} B$ pour le codage, avec les équivalents anti-extensifs respectifs $(X \overset{m,n}{\circ} B) \cap X$, $(X \overset{m}{\circ} B) \cap X$ et $(X \overset{m,0}{\circ} B) \cap X$. Ce dernier, appelé *filtre de rang-maximum* [81], est une ouverture.

6.2 Illustrations des opérations partielles : détection des pointes d'un objet, lissage des contours et filtrage d'images numériques

Pour des images abîmées par un bruit, l'application des opérateurs partiels se répercute par une suppression du bruit impulsionnel au même titre que l'application des opérations morphologiques usuelles. Par contre, leur action diffère considérablement le long de la frontière : une figure égale à son ouvert est déformée d'une certaine manière après emploi d'un filtre partiel de seuil non nul, avec en contrepartie une plus faible sensibilité à une modification de l'ensemble X . C'est donc du côté des applications qui mettent l'accent sur un traitement des contours qu'apparaîtront les nouveaux outils dérivés des filtres partiels.

Les applications présentées dans la suite du texte étayent cette conviction. La première recourt à l'utilisation de filtres partiels pour élargir des opérateurs de détection de traits topographiques. La deuxième, le lissage des contours, décrit un schéma destiné à transformer un jeu de contours de régions représentées par une image de labels. Cette démarche, nouvelle à notre connaissance, s'apparente au raisonnement mené pour la construction du *centre morphologique* [88, chapitre 8]. Dans une troisième application, nous reprenons ce raisonnement pour une image à teintes de gris.

6.2.1 Détection des pointes d'un objet

En jouant sur différentes valeurs des seuils, l'expérimentateur peut obtenir des effets fort variés sur les contours. Pour construire des outils, il suffit de reprendre les outils de détection connus, comme les divers gradients morphologiques, et de remplacer certaines opérations par des opérations partielles similaires.

Gradient et chapeau haut-de-forme partiels

L'anti-extensivité de l'érosion morphologique, garantie si l'élément structurant contient l'origine, signifie que l'image résultant de la différence ensembliste entre l'image originale et l'érodée, appelée gradient morphologique interne, concentre son énergie à l'intérieur de la figure près des bords. Nous pouvons aussi bien adopter le raisonnement dual et prendre le dilaté moins l'original, c'est-à-dire le gradient morphologique externe ; le résultat s'interprète de la même manière. Par extension, le chapeau haut-de-forme partiel est obtenu par différence entre l'original et le résultat de la pseudo-ouverture limitée à X .

Ces procédés s'appliquent tout autant à des opérations partielles, par prolongement naturel, en présence d'un bruit sur l'image. Avant de définir le chapeau haut-de-forme partiel à titre d'illustration, nous désirons souligner le rôle des seuils dans les opérations partielles : les opérations morphologiques sont très sensibles à l'élément structurant utilisé, parfois au pixel près, alors que les opérations partielles atténuent légèrement cette sensibilité. Cette remarque se double d'une autre considération : la construction des opérations partielles met en jeu des seuils qui conditionnent le résultat. Seule la combinaison élément structurant-seuil(s) conduit à une interprétation correcte. Plus que jamais, l'expérimentateur doit fixer précautionneusement son objectif pour s'en tirer à son avantage dans le dédale des opérations qui sont à sa disposition.

Définition 47 *Le chapeau haut-de-forme partiel de seuils (m,n) de X par B est défini par*

$$HF^{m,n}(X) = X \setminus [(X \overset{m,n}{\circ} B) \cap X] \quad (6.14)$$

Pour détecter les bords intérieurs d'un objet, il conviendra de choisir un seuil de dilata-tion égal à la longueur d'un côté de l'élément structurant. La figure 6.1 compare le gradient morphologique interne en 8-adjacence, $X \setminus (X \ominus B)$, au chapeau haut-de-forme par un carré 2x2 construit avec la paire de seuils $(0, 2)$. Hormis quelques pixels, les résultats concordent alors même que le raisonnement sous-jacent diffère considérablement. À ce stade d'avancement théorique, seule l'expérience peut guider l'utilisation de tel élément structurant ou de telle paire de seuils.

Recherche des coins convexes d'une figure

Un emploi dicté par les propriétés des opérateurs partiels est la détection des parties anguleuses, puisqu'il s'agit des parties d'un objet présentant une grande sensibilité au bruit. La figure 6.2 illustre la détection d'angles droits convexes d'un objet. Les traits noirs de l'image de droite sont dérivés d'un chapeau haut-de-forme partiel par un carré 4x4 avec la paire de seuils $(0, 3)$. Le chiffre 3 est idéal car, étant inférieur à la longueur d'un côté, il empêche la détection des bords droits de la figure alors que, dans les coins,

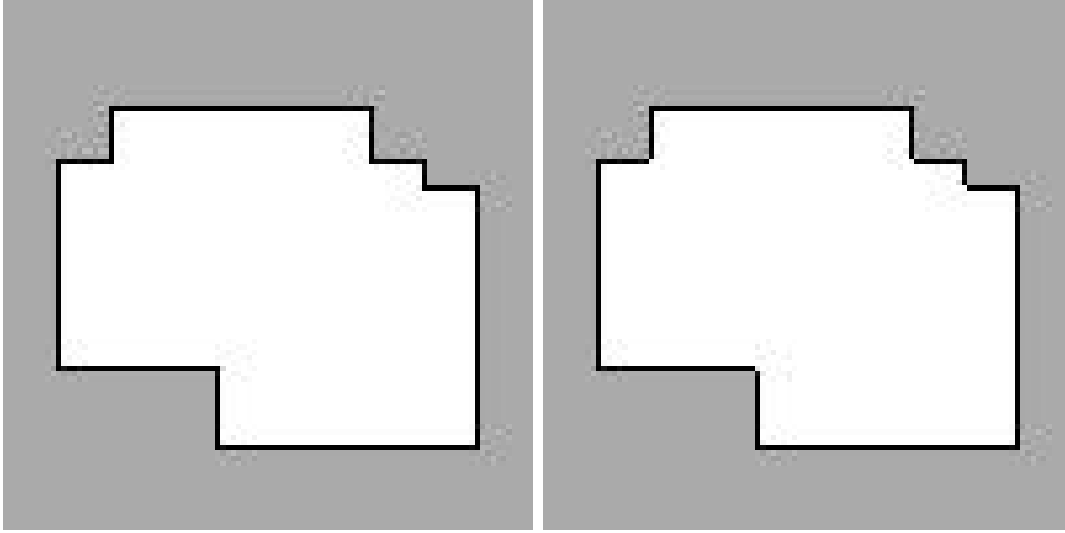


FIG. 6.1 - *Comparaison de deux différences ensemblistes, représentées en noir. L'image de gauche résulte du gradient morphologique interne (l'ensemble moins l'érodé), tandis que l'image de droite a été obtenue en soustrayant le résultat de la pseudo-ouverture de seuils $(0, 2)$ de l'image originale. On peut aussi obtenir l'image de droite par gradient morphologique interne en 4-adjacence.*

la dilatation supprime des points formant une équerre. En d'autres termes, l'ouverture partielle de seuils $(0, 3)$ élimine les points qui n'appartiennent pas à 4 copies différentes de l'élément structurant. Dans cet exemple, le chapeau haut-de-forme usuel par un losange fournit le même résultat. Cela n'est plus vrai si le contour est bruité.

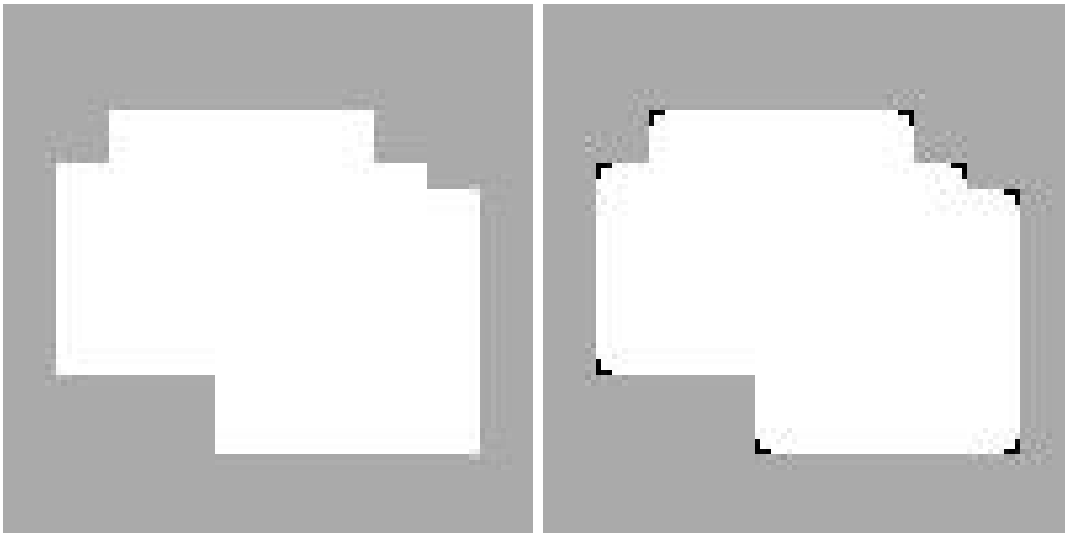


FIG. 6.2 - *Détection des parties anguleuses convexes d'un objet par application du chapeau haut-de-forme partiel.*

Ce dernier exemple illustre simultanément l'énorme richesse que contient le concept d'érosion partielle et la précision demandée dans son utilisation. Raisonner en termes d'opérations partielles et non purement morphologiques n'est pas aussi facile parce que

l'imprécision introduite par les seuils ne souffre aucune interprétation générale simple.

6.2.2 Algorithmes de lissage des contours

Par l'entremise des opérations partielles, il est possible de mettre au point des algorithmes de filtrage des contours ayant des fonctions très spécifiques. À la différence des opérateurs comme les filtres alternés séquentiels, où l'on utilise un élément structurant unique dont la taille change au fur et à mesure, les opérations que nous proposons dépendent d'un seuil et d'une famille d'éléments structurants.

Le filtrage d'une seule région ou d'une collection de régions ne se conçoit pas d'une seule manière; la différence fondamentale tient au rôle que joue l'arrière-fond. Ou bien l'arrière-fond n'apporte aucune information pertinente, ou alors, il est au contraire un objet au même titre que l'objet principal. Concrètement, ce dernier cas signifiera que les deux objets en présence sont traités sur un pied d'égalité, approche à tenir pour le filtrage de régions. On s'oriente alors vers un *filtre auto-dual*.

Description d'un procédé de filtrage pour une image de labels

La description des techniques de filtrage suit le cheminement que voici: (1) décrire les nuances algorithmiques en considérant un seul objet, (2) appliquer (1) à un ensemble d'objets en gérant les échanges des points entre régions pour que le traitement soit complètement symétrique. Conceptuellement, la méthode s'apparente à celle qui conduit à la notion de *centre morphologique* pour une image binaire. La seule différence, toutefois importante, est qu'il s'agit dans le cas présent d'une image de labels représentée par une série de niveaux de gris non liés entre eux.

(1) *Filtrage d'une seule région.*

Nous désirons lisser le contour d'un objet X en tenant compte des éléments structurants contenus dans une palette $\mathcal{B}$. Plusieurs considérations entrent dans la composition du procédé de filtrage d'une région:

- * **Filtrage simple au moyen d'un élément structurant unique.** Pour réaliser un filtrage qui agit de plein concert avec la représentation de forme, il est préférable de *choisir un élément structurant contenu dans $\mathcal{B}$* . Le filtrage le plus simple à construire se compose d'une érosion partielle suivie d'une dilatation partielle de seuils a priori quelconques. Prenons l'exemple de X traité par une pseudo-ouverture de seuils $(m, 0)$ avec un carré. Le filtre place tous les carrés dont l'intersection avec X diffère au plus de m pixels. Si le seuil de dilatation partielle est différent de 0, certains points de contours de l'objet disparaissent.
- * **Filtrage avec une famille d'éléments structurants et un seuil relatif.** Pourquoi ne pas choisir plusieurs éléments structurants pour effectuer le filtrage? Nous avons choisi cette option. Le filtre procède donc en premier lieu à la pseudo-ouverture par chacun des éléments structurants de la famille $\mathcal{B}$, puis il réunit les différents résultats. Une première expression du filtre F appliqué à la région X devient:

$$F(X) = \bigcup \{X \overset{m,n}{\circ} B \mid B \in \mathcal{B}\} \quad (6.15)$$

La famille $\mathcal{B}$ ne peut bien évidemment contenir le pixel isolé dont la pseudo-ouverture fournirait l'ensemble X lui-même. Si nous avons pris $(X \overset{m,n}{\circ} B) \cap X$ au lieu de $X \overset{m,n}{\circ} B$, $F(X)$ serait inclus dans X . L'intérêt du procédé de filtrage tient précisément au fait qu'on autorise une extension locale de X . Cet effet deviendra plus clair au moment d'analyser le filtrage d'un ensemble de régions. Il convient encore d'attribuer une paire de seuils à chaque pseudo-ouverture. L'idée centrale consiste à choisir une paire de seuils relatifs, identiques pour la plupart des éléments B . La conséquence de ce choix est double : primo, dans un schéma global de codage des contours, tel celui avec une boucle de rétroaction, n'interviennent plus que deux seuils, éventuellement liés, secundo, le seuil est accordé à la sensibilité au bruit de l'élément. En choisissant un seuil proportionnel à la surface, le filtrage autorise une erreur plus grande pour des éléments de grande taille, contrecarrant la sensibilité au bruit. En conclusion, les filtres s'expriment par :

$$F^{\alpha,\beta}(X) = \bigcup \{X \overset{\alpha,\beta}{\circ} B \mid B \in \mathcal{B}\} \quad (6.16)$$

où α et β sont des seuils relatifs identiques pour chaque élément structurant B . Notons que les seuils absolus sont calculés en arrondissant au plus petit entier inférieur. Par exemple, si α vaut 0,1 et $\mathcal{A}(B) = 4$, la valeur considérée pour m sera 0.

- * **Processus itératif.** Comme le filtre n'est pas idempotent, son application répétée produit des résultats intermédiaires différents. L'itération du processus précédent a pour effet de grignoter progressivement les petites régions, en rendant l'objet pour ainsi dire plus convexe. Loin de la valeur de la médiane, valeur susceptible d'engendrer des oscillations, la convergence est suffisamment rapide pour permettre d'interrompre le processus après quelques itérations. La pratique montre que le filtre itéré, noté $F_{\infty}^{\alpha,\beta}$, lisse les contours et supprime les petites régions contenues dans les grandes régions. Par contre, les parties de l'image à grande densité de contours ne subiront aucun lissage parce que la dilatation partielle l'interdit, à condition de choisir un seuil de dilatation non nul. Ainsi, les contours de l'image sont globalement conservés sans effet de lissage exagéré, mais avec la particularité voulue de s'intégrer au procédé de représentation de la forme.

(2) Filtrage d'un ensemble de régions.

Le problème qui se pose lors du filtrage d'une image de labels est le partage des points qui appartiennent au résultat filtré de plusieurs régions ou celui des points délaissés par toutes les régions.

- * **Gestion des conflits d'attribution de points.** La représentation des contours concerne non pas une seule région mais une collection de régions a priori de toutes formes et de toutes tailles. La construction des filtres pour une famille de régions se résume à *filtrer chaque région par pseudo-ouverture et à partager par une règle préfixée les points à indétermination*. En d'autres termes, nous commençons par appliquer le filtre $F_{\infty}^{\alpha,\beta}$ à chacune des régions X_k , puis les points appartenant à deux ou à aucune des régions reviennent à leur ensemble

d'origine. En conclusion, la définition formelle du filtre de contours est :

$$p \in G_{\infty}^{\alpha,\beta}(X_k) \Leftrightarrow \begin{cases} p \in F_{\infty}^{\alpha,\beta}(X_k) \text{ et } p \notin F_{\infty}^{\alpha,\beta}(X_i), \forall i \neq k, \text{ ou} \\ p \in X_k \text{ et } \exists i, j \mid i \neq j, p \in F_{\infty}^{\alpha,\beta}(X_i) \text{ et } p \in F_{\infty}^{\alpha,\beta}(X_j), \text{ ou} \\ p \in X_k \text{ et } \forall i, p \notin F_{\infty}^{\alpha,\beta}(X_i) \end{cases} \quad (6.17)$$

En plus de la complexification du raisonnement qu'entraîne l'usage de filtres partiels, la question algorithmique menant à une réalisation de ces filtres a dû être examinée avec attention. La tâche paraissait délicate en raison de la présence des seuils introduits dans les définitions des opérateurs partiels. Une étude menée en profondeur a abouti à la mise au point d'algorithmes de calcul spécifiques dont le chapitre 5 fait un large écho. Il en résulte que le surcoût du calcul d'une pseudo-ouverture est seulement de quelques pour cent par rapport à celui d'une ouverture morphologique usuelle.

Des exemples d'images filtrées

Pour décrire aisément les éléments structurants intervenant dans le filtre et les seuils associés à chacun d'eux, nous avons adopté une notation 'lispienne'⁴ qui précise élément après élément et leur adjoint les seuils. Par exemple, (**c16** 0,1 0,05 **l5** 0,05 0,0) signifie qu'il s'agit d'un filtre construit avec un carré de taille 16, de seuils 10% à l'érosion et 5% à la dilatation, et un losange de taille 5 avec $\alpha=0,05$ et $\beta=0,0$.

Les résultats peuvent s'interpréter suivant quelques axes tels l'effet des éléments structurants, l'influence du seuil de dilatation pour un même seuil à l'érosion, la construction d'un filtre en correspondance avec la forme des éléments structurants utilisés dans une palette de codage, ou la progressivité du filtrage par ajout d'éléments structurants de plus en plus petits. Ci-après, nous discutons deux de ces aspects.

Effet des éléments structurants. La forme des éléments structurants conditionne le résultat comme en témoigne la figure 6.3, où l'on voit deux images, de taille 128x128, filtrées⁵ par les séries (**c8** 0,1 0,0 **c4** 0,1 0,0) et (**c8** 0,1 0,0 **l5** 0,1 0,0). La différence entre les deux filtres, remplacer un carré de taille 4 par un losange de taille similaire, déteint immédiatement sur le résultat.

Action du filtrage sur la taille de la représentation. Diminuer le débit des contours est l'objectif majeur de l'introduction d'un filtrage dans la boucle de codage des contours. En termes méthodologiques, cela équivaut à allier le codage et le filtrage. La représentation des formes proposées au chapitre 7 transforme chaque région X_k en une série de triplets (B, x, y) , où $B \in \mathcal{B}'$ indique un élément structurant contenu dans X_k et $(x, y) \in \mathbb{Z}^2$, sa position dans le plan. Pour une telle description par éléments de surface, il va de soi que, s'il y a adéquation entre les éléments structurants de filtrage et de la description, le nombre de triplets diminue et, de fil en aiguille, le coût de codage car il en dépend directement.

4. Le LISP est un langage fonctionnel interprété, utilisé dans le domaine de l'intelligence artificielle. Il a servi de base au logiciel **Xlim**, dans lequel nous avons intégré tous nos travaux.

5. Le signal original est la première image de la figure 6.4.

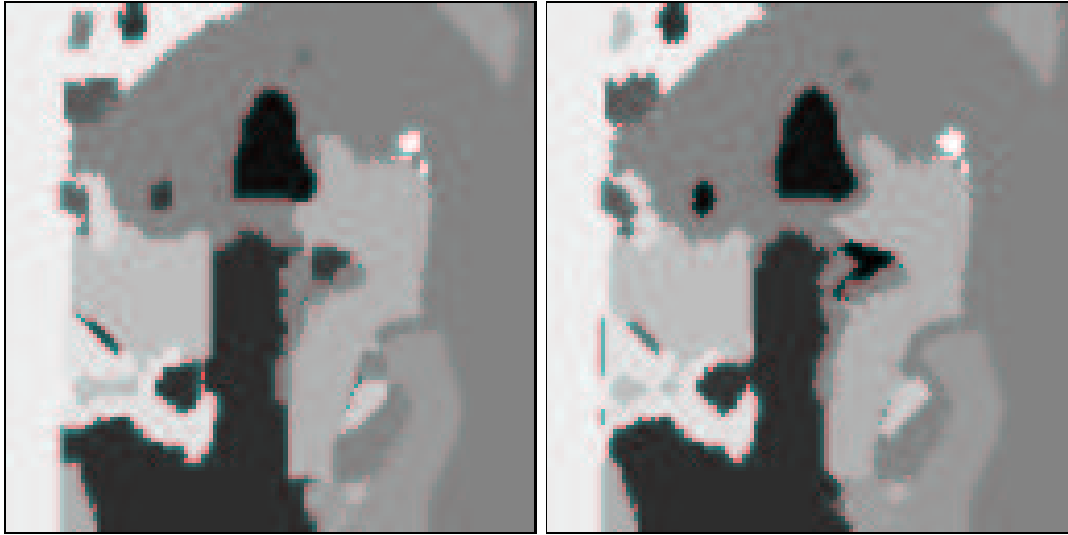


FIG. 6.3 - Influence de la forme d'un élément structurant dans la construction d'un filtre. L'image de gauche a été filtrée par des carrés de taille 8 et 4, toutes les pseudo-ouvertures ayant été menées avec la paire de seuils relatifs (10%, 0%). L'image de droite est obtenue en remplaçant le carré de taille 4 par un losange de diagonale 5, toutes choses égales d'ailleurs.

La figure 6.4 et le tableau 6.1 collectent des informations sur les représentations de quelques images filtrées. La dernière ligne du tableau montre que certains filtres apportent une économie de triplets considérable alors que le lissage des contours reste acceptable. Ce résultat provient de l'étroite liaison qui rapproche le procédé de filtrage et la représentation.

Description des filtres	Points	Triplets
Aucun filtrage	100%	100%
(c16 0,05 0,03 c8 0,03 0 c4 0,3 0 c2 0 0 l12 0,05 0,03 l5 0 0)	88%	85%
(c15 0,05 0,03 c3 0,05 0,03 l10 0,05 0,03 l3 0,05 0,03)	90%	90%
(c15 0,05 0 c10 0,05 0 c5 0,05 0 c3 0,05 0 l10 0,05 0 l5 0,05 0)	73%	60%

TAB. 6.1 - Tableau comparatif de représentations d'images filtrées. La première colonne décrit les filtres appliqués aux différentes images de la figure 6.4. Les colonnes suivantes chiffrent le nombre de points de contours et le nombre de triplets compris dans la représentation en valeurs relatives par rapport à celles de l'image originale. La palette de représentation est formée de 8 éléments : **c64**, **c32**, **c16**, **c8**, **c4**, **l5**, **c2**, **c1**.

6.2.3 Filtrage d'images en niveaux de gris

Les filtres partiels ont été créés initialement pour obtenir un lissage des contours avec des éléments de surface. Rien n'empêche de les utiliser sur des images en niveaux de gris. Le passage à des images à teintes de gris se réalise soit en généralisant l'expression de $F_{\infty}^{\alpha,\beta}$, soit en appliquant ce filtre à chaque image binaire résultant d'un seuillage de l'image numérique et en empilant les résultats, opération licite en vertu de la croissance des filtres. Nous avons testé l'efficacité des filtres partiels sur des images à teintes de gris,

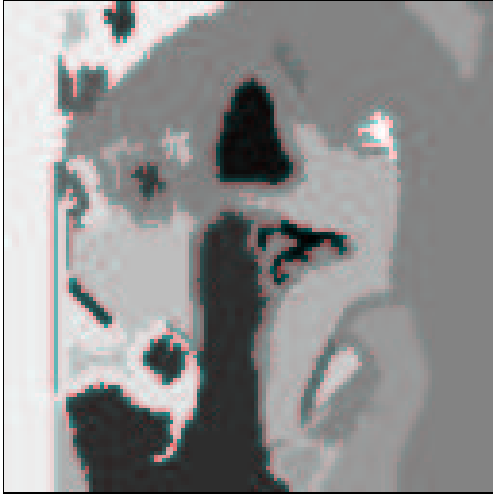
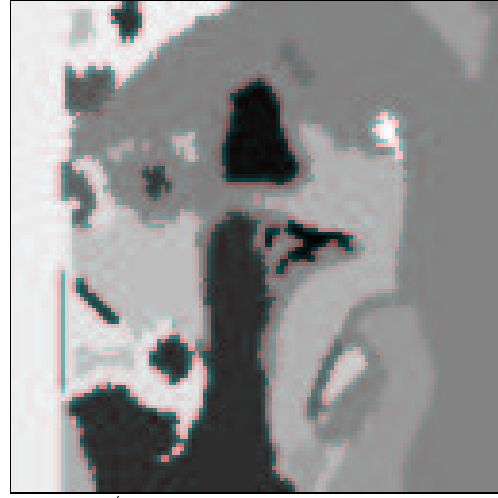
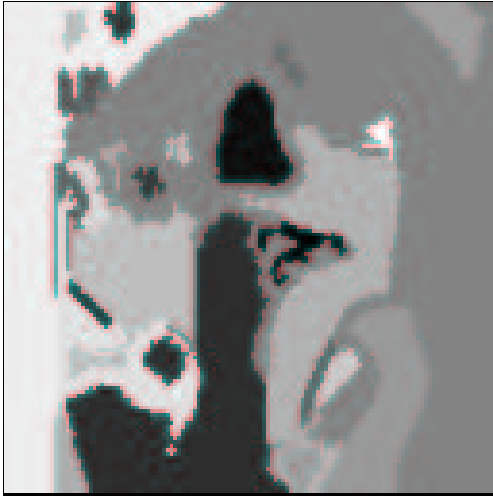


Image de labels originale.



Filtre: (c16 0,05 0,03 c8 0,03 0 c4 0,3 0 c2 0 0 112 0,05 0,03 15 0 0).



Filtre: (c15 0,05 0,03 c3 0,05 0,03 110 0,05 0,03 13 0,05 0,03).



Filtre: (c15 0,05 0 c10 0,05 0 c5 0,05 0 c3 0,05 0 110 0,05 0 15 0,05 0).

FIG. 6.4 - Images filtrées pour une comparaison de leur représentation.

bien que cela ne fasse pas partie de nos premiers objectifs. Ceci dit, la réalisation sur des images complète merveilleusement l'interprétation de l'action des filtres. La figure 6.5 montre quatre images : l'image (a) est l'originale ; nous avons représenté l'image produite par un filtre (c4 0,25 0,13) à sa droite, et enfin, les images (c) et (d) ont été traitées par des filtres alternés séquentiels $\phi\gamma$ et $\gamma\phi$ ⁶ pour étayer la comparaison.

Il apparaît très clairement sur cet exemple que le filtre partiel $F_{\infty}^{\alpha,\beta}$ généralisé supprime les petites régions attenantes ou contenues dans une grande région. Ainsi, les cheveux clairs ont disparu de même que la ligne noire contenue dans le bandeau. En outre, ce filtre efface tout aussi bien les petites régions blanches que noires car il est *auto-dual*, ce qui n'est

6. Ces combinaisons d'ouvertures γ et de fermetures ϕ se lisent de la droite vers la gauche. De plus, les indices repris dans la légende de la figure 6.5 désignent la largeur du carré choisi comme élément structurant.



FIG. 6.5 - *Traitement d'une image en niveaux de gris par un filtre partiel et comparaison avec des filtres alternés séquentiels.*

certes pas usuel en traitement d'image. Les théoriciens de la morphologie mathématique recourent généralement à des astuces pour atténuer l'asymétrie des traitements, telle la mise en cascade d'opérations complémentaires. En comparant les images (c) et (d), on peut voir que les filtres $\gamma\phi$ et $\phi\gamma$ adoptent un comportement différent sur les parties claires ou sombres de l'image. La forme de l'élément structurant y est également plus visible.

En conclusion, les filtres partiels du type $F_{\infty}^{\alpha,\beta}$ éliminent les petites régions, ils préservent globalement les contours des autres régions et présentent une plus faible sensibilité au bruit que les opérations morphologiques correspondantes, grâce à la mise en œuvre du concept d'érosion partielle. Comparer en toute généralité les filtres partiels aux filtres morphologiques auto-duaux est assez difficile, mais il faut remarquer que les filtres partiels agissent différemment le long des frontières des objets et qu'ils permettent d'obtenir des effets de filtrage très marqués sans perdre la propriété d'auto-dualité.

6.3 Le polymorphisme, un nouvel outil de description géométrique : application au lissage des contours

Lors du filtrage d'une image de labels, un point est obligatoirement transféré d'une région à une autre. La difficulté d'interprétation se double alors d'une difficulté de décision : quel critère peut-il amener à déplacer des points ? Nous avons vu que les filtres partiels permettent de générer des procédés de lissage de contours, conçus à partir de la pseudo-ouverture, qui peuvent être très variés.

La nouvelle notion de polymorphisme, thème de cette section, débouche sur une alternative intéressante au filtrage des contours, tout en étant étroitement liée aux opérations partielles. Elle s'apparente à la notion de fonction de distance, connue en morphologie mathématique ; en chaque point, cette dernière attribue la distance au plus proche point de la frontière. À l'inverse de la fonction de distance cependant, le polymorphisme peut tenir compte d'un élément structurant particulier.

Après avoir défini le polymorphisme d'un point et d'une image à teintes de gris, nous passerons ses principales propriétés en revue. Il s'en dégage une étroite liaison avec la notion d'ouverture morphologique. De fait, le polymorphisme complète cette interprétation en différenciant les positions des points dans l'ouvert, avantage utile pour accélérer, par exemple, le choix des éléments structurants dans l'optique d'une représentation de la forme. Nous terminons cette section en développant un schéma de lissage des contours entièrement basé sur la notion de polymorphisme.

6.3.1 Définition du polymorphisme

La question primordiale de la représentation d'une région par un élément structurant se résume à savoir si un point fait partie de l'ouvert $X \circ B$. Pour extraire une information relative au recouvrement d'une région par un élément structurant, nous proposons le concept de polymorphisme. Le polymorphisme d'un point x est le nombre des différents translatés de B qui sont inclus dans X et comprennent x . Un exemple dans $\mathbb{Z}^2$ est fourni à la figure 6.6. On y voit les deux seuls carrés 3×3 , dessinés en gris, comprenant le point x et inclus dans X . La zone foncée couvre la région où les deux éléments s'intersectent ; la valeur du polymorphisme est plus élevée en ces points.

Le polymorphisme d'une région s'obtient par sommation du polymorphisme de ses points. Voici les définitions formelles.

Définition 48 (Polymorphisme d'un point et d'une région) Soient X une région de $\mathbb{Z}^2$ et x un point de $\mathbb{Z}^2$. Le polymorphisme de x est défini par

$$P_{X,B}(x) = \mathcal{A}(\{p \in \mathbb{Z}^2 | x \in B_p \subseteq X\}) = \mathcal{A}(\{p \in X \ominus B | x \in B_p\}) \quad (6.18)$$

Le polymorphisme d'une région par rapport à l'élément structurant B est la fonctionnelle, notée $P_B(X)$, qui fait correspondre à X l'entier calculé par la somme des polymorphismes de chacun de ses points $P_{X,B}(x) : P_B(X) = \sum_{x \in X} P_{X,B}(x)$.

Remarquons que le polymorphisme d'une région est égal au produit de l'aire de l'érodé $X \ominus B$ et de l'aire de B :

Comme seconde illustration, les polymorphismes des points de l'image segmentée, présentée à la page 100, sont reproduits à la figure 6.8 pour différents éléments structurants: (b) un carré 15x15, (c) un carré 32x32 et (d) un losange d'une largeur 25 de diagonale. Remarquons que toutes ces images contiennent les contours des régions en sur-impression.



(a) Contours d'une image de labels.



(b) Polymorphisme pour un carré 15x15.



(c) Polymorphisme pour un carré 32x32.



(d) Polymorphisme pour un losange.

FIG. 6.8 - *Le polymorphisme des points d'une image de labels. La convention prise pour exprimer les teintes de gris est inversée par rapport à celle suivie ailleurs. Ici, la valeur du noir est 255. D'autre part, l'apparition des courbes de niveau est due à un dépassement de la dynamique $[0, 255]$.*

Nous avons introduit la dilatation partielle avec l'objectif d'affecter le comportement de la dilatation morphologique pour les points situés sur la frontière. Or, c'est bien le long du contour que l'on trouve les points dont la valeur du polymorphisme est la plus faible. Par exemple, si X et B sont des carrés, $(X \ominus B) \overset{1}{\oplus} B$ supprime tous les coins de X voire le carré entier si $X = B$. Cette constatation suggère l'existence d'un lien entre les opérations partielles et le polymorphisme. De fait, $(X \ominus B) \overset{m}{\oplus} B$ sélectionne les points dont

le polymorphisme vaut $m + 1$ ou plus.

Propriété 50

$$x \in (X \ominus B) \overset{m}{\oplus} B \Leftrightarrow P_{X,B}(x) > m \quad (6.20)$$

Démonstration

Par définition de la dilatation partielle, $x \in (X \ominus B) \overset{m}{\oplus} B \Leftrightarrow \mathcal{A}((X \ominus B) \cap \check{B}_x) > m$. Il y a donc $m + 1$ points de $X \ominus B$ appartenant à $\check{B}_x$. Soit p un tel point. $p \in \check{B}_x \Leftrightarrow p - x \in \check{B} \Leftrightarrow x - p \in B \Leftrightarrow x \in B_p$. Mais comme p est contenu dans l'érodé, $B_p \subseteq X$. ■

La notion de polymorphisme s'étend sans peine à une famille d'éléments structurants ou à des images en niveaux de gris.

Définition 51 (Polymorphisme pour une famille d'éléments) *Le polymorphisme de $x \in \mathbb{Z}^2$ par rapport à X et par rapport à une famille $\mathcal{B}$ est la somme des polymorphismes obtenus pour chacun des éléments structurants :*

$$P_{X,\mathcal{B}}(x) = \sum_{B \in \mathcal{B}} P_{X,B}(x) \quad (6.21)$$

Définition 52 (Définition du polymorphisme d'une fonction) *Soit f une fonction en niveaux de gris de $\mathbb{Z}^2$ dans $\mathbb{R}$. Le polymorphisme de f pour l'élément structurant plan B est l'opérateur défini par :*

$$P_B(f)(h) : \mathbb{Z}^2 \rightarrow \mathbb{Z} : f(h) \rightarrow \begin{cases} 0 & \text{si } f(h) \neq f \circ B(h) \\ \mathcal{A}(\{p \in \mathbb{Z}^2 | h \in B_p \text{ et } f \circ B(p) \geq f(h)\}) & \text{sinon} \end{cases} \quad (6.22)$$

Le polymorphisme d'une fonction est insensible à l'ajout d'une constante et à un changement d'échelle : $P_B(f + v) = P_B(f)$ et $P_B(\alpha f) = P_B(f)$ pour tout $\alpha, v \in \mathbb{R}$. Suivant les termes d'une interprétation topographique, la valeur du polymorphisme est élevée au creux de larges vallées, sans que la hauteur des sommets voisins n'intervienne ; nous avons marqué de telles configurations par des croix blanches dans l'image de droite de la figure 6.9. En ce sens, le polymorphisme fait le contrepoint de la notion de dynamique introduite par GRIMAUD en morphologie mathématique [31]. La dynamique d'un minimum est définie comme la plus petite hauteur à grimper pour parvenir à un minimum inférieur, quelle que soit la distance entre ces deux minima.

Par extension du polymorphisme, on pourrait imaginer d'attribuer la surface de la tache connexe à un minimum, obtenue dans le plan de sa hauteur. De cette manière, il y aurait une hiérarchisation des minima, utilisable par exemple dans une segmentation par ligne de partage des eaux.

6.3.2 Propriétés du polymorphisme binaire

Le polymorphisme jouit de quelques propriétés partagées par la plupart des opérateurs morphologiques : croissance, invariance en translation et connaissance locale. Les propriétés non démontrées sont des applications immédiates du théorème 49.

Propriété 53 (Croissance) *Le polymorphisme est croissant. Si $X \subseteq Y$ alors $P_B(X) \leq P_B(Y)$.*

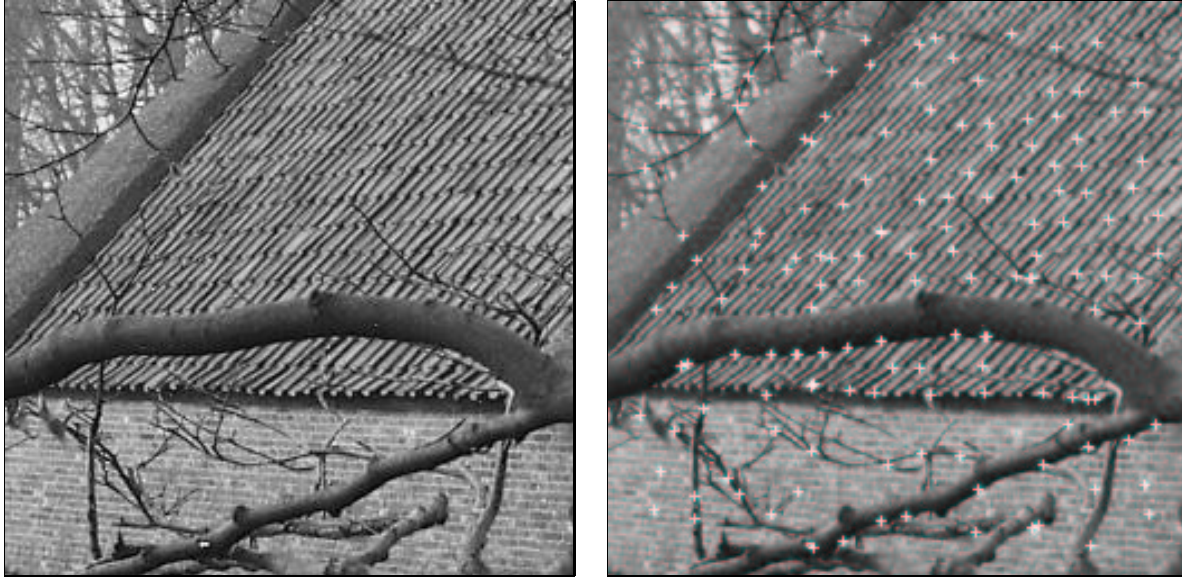


FIG. 6.9 - Marquage des points à polymorphisme élevé dans une image à teintes de gris. Les croix blanches indiquent l'emplacement des points où le polymorphisme par un carré 12×12 est maximum.

Propriété 54 (Invariance en translation)

$$\forall p \in \mathbb{Z}^2, \quad P_B(X_p) = P_B(X) \quad (6.23)$$

D'autres propriétés fixent les bornes des valeurs du polymorphisme en fonction de l'élément structurant considéré. Ainsi, le polymorphisme d'une figure qui peut se représenter par un seul élément structurant vaut $\mathcal{A}[B]$. De plus, la valeur de $P_{X,B}(x)$ ne dépasse jamais $\mathcal{A}[B]$.

Propriété 55 Pour B borné,

$$\text{si } X = B_p \text{ alors } P_B(X) = \mathcal{A}(B). \quad (6.24)$$

Propriété 56

$$\forall x \in \mathbb{Z}^2, \quad P_{X,B}(x) \leq \mathcal{A}(B) \quad (6.25)$$

Démonstration

Ce résultat est évident car, dans le meilleur des cas, le point x occupe successivement toutes les positions de B translaté. Comme le nombre de ces positions est limité à l'aire de B , $P_{X,B}(x) \leq \mathcal{A}(B)$. ■

Ces relations tissent un premier lien entre l'ouvert d'un ensemble et le polymorphisme. Il existe un autre lien qui précise que le polymorphisme des points extérieurs à l'ouvert est nul et, inversement, que tout point de polymorphisme non nul appartient à l'ouvert.

Propriété 57 Si $x \notin X \circ B$ alors $P_{X,B}(x) = 0$. Par-là même, $P_B(\emptyset) = 0$.

Démonstration

Par interprétation de l'ouverture de X , $x \in X \circ B \Leftrightarrow \exists p | x \in B_p \subseteq X$. Si un tel point p n'existe pas, x n'appartient pas à l'ouvert et suivant la définition 6.18, $P_{X,B}(x) = 0$. ■

Le polymorphisme : un outil pour exprimer le remplissage de X par un élément structurant B . Il est possible d'orienter l'optimisation d'une représentation de forme par éléments structurants en fonction des valeurs du polymorphisme, car elles expriment, par comparaison avec la surface de B , le degré de recouvrement des éléments structurants à l'intérieur de l'ouvert. Tout d'abord, remarquons que tout ouvert non vide comprend des points inclus dans une seule copie de l'élément structurant. Pour plus de généralité, nous passons momentanément à $\mathbb{R}^2$.

Lemme 58 Soient X, B des ensembles compacts de $\mathbb{R}^2$ et $X \circ B$ un ouvert morphologique⁷ d'intérieur non vide. On trace les tangentes à $X \circ B$ à partir d'un point r extérieur à l'enveloppe convexe de $X \circ B$; ces tangentes sont telles que $X \circ B$ est totalement inclus dans le secteur angulaire qu'elles délimitent. Les deux points de contacts les plus proches de r le long des tangentes sont tels qu'ils n'appartiennent chacun qu'à un seul élément B_p contenu dans l'ouvert.

Démonstration

Soit a le point le plus proche de r sur une des tangentes. Comme a appartient à l'ouvert, il existe p tel que $a \in B_p \subseteq X \circ B$. Supposons que a soit inclus dans une autre copie de B contenue dans l'ouvert : $\exists q \neq 0 \mid a \in B_{p+q} \subseteq X \circ B$. En toute généralité, on peut placer l'origine des axes $\vec{x}$ et $\vec{y}$ en a de sorte que l'axe $\vec{y}$ soit confondu avec la tangente ; ceci est illustré à la figure 6.10.

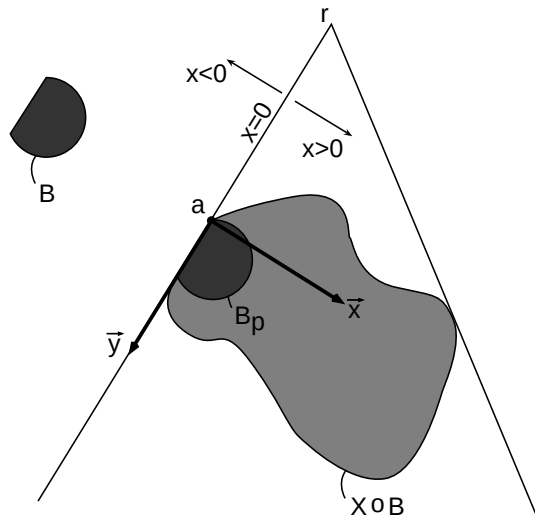


FIG. 6.10 - Placement des axes de $\mathbb{R}^2$.

Par définition de la tangente et suite au placement de l'origine, il est clair que la composante q_x de q suivant l'axe $\vec{x}$ doit être positive : $q_x \geq 0$. D'autre part, $a \in B_{p+q} \Leftrightarrow a - q \in B_p$. Or, tous les points de B_p ont une coordonnée positive suivant $\vec{x}$ et comme $a_x = 0$, cela signifie que $q_x \leq 0$. En combinant ce résultat avec le précédent, on a $q_x = 0$. Si la composante suivant $\vec{x}$ de q est nulle, il reste à examiner la valeur q_y . Il est clair que $q_y \geq 0$. En effet, si ce n'était pas le cas, le point de B_{p+q} correspondant à a dans B_p serait un point plus proche

⁷ Partout dans cette section, l'ouvert est considéré au sens morphologique ; il faut prendre garde à ne pas confondre la notion d'ouvert morphologique avec celle d'ouvert topologique.

de r le long de la tangente car $q_x = 0$. D'un autre côté, le point $a - q$ ($\in B_p$) ne peut avoir une coordonnée suivant $\vec{y}$ inférieure à celle de a , autrement dit $a_y - q_y \geq a_y \Leftrightarrow q_y \leq 0$. À nouveau, par combinaison avec $q_y \geq 0$, on a $q_y = 0$. Ainsi, $q = 0$, ce qui contredit l'hypothèse. a appartient donc à un seul translaté de l'élément structurant contenu dans l'ouvert. ■

Théorème 59 *Soient X, B des ensembles compacts de $\mathbb{R}^2$ et $X \circ B$ l'ouvert morphologique d'intérieur non vide. Il existe 3 points de l'ouvert qui n'appartiennent qu'à un seul translaté de B contenu dans l'ouvert. Dans $\mathbb{R}$, il y a 2 points de ce type.*

Démonstration

Pour établir le théorème, nous commençons par construire un triangle dont les côtés sont tangents à l'ouvert $X \circ B$ et dont les sommets sont en dehors de l'enveloppe convexe de cet ensemble. Si X et B sont des compacts, l'ouvert est aussi un compact et un tel triangle existe toujours. En vertu du lemme 58, deux points inclus dans un seul translaté sont associés à chaque sommet du triangle. Dans le pire des cas, ces points sont confondus et alors ils sont au nombre de 3. Dans $\mathbb{R}$, les deux points de $X \circ B$ situés le plus à gauche et à droite sont clairement contenus dans un seul élément structurant. ■

Conjecture 60 *Si X et B sont des ensembles compacts d'un espace $\mathbb{R}^n$ et l'ouvert morphologique $X \circ B$ est d'intérieur non vide, il existe $n + 1$ points de l'ouvert qui ne sont inclus que dans un seul translaté de B contenu dans l'ouvert. Ce résultat a été prouvé dans $\mathbb{R}$ et $\mathbb{R}^2$.*

Ces points appartiennent donc à un seul élément structurant. Dans une technique de représentation de l'ouvert par éléments de surface, il faudra recouvrir tous les points, alors autant commencer par placer les éléments structurants de manière à recouvrir en premier lieu les points n'appartenant qu'à un seul élément structurant. Ce raisonnement peut encore être répété après avoir retranché ces points, voire les éléments structurants correspondants. Cette idée est mise en œuvre dans le chapitre 7 (cf. page 177).

6.3.3 Comment calculer le polymorphisme?

De même qu'au départ les praticiens de la morphologie mathématique pensaient à la lourdeur du calcul, on pourrait être tenté de rejeter le concept parce que la réalisation semble exiger une recherche exhaustive. Pourtant, il y a moyen de construire un algorithme de calcul rapide. La propriété suivante est à la base d'une implémentation pratique du polymorphisme. Pour l'illustrer, nous proposons au lecteur de faire l'exercice sur l'exemple de la figure 6.7.

Propriété 61 *Pour tout point x de X ,*

$$P_{X,B}(x) = \mathcal{A}[(X \ominus B) \cap (\check{B})_x] \quad (6.26)$$

Démonstration

On mène un raisonnement similaire à celui qui a conduit à la propriété 50. Soit un point

p appartenant à $(\check{B})_x$. Après transformation, $p \in (\check{B})_x \Leftrightarrow x \in B_p$. Or, p est contenu dans l'érodé et dès lors $B_p \subseteq X$. Ainsi, $x \in B_p \subseteq X$. Il suffit de prendre l'aire de l'ensemble des p qui remplissent cette condition pour obtenir le polymorphisme. ■

L'égalité 6.26 fait intervenir une simple érosion. D'autre part, les algorithmes de calcul par histogramme local permettent de déterminer la valeur de l'intersection avec l'érodé en une seule étape; il suffit pour cela de relever la fréquence des points différents de 0. Dans le cas d'une image de labels, il est même possible d'obtenir le polymorphisme de tous les points par le mécanisme suivant. Chaque point de l'érodé reçoit la valeur du label uniquement si la fréquence de ce label est égale à l'aire de B à l'intérieur de la fenêtre. Un second balayage avec l'histogramme parcourt l'érodé, établit le nombre de points de $\check{B}_p$ ayant la même valeur de label qu'en p dans l'image de départ, et affecte cette valeur au point p . En conclusion, le temps de calcul pour traiter une image de labels correspond à celui de l'ouverture d'une image binaire.

6.3.4 Application au filtrage

Dans une technique de représentation de la forme par éléments structurants, la position exacte des points du bord est critique et peut entraîner une forte sensibilité au bruit. La valeur du polymorphisme donne une sorte d'image de cette sensibilité, tout en caractérisant le recouvrement entre éléments. Ainsi tout naturellement, le polymorphisme d'un point 'central' est plus élevé que celui d'un point de la frontière.

Le procédé de lissage des contours décrit ci-après reprend le principe développé lors de l'étude des filtres partiels: changer l'appartenance de certains points en fonction d'un critère évaluant l'efficacité d'une représentation. Dans le cas présent, il s'agit de maximiser le polymorphisme total en échangeant des points pour lesquels $P_{X,B}(x)$ est petit.

Le schéma de filtrage proposé, valable pour une image de labels L composée de M régions ($L = \bigcup_{1 \leq j \leq M} X_j$), comprend plusieurs étapes:

- (1) Calcul du polymorphisme des points de l'image de labels. Tous les points de polymorphisme nul reçoivent un label fictif, noté l_f .
- (2) Tous les points dont le polymorphisme dépasse un certain seuil sont attribués à leur région d'appartenance, une fois pour toutes.
- (3) Phase itérative:
 - (3.1) Pour chaque point non attribué et ayant un voisin de label différent, on calcule la variation de polymorphisme subséquente à une réallocation du label. Si la configuration l'autorise, on choisit tour à tour tous les labels voisins.
 - (3.2) Parmi tous les choix possibles, on réalloue le point qui produit le plus grand accroissement de polymorphisme total.
 - (3.3) Mise à jour des polymorphismes et des variations résultant d'échanges entre régions dans une fenêtre de taille $B \oplus \check{B}$, centrée sur le point modifié.
 - (3.4) Retour en (3.1) tant que l'accroissement de polymorphisme est strictement supérieur à 0.

Une petite remarque est propre à l'étape (3.2). Il se peut très bien que plusieurs points, produisant un même accroissement de polymorphisme total, soient compris dans

une fenêtre de type $B \oplus \check{B}$, c'est-à-dire la zone dans laquelle les points s'influencent. On peut fixer une règle arbitraire pour sélectionner un tel point, le résultat final n'étant pas très sensible à la manière de le choisir.

Nous avons rassemblé quelques images filtrées à la figure 6.11. La notation ζ_i représente le filtre 'polymorphique' avec un élément structurant carré de taille i , où seuls les points de polymorphisme nul étaient susceptibles d'une réallocation. À la manière des filtres alternés séquentiels, la mise en cascade de filtres de taille croissante, comme ceux utilisés pour produire les images (c) et (d), permet le lissage progressif des grandes régions alors que les petites disparaissent.

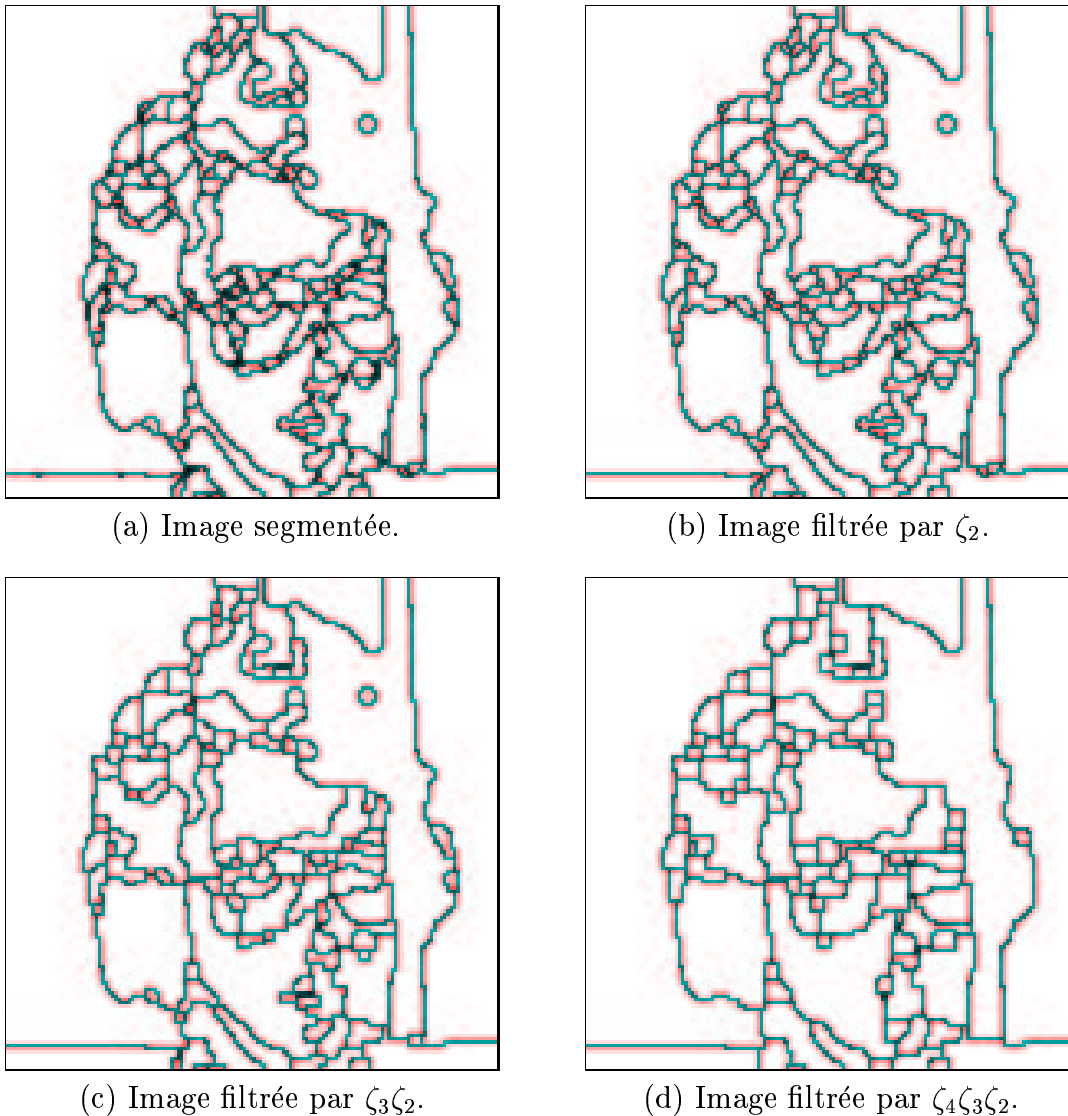


FIG. 6.11 - Comparaison des contours d'images de labels filtrées par application du polymorphisme. ζ_i représente le filtre 'polymorphique' avec un élément structurant carré de taille i .

Conclusions du chapitre

Dans une description d'image orientée par la notion d'objet, représenter les contours est une affaire délicate et d'autant plus importante que le débit associé aux contours est élevé par rapport au débit des textures. Pour simplifier la représentation de la forme et diminuer la sensibilité à un changement local, il est intéressant de lisser les contours. Encore faut-il que le filtrage soit cohérent avec la représentation !

Ce chapitre a détaillé deux techniques de lissage basées sur une approche commune qui consiste à travailler à partir d'éléments de surface et à échanger des points entre régions.

La première famille de filtres s'articule sur la notion d'*opérateur partiel* dans le cadre général de la morphologie à seuillage. Quelques illustrations ont complété les développements théoriques de manière à éclaircir les effets des opérateurs partiels. Puis, nous avons décrit un procédé de lissage de contours auto-dual et analysé son comportement.

Le concept de *polymorphisme d'un point et d'une région* est à l'origine de la seconde technique de lissage des contours. Dans un premier temps, nous l'avons défini et étudié ses principales propriétés. Il apparaît que le polymorphisme enrichit une simple description menée au moyen de l'ouverture ; le résultat nuance mieux l'appartenance d'un point à une région, tout en conservant le caractère local des opérations morphologiques. Quant à l'implémentation, elle ne pose pas la moindre difficulté parce que les algorithmes rapides utilisant l'histogramme local conviennent parfaitement. In fine, nous avons intégré l'outil que constitue le polymorphisme dans un schéma de lissage de contours.

Si nous l'avons développé dans le cadre des opérations partielles et du polymorphisme, le principe de filtrage est très général. Il revient à se doter d'un moyen de caractérisation géométrique local des objets et à distribuer les points entre régions pour augmenter la régularité géométrique. On pourrait par exemple imaginer l'utilisation de la fonction d'étanchéité ou de la bissectrice conditionnelle (cf. [87]) pour lisser les contours.

Chapitre 7

Représentation et codage des contours

Introduction

Ce chapitre développe une représentation de la forme et la transforme à des fins de codage.

Représenter la forme d'un objet unique exige la description précise de son contour. S'agissant d'une image de labels, la question est quelque peu différente car une région partage certaines frontières avec les régions voisines. Pour éviter une double représentation des contours des régions et, plus généralement, pour exprimer l'organisation spatiale des régions, nous avons choisi d'interpréter les relations de voisinage des régions par un graphe. Ce modèle, dont le rôle apparaît plus loin, est exposé à la section 7.1.

La section 7.2 détaille quelques représentations de la forme décrites dans la littérature. Nous distinguons les méthodes externes, basées sur une description des frontières par suivi de contour, des méthodes internes qui recouvrent l'intérieur d'une région par une série d'éléments structurants.

Le cadre général d'une représentation de base fait l'objet de la section 7.3. Le principe consiste à remplir une région au moyen d'éléments structurants contenus dans une palette $\mathcal{B}$. Afin de réduire le nombre total d'éléments structurants nécessaires, on autorise le recouvrement des régions déjà représentées, ce qui suppose un ordre de représentation des régions. Il n'en résulte aucune ambiguïté si l'ordre est connu du procédé de recomposition τ . Pour rappel, l'approximation est obtenue par recomposition de la représentation : $\pi(X) = \tau[\rho(X)]$.

La représentation étudiée n'est pas exempte d'un certain recouvrement géométrique entre éléments structurants d'un même type. La section 7.4 explique par quels procédés il est possible de restreindre les contributions de certaines familles d'éléments sans affecter la qualité de l'approximation.

En vue de réaliser le codage de la représentation, une phase de réduction, développée à la section 7.5, opère une réorganisation et une simplification supplémentaire de la représentation par un algorithme adaptatif. De plus, elle fixe l'ordre de codage des régions. Les quelques essais effectués montrent que le coût de codage de la représentation est supérieur au coût d'un codage par suivi de contours. Il faut néanmoins se garder de tirer

des conclusions hâtives que certaines modifications, parmi toutes celles envisageables et proposées à la fin du chapitre, pourraient contredire.

Ce chapitre est un essai qui débouche sur des nouvelles pistes de recherche ; il ne prétend pas résoudre la question du codage.

7.1 Éléments de modélisation

La difficulté majeure rencontrée dans la représentation d'une image de labels est la traduction rigoureuse de la démarche. Comment, par exemple, décrire brièvement la forme d'une région ou exprimer qu'une région est voisine d'une autre ? À la base, il y a donc un problème de modélisation.

Pour répondre à ces questions, nous avons utilisé deux théories existantes. La première est la morphologie mathématique. La seconde, la *théorie des graphes*, opère à un autre niveau ; dans notre cas, elle sert à modéliser l'organisation spatiale des régions.

Avant d'entamer la description de représentations des contours, nous décrivons les modèles relatifs aux régions, employés dans ce chapitre.

Modèle de régions. Après la segmentation, la représentation $\rho(X)$ comprend M régions X_k ($1 \leq k \leq M$ et $X_k \subseteq \mathbb{Z}^2$) connexes et disjointes deux à deux. Autrement dit, nous émettons l'hypothèse qu'aucune région n'est coupée en deux par une autre :

$$\rho(X) = \{X_k \subseteq \mathbb{Z}^2 | 1 \leq k \leq M\} \text{ avec } \begin{cases} \bigcup_{k=1}^M X_k \text{ remplit toute l'image} \\ \forall i, j \in \{1, \dots, M\} : X_i \cap X_j = \emptyset \\ \forall k \in \{1, \dots, M\} : X_k \text{ est connexe} \end{cases} \quad (7.1)$$

Nous parlerons parfois de l'ensemble X –à ne pas confondre avec $\rho(X)$ – pour décrire les traitements sur un seul objet et éviter de devoir écrire systématiquement X_k .

Modélisation de l'organisation spatiale des régions par un graphe. La caractérisation de l'organisation spatiale des objets est fondamentale dans une approche par objet. Mais comment modéliser correctement la disposition des objets dans le plan ? La réponse est loin de tomber sous le sens. Tout comme PAVLIDIS [70], nous proposons de gérer l'organisation des régions par un *graphe*.

Définition 62 *Un graphe G est un ensemble F non-vide muni d'une relation R anti-réflexive et symétrique.*

La relation R considérée est la *relation de voisinage* entre régions définie ci-après.

Définition 63 *Deux régions X_k et X_l sont voisines si et seulement s'il existe un point quelconque de X_k dont le 4-voisinage¹ comprend un point de l'autre région, c'est-à-dire que*

$$\exists (x, y) \in X_k | V_4(x, y) \cap X_l \neq \emptyset \quad (7.2)$$

1. Le 4-voisinage d'un point est défini à la page 35.

L'ensemble F lié au graphe G contient les *sommets* du graphe. Le nombre de sommets de G est appelé *ordre* de G .

Définition 64 Le graphe G associé à la représentation par objets $\rho(X)$ est l'ensemble des sommets tel qu'il y ait un sommet par région X_k ($1 \leq k \leq M$), muni de la relation de voisinage. L'ordre du graphe vaut M .

La figure 7.1 montre une image de labels et le graphe associé.

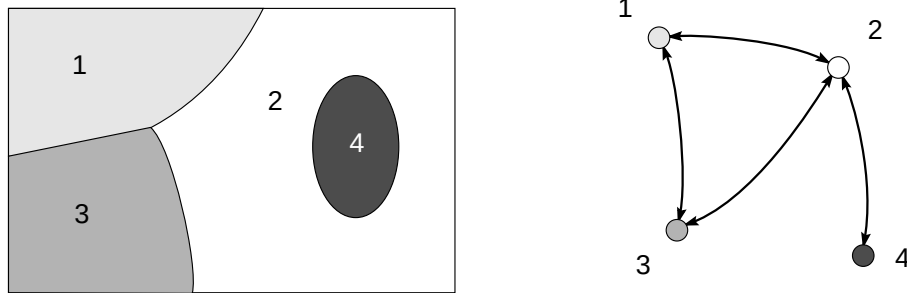


FIG. 7.1 - Une image de labels et le graphe qui modélise l'organisation spatiale des régions.

7.2 Description de représentations de formes

La description d'objets est particulièrement importante en analyse d'images et en reconnaissance de formes. Les solutions proposées pour la représentation de formes subissent toutes l'influence de la finalité de l'application. Ceci explique pourquoi il existe une multitude de procédés de description d'objets, classés en deux catégories: les descriptions *externes* et les descriptions *internes*. Les premières opèrent une description du contour; les secondes décrivent la surface de l'objet.

Nous allons développer ci-après quelques descriptions externes; l'une d'entre elles nous servira de point de comparaison pour estimer l'efficacité de codage. Après cette courte énumération de méthodes externes, nous examinerons quelques représentations internes, dont l'importante description par squelette, indirectement à la base de notre représentation.

7.2.1 Les représentations externes

La description externe des régions consiste en une représentation de contours effectuée point par point. Partant d'un point quelconque d'un contour, l'algorithme suit le contour en indiquant la direction suivante et la code. FREEMAN [25] est l'inventeur de ce procédé de codage. Initialement, l'information nécessaire pour indiquer les déplacements sur une grille rectangulaire en 4 ou 8-adjacence exige respectivement 2 ou 3 bits par point de contour, ce qui est énorme. Les solutions proposées dans la littérature scientifique visent à réduire cette quantité:

- Codage des contours par segments de droite.

DUDA et HART [19] sont parmi les premiers à avoir proposé des substituts au codage

de FREEMAN. Leur méthode revient à décomposer progressivement la courbe en segments de droite de plus en plus fins jusqu'à satisfaire à un critère d'erreur pour chaque segment. La construction de l'approximation $\pi(X)$ s'appuie donc sur une distance et un algorithme désignant quel segment est pertinent. DUNHAM [20] a imaginé un algorithme capable de choisir les segments de manière à minimiser leur nombre tout en respectant un critère d'erreur.

– *Représentation par directions préfixées.*

L'inconvénient d'un algorithme de codage par suivi de contour est parfois la variété des orientations des contours, qui entraîne une chute des performances dans le cas de formes peu régulières. MINAMI [65] approxime une courbe par des morceaux de segments de longueur fixe mais d'orientations définies sur une fenêtre discrète d'une certaine taille. Bien que la courbe ne puisse être restituée parfaitement, la méthode présente de bonnes performances pour des figures complexes.

– *Codage de FREEMAN amélioré dans une grille rectangulaire.*

La méthode nous servant de point de comparaison est celle développée par EDEN et KOCHER [22]. Elle correspond à une variante améliorée du codage de FREEMAN. Au coût de 1,2 bit par point de contour obtenu par ces auteurs précités, il convient d'ajouter le coût de représentation des points de départ. Nous prendrons donc le chiffre moyen de *1,3 bit par point de contour* en guise de comparaison.

Bon nombre de méthodes externes existent encore. Nous limitons ici leur description.

L'avantage qu'offre une représentation externe par rapport à une description interne est la possibilité de coder des contours non fermés ; une représentation interne requiert des contours tous fermés. Ce n'est nullement gênant dans notre cas puisque l'opérateur de segmentation fournit uniquement des régions fermées. Par contre, les méthodes externes conviennent peu à une description du contour à plusieurs niveaux de résolution et comment introduire la progressivité de transmission ?

7.2.2 Les représentations internes

Le point précédent a présenté des représentations externes, principalement dans l'optique d'un codage. La représentation que nous proposons appartient à l'autre famille, celle des représentations internes. Autrement dit, la description porte sur la surface de chaque objet. Pour certaines applications, une description par éléments de surface est plus efficace, d'autant qu'elle permet de travailler aisément à plusieurs niveaux de résolution.

L'important procédé de squelettisation décrit ci-après offre une première approche à plusieurs degrés de précision. Comme il est à l'origine de la plupart des techniques de représentations internes, y compris de la nôtre, nous développons son expression.

Introduction au squelette. Considérons l'espace $\mathbb{R}^2$ et une forme géométrique X . Une transformation simple consiste à réduire l'épaisseur des composantes connexes de X de sorte que l'on ne modifie pas leur nombre². Si ψ est une transformée qui réduit l'épaisseur

2. Le nombre de connexité indique le nombre de coupures possibles n'affectant pas la connexité de la figure.

de X de manière isotrope, $\psi(X)$ produit un ensemble de lignes qui dessinent l'ossature de X ; c'est la notion de *squelette* due à BLUM [3] et dénommé initialement *axe médian* ou *axe symétrique*. Plus précisément, BLUM définit le squelette comme le lieu des points d'intersection des ondes envoyées des bords vers le centre de l'objet et qui se propagent à vitesse uniforme. La figure 7.2 montre le squelette de quelques objets.

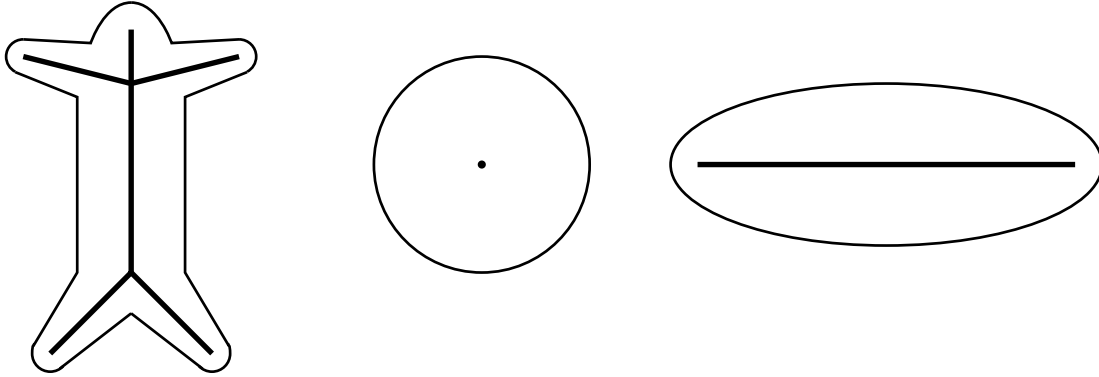


FIG. 7.2 - Le squelette de quelques figures simples.

Définition 65 (Définition du squelette) Soit un ensemble continu X et sa frontière ∂X . Un point x de X appartient au squelette de X , noté $S(X)$, s'il existe un disque centré en x inclus dans X et touchant ∂X en deux points au moins. CALABI définit le squelette comme le lieu des centres des boules maximales B contenues dans X ³, une boule étant maximale si aucune autre boule de X ne la contient.

La squelettisation, autrement dit l'opérateur $X \rightarrow S(X)$, est anti-extensive, elle respecte l'invariance en translation et la compatibilité avec les homothéties, mais la continuité n'est pas assurée. Par exemple, le squelette d'un polygone régulier est constitué de segments de droite joignant son centre aux sommets. Lorsque, par un passage à la limite, le nombre de côtés tend vers l'infini et que le polygone devient un cercle, le squelette devient pratiquement équivalent au polygone alors que le squelette d'un cercle se réduit à son centre.

Certaines techniques de codage d'objets binaires utilisent la notion de squelette. D'autres adoptent une démarche similaire sans introduire la squelettisation. À titre d'exemple, nous étudions deux représentations dont s'inspirent nos travaux :

1. *Le codage d'objets au moyen du squelette : une représentation et une approximation.*

Si en chaque point du squelette, on dessine le disque de taille maximale correspondant, l'union de ces cercles reconstitue l'objet de départ et l'enveloppe des disques formera le bord de l'objet. Soit $S_r(X)$, $r > 0$, le sous-ensemble du squelette $S(X)$ des centres des disques dont le rayon est égal à r ; nous indiquerons plus loin comment

3. Les deux définitions ne sont pas tout à fait équivalentes, mais les deux squelettes ainsi définis ont la même fermeture topologique. Des difficultés apparaissent quand on fait la distinction entre ouvert et fermé topologique. Ainsi, le lieu des boules ouvertes d'un fermé n'est pas toujours fermé. Pour éviter ce type d'inconvénient, on suppose généralement que X est un ouvert topologique. Nous renvoyons à MATHERON [60, 61] pour une étude complète des propriétés topologiques du squelette.

ces sous-ensembles $S_r(X)$ s'obtiennent par des opérations morphologiques d'érosion et d'ouverture. Le squelette s'exprime de manière équivalente par :

$$S(X) = \bigcup_{r>0} S_r(X). \quad (7.3)$$

Le squelette dans $\mathbb{Z}^2$. Bien que le concept de squelette soit difficile à étendre au cas d'images discrétisées, cela reste possible. SERRA [87] a proposé une définition de la squelettisation valable pour tout ensemble discret, borné et non vide. Cette définition est basée sur un résultat de LANTUÉJOUL [46] qui a traduit l'expression du squelette en termes morphologiques. Soit nB de $\mathbb{Z}^2$ le résultat de $n - 1$ dilatations successives par B , le squelette d'un ensemble X s'obtient par l'union de sous-ensembles S_n , chacun d'eux étant l'ensemble des points de référence de copies de nB incluses dans X , mais non comprises dans les ouverts de taille supérieure de type $X \circ (n + i)B$. Formellement, on a

$$S_n(X) = (X \ominus nB) \setminus [(X \ominus nB) \circ B] \quad (7.4)$$

$$S(X) = \bigcup_{n=0}^{\infty} S_n(X) \quad (7.5)$$

Dans cette expression, B peut par exemple être un carré de $(2n+1) \times (2n+1)$ pixels. Le sous-ensemble de taille n , $S_n(X)$, s'obtient en érodant X par nB , puis en ne conservant du résultat $X \ominus nB$ que les parties anguleuses et les lignes d'épaisseur nulle. Le squelette de taille n est la différence entre $X \ominus nB$ et son ouverture par B , $(X \ominus nB) \circ B$.

De ces propos, il découle que la famille des couples $(S_n(X), n)$ forme une description $\rho(X)$ de l'image. Pour obtenir une approximation $\pi(X)$ de l'image, il faut encore se munir d'une recomposition $\tau(X)$. La squelettisation est une des rares transformations morphologiques à posséder un inverse. L'image numérisée X est reconstruite avec exactitude par la réunion de ses sous-ensembles $S_n(X)$ dilatés par l'élément structurant de taille adéquate :

$$\pi(X) = X = \bigcup_{n=0}^{\infty} (S_n(X) \oplus nB) \quad (7.6)$$

Il apparaît clairement que plus l'indice n est faible, plus les détails de reconstruction sont fins.

Par suite de la propriété d'anti-extensivité, le squelette de X est plus mince que X . De plus, les sous-ensembles $S_n(X)$ forment une partition du squelette. Chacun d'eux est également une image binaire. Il est naturel d'envisager le codage du squelette plutôt que le codage de l'image de départ, espérant ainsi que le débit binaire nécessaire au codage des $S_n(X)$ soit moindre que le débit initial. Par le codage des $S_n(X)$ à valeurs de n décroissantes, on reconstruit progressivement l'image, à partir des traits grossiers jusqu'au détail les plus fins. BRANDT *et al.* [6] ont appliqué un codage par squelette à un jeu d'images de test défini par le CCITT pour des transmissions par

fax. Ils concluent que les performances s'apparentent à celles des codeurs classiques pour du texte et les dépassent légèrement lorsque l'image contient des graphiques.

Parmi les différentes extensions du squelette, KRESCH et MALAH [41] ont défini un squelette généralisé obtenu pour une famille d'éléments structurants.

2. *La représentation d'un objet par la décomposition morphologique de PITAS.*

L'algorithme développé par PITAS *et al.* [75] décrit l'image par une union d'objets appartenant tous à la même famille (disques, carrés, triangles, ...) de taille variable. Toutefois, aucun recouvrement n'est permis. Pour ce faire, il suffit de soustraire, de l'objet X , les parties déjà reconstruites; les objets suivants de la famille combleront les parties résiduelles. La figure 7.3 compare deux approximations d'un rectangle par des disques dont le rayon est limité inférieurement. L'image de gauche est obtenue à partir du squelette du rectangle, dessiné en gras. Dans l'approximation de l'image de droite, nous avons interdit tout recouvrement entre les disques de taille différente. Certes, cette approximation ne remplit pas autant le rectangle que la première mais, en contrepartie, le 'pseudo-squelette' associé, lieu des centres des disques, ne contient qu'une ligne horizontale et quatre points.

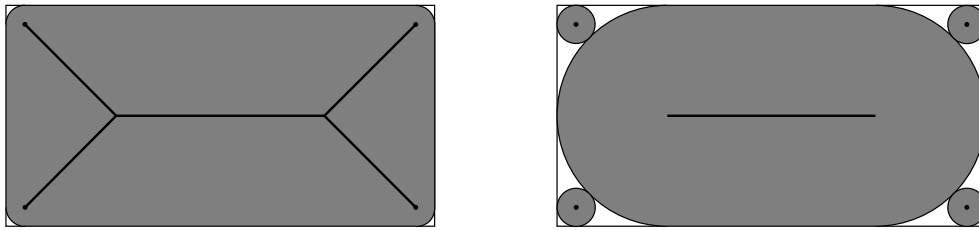


FIG. 7.3 - *Comparaison de deux approximations géométriques d'un rectangle par des disques. La figure de gauche montre la surface reconstruite à partir du squelette. L'autre image est l'approximation, obtenue à partir de la représentation de PITAS, lorsqu'on ne prend pas de disque plus petit que ceux dessinés dans les coins.*

La représentation de PITAS soulève l'intéressante *question du recouvrement géométrique*. On peut se demander dans quelle mesure le squelette convient pour une description compacte des objets parce que les recouvrements de reconstruction peuvent être importants. Un exemple de chevauchement des boules maximales du squelette est illustré à la figure 7.4. Du point de vue du codage, le recouvrement géométrique introduit une probable redondance d'information.

Un premier remède au recouvrement consiste à chercher le squelette minimum, c'est-à-dire celui comprenant uniquement les points indispensables à la reconstruction exacte de l'objet; MARAGOS *et al.* [57] ont développé de telles méthodes pour lesquelles, par ailleurs, on pourrait faire usage du polymorphisme. Même dans ce cas, il n'est pas possible de s'affranchir de tout recouvrement géométrique.

La section suivante décrit une représentation proposée par RONSE et MACQ [82], conceptuellement assez proche de celle de KRESCH et MALAH, et qui limite le recouvrement à l'intérieur d'un objet. À l'inverse de la plupart des travaux antérieurs, nous examinons la question de la représentation de plusieurs objets et non d'une seule région.

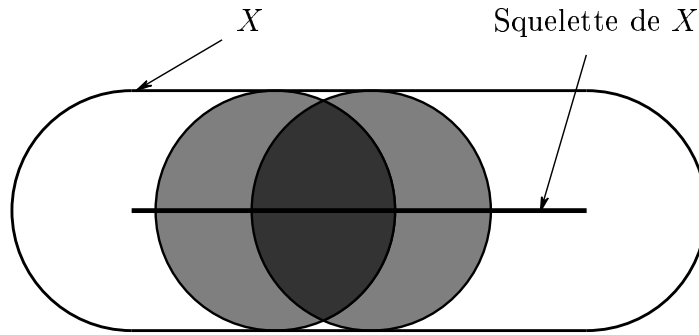


FIG. 7.4 - Le recouvrement des boules maximales du squelette.

7.3 Notre représentation des contours : description de la forme par une famille d'éléments structurants avec recouvrement

Une description en termes d'objets remplit plusieurs objectifs : interprétation automatique d'image, modélisation de matériaux (séparation de phase), reconnaissance de formes, etc. Chaque application représente à sa façon le jeu de contours. Quel dommage qu'il n'y ait pas à la base un langage commun !

Nous sommes confronté à une situation semblable. Telle représentation convient mieux pour le codage des contours, une autre pour le filtrage géométrique des régions, une dernière se prête avec aisance à une compensation de mouvement par objet. Il faut donc une *représentation fondamentale* pour éviter que les liens entre les étapes de codage ne s'estompent. En dégagant une représentation suffisamment générale dans ces principes, il suffira de préciser les modifications apportées –et de garantir qu'elles soient réversibles!– pour chaque traitement. Les phases de *simplification* et de *réduction* dont il sera question plus loin ne sont rien de plus que des désignations d'opérations qui modifient et particularisent la représentation de base. Étudions cette représentation de base.

L'étude qui suit porte d'abord sur la représentation d'un *objet unique*, puis sur la représentation d'une *collection d'objets*.

7.3.1 Représentation fondamentale d'un objet unique : description de la forme par une famille d'éléments structurants

Soit à représenter un objet unique X . L'idée maîtresse des développements à venir est simple. La morphologie mathématique compare l'objet à représenter à la forme d'un élément structurant. De plus, l'ouverture par cet élément B exprime le 'contenu' de X en B . Dans ces conditions, pourquoi ne pas étendre la famille des éléments structurants B de manière à décrire plus efficacement X ?

Nous imposons que π soit une approximation anti-extensive⁴, $\pi(X) \subseteq X$. Pour permettre la progressivité de reconstruction, on construit une séquence $\pi_1, \pi_2, \dots, \pi_n$ d'ap-

4. Il s'agit là d'une hypothèse de travail contestée par certains auteurs. Ainsi, JAGADISH *et al.* [38] préfèrent représenter une forme par une union ou une différence de primitives rectangulaires.

proximations de X qui vérifient $\pi_1(X) \subseteq \pi_2(X) \subseteq \pi_3(X) \dots \subseteq \pi_n(X)$ pour tout ensemble ou objet X . Tout l'art consiste évidemment à obtenir des approximations successives de manière cohérente. À cette fin, nous introduisons un autre concept : l'ouverture par une famille d'éléments structurants.

Définition 66 *Soit une famille $\mathcal{B}$ d'éléments structurants, l'ouverture $\gamma_{\mathcal{B}}$ de X par $\mathcal{B}$ est l'union des ouvertures de X par les éléments structurants $B \in \mathcal{B}$:*

$$\gamma_{\mathcal{B}}(X) = \bigcup_{B \in \mathcal{B}} (X \circ B) = \bigcup \{B_p | p \in \mathbb{R}^2, B \in \mathcal{B} \text{ et } B_p \subseteq X\} \quad (7.7)$$

Notons au passage que $\gamma_{\mathcal{B}}$ est croissante avec $\mathcal{B}$. En d'autres termes, si $\mathcal{B} \subseteq \mathcal{C}$ alors $\gamma_{\mathcal{B}} \subseteq \gamma_{\mathcal{C}}$ pour tout ensemble X . L'ouverture $\gamma_{\mathcal{B}}$ est invariante en translation ($\gamma_{\mathcal{B}}(X_p) = [\gamma_{\mathcal{B}}(X)]_p$), croissante ($X \subseteq Y \Rightarrow \gamma_{\mathcal{B}}(X) \subseteq \gamma_{\mathcal{B}}(Y)$), anti-extensive ($\gamma_{\mathcal{B}}(X) \subseteq X$) et idempotente ($\gamma_{\mathcal{B}} \cdot \gamma_{\mathcal{B}} = \gamma_{\mathcal{B}}$) ; $\gamma_{\mathcal{B}}$ rencontre donc les propriétés énoncées précédemment pour le choix d'un bonne approximation d'objets π . Inversement, toute transformation d'un sous-ensemble de $\mathbb{R}^2$ qui respecte ces propriétés, est une opération d'ouverture $\gamma_{\mathcal{B}}$ par une famille d'éléments structurants $\mathcal{B}$.

Théorème 67 [82] *Pour faire ressortir l'intérêt de $\gamma_{\mathcal{B}}$, montrons qu'à $\gamma_{\mathcal{B}}$ sont associés une représentation $\rho_{\mathcal{B}}$ et une recomposition $\tau_{\mathcal{B}}$ telle que*

$$\gamma_{\mathcal{B}} = \tau_{\mathcal{B}} \cdot \rho_{\mathcal{B}} \text{ et } \rho_{\mathcal{B}} \cdot \tau_{\mathcal{B}} \cdot \rho_{\mathcal{B}} = \rho_{\mathcal{B}} \quad (7.8)$$

Démonstration

Comme $\gamma_{\mathcal{B}}$ est l'union des translations des éléments structurants de $\mathcal{B}$ qui sont inclus dans X , X se représente en donnant pour chaque $B \in \mathcal{B}$ l'ensemble érodé $X \ominus B$. On définit dès lors la représentation $\rho_{\mathcal{B}}$ comme la collection des paires $(B, X \ominus B)$, pour tout $B \in \mathcal{B}$. La reconstruction $\tau_{\mathcal{B}}(\rho_{\mathcal{B}}(X))$ à partir de $\rho_{\mathcal{B}}$ s'obtient par la réunion des dilatés $Y \oplus B$ pour toutes les paires (B, Y) de $\rho_{\mathcal{B}}$. Alors, $\gamma_{\mathcal{B}}(X) = \tau_{\mathcal{B}}(\rho_{\mathcal{B}}(X))$. De surcroît, pour tout $B \in \mathcal{B}$, $(X \circ B) \ominus B = \gamma_{\mathcal{B}}(X) \ominus B = X \ominus B$. Et donc $\rho_{\mathcal{B}}(\gamma_{\mathcal{B}}(X)) = \rho_{\mathcal{B}}$. ■

Définition d'une représentation

De la discussion précédente, il découle que la représentation $\rho(X)$ se compose de l'élément structurant B et de l'ensemble érodé $X \ominus B$. La recomposition s'obtient par dilatation de l'ensemble érodé et l'approximation résulte de l'union des ouvertures.

Définition 68 *En résumé, les trois éléments suivants suffisent à une description cohérente de la forme de l'objet X unique :*

– représentation $\rho(X)$:

$$\rho(X) = \{(B, X \ominus B) | B \in \mathcal{B}\} \quad (7.9)$$

– recomposition τ :

$$\tau(B, Z) = \{Z \oplus B | Z \subseteq \mathbb{R}^2 \text{ et } B \in \mathcal{B}\} \quad (7.10)$$

– approximation $\pi(X) = \tau.\rho(X)$:

$$\pi(X) = \bigcup \{Z \oplus B \mid (B, Z) \in \rho(X) \text{ et } B \in \mathcal{B}\} \quad (7.11)$$

Une démarche concrète de recherche de représentation optimale part du principe qu'il faut à tout prix une approche hiérarchique dans la description de l'image, cette hiérarchisation étant obtenue en structurant la famille $\mathcal{B}$ en termes de taille des éléments structurants. Elle est à la base d'un algorithme descriptif proposé par RONSE et MACQ [82] et étudié en profondeur dans [80]. Comme nous le verrons plus loin, notre représentation est plus souple, car elle impose une hiérarchie uniquement au moment du codage et pas vraiment durant la phase de recherche de la représentation. En soi, l'étude complète les travaux précités, mais elle résout aussi de nombreuses questions demeurées en suspens. En fait, la représentation de plusieurs objets a été revue.

7.3.2 Représentation d'une collection d'objets

La représentation d'une collection d'objets exploite le *recouvrement géométrique des objets déjà codés*.

Soit à coder plusieurs objets ordonnés séquentiellement $X_0, X_1, \dots, X_M$; nous avons ajouté la région X_0 à $\rho(X)$ pour initialiser le processus. Après avoir codé les objets $X_0, \dots, X_{k-1}$ ($k > 1$), il n'est pas interdit que la description de X_k déborde sur $X_0, \dots, X_{k-1}$. Plus formellement, posant $P = X_0 \cup \dots \cup X_{k-1}$, l'incrément servira à coder la partie utile de l'ensemble $P \cup X_k$. Par le recouvrement géométrique des régions, on nourrit l'espoir d'identifier B plus facilement à l'intérieur de $P \cup X_k$. On peut encore franchir un pas supplémentaire en incluant dans P , le bord S de l'image. Dès lors, la séquence d'objets $X_0, X_1, \dots, X_M$ d'une image segmentée est initialisée en prenant toujours pour X_0 un bord dont l'épaisseur sera égale à la plus grande dimension des éléments structurants.

Le recouvrement géométrique des régions exige en parallèle l'introduction d'un ordre de codage des régions. Physiquement, la relation d'ordre traduit qu'un élément, qui est somme toute un objet au sens morphologique, choisi après un autre est caché par le précédent. Il n'y a, à l'heure actuelle, pas encore de véritable solution au délicat problème de l'ordre des régions. Néanmoins, dans le cadre de la réduction de représentation et du codage de la forme, nous proposons un nouveau critère d'ordre appelé *critère de performance*.

Tous les détails algorithmiques détaillés plus loin concernent une représentation où le recouvrement est autorisé. Cela ne signifie nullement qu'il faille tout récrire pour une représentation interdisant le recouvrement. Pour réaliser le passage, il suffira de remplacer l'expression $P = X_0 \cup \dots \cup X_{k-1}$, c'est-à-dire la totalité de la surface déjà représentée au moment de représenter la région X_k , par l'ensemble vide : $P = \emptyset$.

7.4 Simplification de la représentation

Une première simplification. Pour éviter la redondance de la représentation qui apparaît pour une description de forme par une famille $\mathcal{B}$ suite au recouvrement géométrique de certains éléments structurants, tout en conservant une certaine variété de forme et de

taille impossible à conjuguer dans une approche classique, il est possible d'obtenir une représentation simplifiée grâce aux deux mécanismes décrits ci-après :

1. *Ordre de sélection variable des éléments structurants.*

Nous choisissons une famille maximale composée de plusieurs éléments de différentes tailles. Étant entendu qu'un élément structurant est ajouté par étape dans $\rho(X)$, l'ordre de sélection des éléments à ajouter dans chacune des familles intermédiaires $\mathcal{B}_i \subseteq \mathcal{B}$ n'est pas fixé une fois pour toutes ; il est *adaptatif*. Moyennant un critère adéquat, un algorithme décide à chaque étape quel élément est le plus souhaitable pour une description peu redondante. C'est ici qu'apparaît l'exigence de progressivité. Un ordre de priorité de sélection s'installe alors.

Soient $\mathcal{B}_p \subseteq \mathcal{B}$ une famille de priorité p et B_{pl} un élément l de cette famille. L'ordre de sélection impose qu'un élément structurant $B_{(p-1)l}$ de priorité inférieure n'est inséré dans la famille $\mathcal{B}$ que lorsque tous les éléments de priorité p sont compris dans la famille, autrement dit, $B_{(p-1)l}$ est choisi si et seulement si $\mathcal{B}_p \setminus \mathcal{B} = \emptyset$. La hiérarchisation introduite se situe entre une extrémité souple (tous les éléments ont une même priorité) et une autre, sévère (chaque élément structurant de $\mathcal{B}$ a une priorité différente).

Vers un critère indiquant le choix entre des éléments B de même priorité.

Le critère choisi dans [82] dit que l'élément à injecter est celui qui donne le plus fort incrément dans l'approximation, dans la famille de plus haute priorité où un incrément est possible. Plus formellement, si $X(i)$ est l'image approximée à l'étape i et, si C et D sont les deux candidats, le critère sélectionnera C si l'aire de $(X \circ C) \setminus X(i)$ est supérieure à $(X \circ D) \setminus X(i)$, et choisira D dans le cas contraire. La représentation découlant de ce simple algorithme est très redondante et convient peu à un codage. Toutefois, nous évitons de la simplifier dès maintenant. La représentation réduite proposée plus loin explore les possibilités de réduction de redondance.

La représentation $\rho(X)$ est un ensemble ordonné de triplets.

In fine, l'image se représente par un ensemble *ordonné* de triplets (n, x, y) ⁵ comprenant respectivement le numéro de l'élément structurant retenu parmi tous ceux contenus dans la palette $\mathcal{B}$ et les coordonnées ; c'est la description de la partie utile de l'ensemble érodé.

La question liée à la simplification est le choix des éléments structurants puisqu'il faut fournir des approximations intéressantes des différents objets. Si la famille contient peu d'éléments, la description n'est pas très précise. Par contre, choisir une grande famille accroît la redondance de la description. Les réponses traditionnelles à cette question sont de trois types (cf. [82]) :

- (a) La première revient à choisir expérimentalement une famille restreinte d'objets structurants. À défaut de critère analytique de choix, c'est la seule solution possible.

5. Rigoureusement, les éléments de la représentation sont des quadruplets (n, x, y, o) où o est un numéro d'ordre. Ce numéro est nécessaire si on ne donne pas l'ordre des triplets. Comme il est désormais sous-entendu que la suite des triplets est ordonnée, o devient inutile.

- (b) La deuxième choisit un élément de forme fixée mais dont la taille est adaptée à plusieurs échelles. Le codage par squelette morphologique construit sa représentation de cette manière. Ce style de solution convient pour une description hiérarchique mais elle risque de mener à de mauvais résultats si l'image X est de forme fort différente de B : une série de cercles décrit mal les coins d'une figure.
- (c) Une dernière approche sélectionne a priori une série d'objets de formes diverses (cercle, carré, losange, ...) et produit les familles successives par l'adjonction de ces mêmes éléments, à des tailles de plus en plus fines. Cette solution est souvent plus complexe que la précédente et risque de conduire à une description théoriquement plus redondante, mais elle décrira plus facilement une figure de forme quelconque.

2. *Description de l'objet par le choix de l'élément structurant et de sa position dans l'image en évitant toute redondance de représentation due au recouvrement géométrique.*

L'approximation $X(i)$ de l'ensemble X est obtenue, à partir de $X(i-1)$, par ajout de $X \circ B(i)$ avec un élément structurant $B(i)$ inséré à l'étape i . Comme la représentation $\rho(X)$ comprend les éléments structurants utilisés pour décrire X et leurs positions, la nouvelle description $X(i)$ associerait $B(i)$ à l'érosion $X \ominus B(i)$. Mais une telle description est redondante parce que toute une partie des points reconstitue un morceau déjà couvert dans l'approximation $X(i-1)$. C'est pourquoi, ce n'est pas $X \ominus B(i)$ qui est associé à B , mais l'ensemble des positions p telles que $B(i)_p$ soit inclus dans X , sans l'être dans $X(i-1)$. Il s'agit de la différence ensembliste

$$(X \ominus B(i)) \setminus (X(i-1) \ominus B(i)) \quad (7.12)$$

Définition 69 *La nouvelle représentation $\rho_s(X)$ est une version simplifiée de la représentation de base. Elle mène à trois ensembles :*

- la représentation $\rho_s(X)$. Soit i ($1 \leq i \leq I$) une étape quelconque de travail, $\rho_s(X)$ vaut

$$\{(B, [X \ominus B(i)] \setminus [X(i-1) \ominus B(i)]) \mid B(i) \in \mathcal{B} \text{ et } 1 \leq i \leq I\} \quad (7.13)$$

avec $X(0) = \emptyset$,

- la recomposition τ :

$$\tau(B, Z) = \{Z \oplus B \mid Z \in \mathbb{R}^2 \text{ et } B \in \mathcal{B}\} \quad (7.14)$$

- l'approximation $\pi(X) = \tau.\rho_s(X)$:

$$\pi(X) = \bigcup \{Z \oplus B \mid (B, Z) \in \rho_s(X) \text{ et } B \in \mathcal{B}\} \quad (7.15)$$

L'approximation obtenue au moyen de la représentation simplifiée est identique à la représentation de base, qui entraîne un recouvrement géométrique excessif. Contrairement à l'approche de PITAS, tout recouvrement géométrique n'est pas interdit, permettant la

synthèse d'une représentation plus compacte en vue d'un codage ; il arrive souvent qu'en autorisant le recouvrement, le nombre d'éléments structurants diminue. Dans un certain sens, l'expression de la représentation diffère de celle construite par RONSE et MACQ. Ceux-ci exigeaient que la sélection des éléments se fasse par famille entière au sein de celles ayant même priorité. La raison latente est qu'ils désirent obtenir des listes de positions pour un seul élément structurant. La représentation devient alors en extension :

$$\rho(X) = \{B_i, (x_{i1}, y_{i1}, x_{i2}, y_{i2}, \dots), B_j, (x_{j1}, y_{j1}, x_{j2}, y_{j2}, \dots), \dots\} \quad (7.16)$$

Cette démarche est trop contraignante parce que rien n'empêche de laisser un libre choix durant la phase algorithmique et d'opérer le tri en listes à la fin. La question est apparemment plus délicate si deux candidats apportent par leur ajout un même incrément de surface. Ce n'est pas vraiment le cas car une règle simple –trancher en prenant l'élément déjà le plus utilisé dans la représentation– résout les difficultés.

Les deux mécanismes de simplification précités sont destinés à réduire le recouvrement géométrique qui survient lors de l'ajout d'un type d'élément structurant dans $\mathcal{B}$. Or, $X \ominus B(i)$ contient manifestement des points inutiles. En contrepartie, d'autres points de l'érodé sont *essentiels*, en ce sens que la dilatation de tous les autres points de l'érodé ne permet pas d'obtenir le même dilaté que $(X \ominus B(i)) \oplus B(i)$. On peut définir l'*érodé essentiel* comme le plus petit ensemble tel que son dilaté soit égal à $(X \ominus B(i)) \oplus B(i)$; cet ensemble n'est pas unique. Nous proposons ci-après un *algorithme capable de déterminer un érodé essentiel à partir de l'information de polymorphisme*. Cet algorithme exploite le résultat du théorème 59, fourni à la page 161, qui affirme que tout ensemble $X \circ B$ non vide de $\mathbb{R}^2$ contient plusieurs points dont le polymorphisme pour B vaut 1. La figure 7.5 détaille cet algorithme sur un exemple simple.

L'algorithme de construction de l'érodé essentiel travaille en plusieurs phases. Soit à remplir X par l'élément B :

1. *Initialisation*. On construit l'ensemble $Y = X \circ B$ et une liste, vide au départ, pour accueillir les points de l'érodé essentiel.
2. *Phase itérative* :
 - (2.1) Détection dans Y des points p de polymorphisme égal à 1 dans Y .
 - (2.2) À ces points correspondent des éléments q de l'érodé tels que $p \in B_q \subseteq Y$. On place tous les points q dans la liste, puis les positions retenues d'une étape (2.3) antérieure. Il est très facile de déterminer un point q à partir de p . Par exemple, si B est un carré, il suffit d'examiner le 4-voisinage pour savoir dans quel quadrant se trouve q ; sa position exacte découle ensuite de la position de l'origine dans B .
 - (2.3) Le premier point p de la file est retiré. Si $Y \circ B \neq (Y \setminus B_p) \circ B$, on mémorise la position des éléments structurants nécessaires pour recouvrir $(Y \circ B) \setminus ([Y \setminus B_p] \circ B)$. Expérimentalement, il semble que cette étape ne soit pas nécessaire pour recouvrir, en trame carrée, une région par un élément structurant *rectiligne, carré ou en forme de losange*.
 - (2.4) On adapte Y : $Y \leftarrow Y \setminus B_p$. Il faut encore recalculer les polymorphismes le long de la nouvelle frontière. Le processus reprend alors en (2.1) sauf s'il n'y a plus

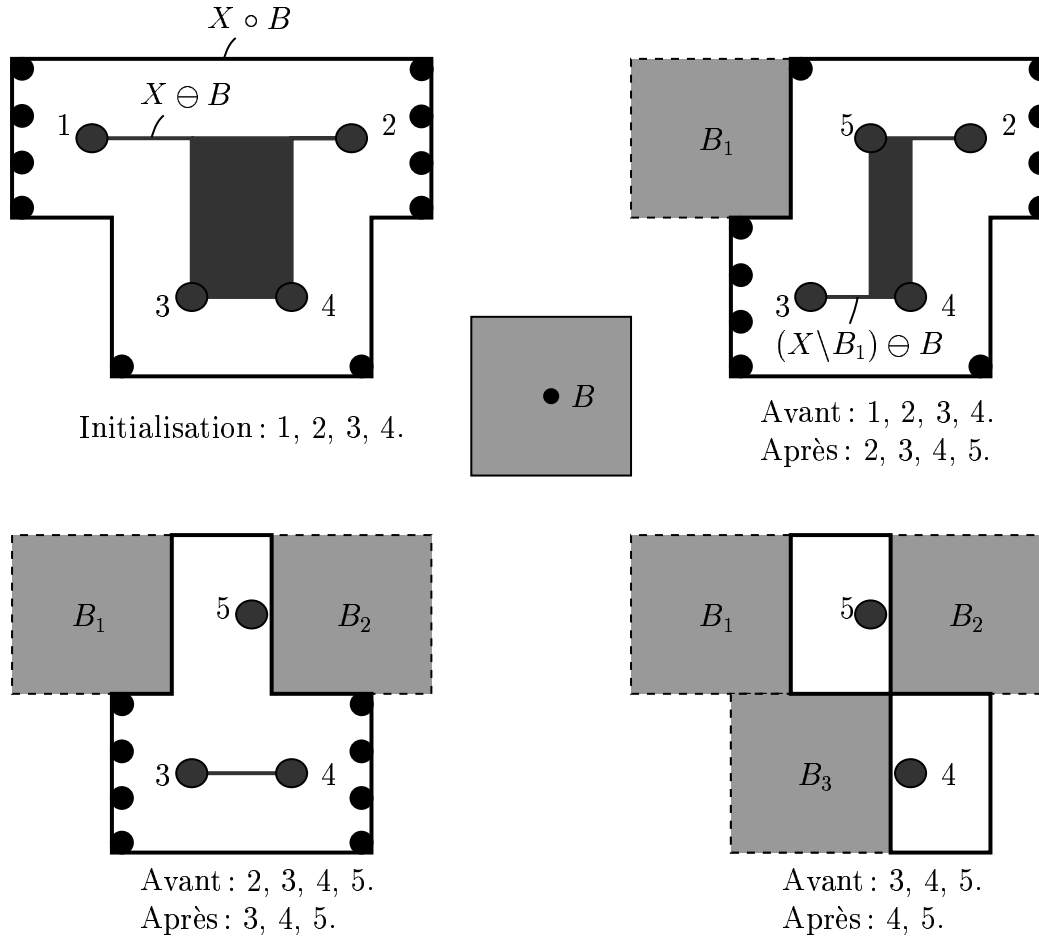


FIG. 7.5 - Détermination des points essentiels de l'érodé pour remplir l'ouvert. En examinant les points dont le polymorphisme vaut 1 (indiqués par des disques noirs), l'algorithme sélectionne les points correspondant de l'érodé; ce sont les points p tels que B_p recouvre au moins un disque noir. Au départ, il y a 4 points essentiels; ils sont placés dans une file. Pour déterminer un recouvrement optimal, on extrait le premier point de la file et X est remplacé par $X \setminus B_1$. Deux points de polymorphisme unitaire apparaissent dans ce dernier ensemble. Seul celui du haut contribue à définir un nouveau point essentiel de l'érodé: le point numéroté 5 est inséré dans la file. Le processus est itéré jusqu'à ce qu'il n'y ait plus de point de polymorphisme unitaire, ce qui signifie que l'ouvert du dernier ensemble obtenu est vide.

de point de polymorphisme unitaire dans Y , ce qui signifie que $Y \circ B = \emptyset$. L'ensemble des points qui ont appartenu ou appartiennent à la file constitue un érodé essentiel.

Cet algorithme est utilisé plus loin pour réduire la représentation de forme simplifiée.

7.5 Réduction de la représentation : vers une représentation semi-externe autorisant le recouvrement de régions

7.5.1 Objectifs de la réduction : organisation et simplification supplémentaire de la représentation

La représentation de base $\rho(X)$ permet déjà une interprétation intéressante. De la notion d'image de labels, nous sommes passé à une description de la forme des objets en termes d'opérations morphologiques. Si la représentation simplifiée réduit déjà $\rho(X)$, elle nécessite des modifications ou plutôt des transformations successives pour un codage. Ces transformations réorganisent la représentation $\rho_s(X)$ pour supprimer toute redondance d'information évidente et la simplifier davantage. Au terme de cette suite d'adaptations, la transformation est *réduite*. C'est la transformation réduite $\rho_r(X)$ qui est placée à l'entrée du codeur entropique et envoyée dans le canal.

Il est utile de rappeler que l'opération de réduction n'affecte nullement l'approximation. Avec la recomposition adéquate, les approximations $\tau_s \cdot \rho_s(X)$ et $\tau_r \cdot \rho_r(X)$ sont identiques.

7.5.2 Description interne ou externe?

La représentation morphologique des objets est une description interne, c'est-à-dire de la surface. Or, il est paradoxal de représenter toute la surface alors que la seule information pertinente est la frontière de l'objet. La démarche que nous adoptons, combine les idées des descriptions internes et externes ; nous la qualifions parfois de *semi-externe*. L'intention n'est plus de recouvrir complètement la surface de l'objet $P \cup X_k$ par une famille d'éléments structurants, mais bien de coder les *nouveaux contours* au moyen des contours des éléments de $\mathcal{B}$, placés à l'intérieur de $P \cup X_k$. La forme d'un objet, dont tous les contours sont représentés, est complètement déterminée. Tout au plus faut-il préciser si l'intérieur non recouvert par des éléments structurants est ou n'est pas une autre région ; on peut aussi supposer que $P \cup X_k$ recouvre tous les trous dans X_k , comme dans [82], mais l'imposer introduit une contrainte inutile. Cette collusion originale explique pourquoi les expressions de 'représentation interne' et de 'codage des contours' sont volontairement confondues ailleurs dans le chapitre. À titre d'exemple, nous avons dessiné une description semi-externe d'un objet X à la figure 7.6.

7.5.3 Lignes directrices

Pour réduire au mieux la représentation, nous proposons de déterminer un *ordre optimal pour le choix des éléments structurants et pour celui des régions*. La détermination

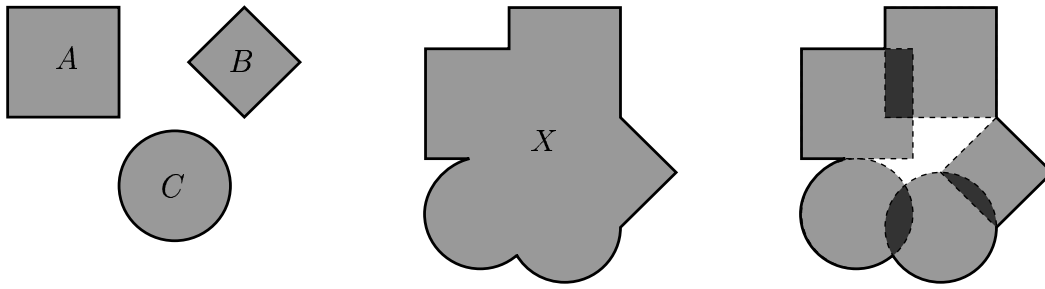


FIG. 7.6 - Exemple de description semi-externe d'une région. Le troisième dessin est obtenu en plaçant des éléments A , B ou C le long de la frontière intérieure de X . La partie foncée indique les endroits où les éléments se recouvrent. Dans une description semi-externe, il n'est pas nécessaire de représenter l'intérieur de X (partie blanche); on économise ainsi un ou plusieurs éléments.

de cet ordre repose sur des *critères* qui font l'objet des pages suivantes. La réduction est ensuite complétée d'une *réduction de l'entropie des couples* (x, y) intervenant dans la *représentation*. La réduction d'entropie exploite le remplissage progressif de l'image.

La réduction de la représentation repose sur plusieurs points importants :

1. *Critère de sélection des éléments structurants de priorité identique.*

La famille servant à la description ρ_s contient des éléments structurants de différentes priorités. Si entre éléments de priorité distincte l'ordre est préétabli, il en va tout autrement pour le choix des éléments B_{pl} et B_{pm} d'une famille de priorité fixée B_p . Pour une description interne, un critère de sélection s'impose : l'élément choisi est celui qui apporte le plus grand incrément de surface. Pour une description semi-externe, on choisit un incrément de périmètre.

Approximations successives par ajout d'une famille d'éléments structurants.

Il est évident que le meilleur choix de famille d'éléments structurants pour décrire une région est celui qui décrit le plus de points de contours avec le moins d'éléments structurants. Dès lors que la priorité p est fixée, la famille d'éléments structurants B_{pl} à ajouter aux précédentes maximise une fonction de performance F , où interviennent le nombre de nouveaux points de frontière représentés par B_{pl} et le nombre d'éléments structurants nécessaires $n[B_{pl}]$

$$F(\mathcal{A}[\{\partial(P \cup X_k) \setminus \partial P\} \cap \{(P \cup X_k) \circ B_{pl}\}], n[B_{pl}]), \quad (7.17)$$

dépendante du nombre d'éléments de frontière représentés par B_{pl} , introduits par l'ajout du dernier objet à coder.

Une détermination arbitraire de la fonction conduit à proposer un critère de sélection que nous appelons *critère de performance* :

Définition 70 *Le critère de performance vaut :*

$$F(B_{pl}) = \frac{\mathcal{A}[\{\partial(P \cup X_k) \setminus \partial P\} \cap \{(P \cup X_k) \circ B_{pl}\}]}{n[B_{pl}]} \quad (7.18)$$

Avec ce critère, B_{pl} est préféré à B_{pm} si $F(B_{pl}) > F(B_{pm})$.

Nous sommes conscient que ce critère fournit un ordre de sélection raisonnable pour la phase de réduction mais peut-être pas pour le codage. Deux attitudes répondent à l'objection : (i) différents critères sont essayés et celui qui paraît idéal est choisi, ou (ii) un critère plus général considérant aussi la phase de codage remplace le critère de performance.

Expérience. Les quelques essais réalisés avec ce critère sur des formes simples, ont montré que ce critère de performance offre une première solution intéressante au délicat problème de sélection des éléments structurants. Les expériences avec différents critères concluent en faveur du critère de performance énoncé plus haut ; ce critère mène généralement au plus faible débit toutes choses égales d'ailleurs, mais en l'absence d'une justification analytique, nous n'oserions certifier qu'il est le meilleur.

Approximations successives par ajout d'un élément structurant d'une famille quelconque.

Comme indiqué précédemment, rien ne nous contraint à insérer les éléments structurants par familles entières dans la représentation d'un objet ; le regroupement en listes a lieu seulement à la fin de la recherche, pas pendant la phase de représentation. Cette liberté supplémentaire revient à choisir une expression simplifiée du critère de performance. Le *critère de performance simplifié* se déduit de l'équation 7.18 ; il s'exprime par

$$F(B_{pl}) = \mathcal{A}[\{\partial(P \cup X_k) \setminus \partial P\} \cap \{(P \cup X_k) \circ B_{pl}\}] \quad (7.19)$$

Au même titre que la question de savoir s'il est intéressant ou non de recourir au recouvrement de régions, nous pouvons construire les approximations successives par ajout de familles entières ou en incluant à chaque étape l'élément qui convient le mieux parmi les familles de même priorité. Si *les algorithmes décrits ci-après traitent le cas d'approximations par ajout de familles complètes*, les modifications à apporter sont mineures. Pour éviter d'alourdir le texte, nous ne décrirons qu'une seule situation.

Comment obtenir l'expression d'une représentation réduite ?

Il ne suffit pas de sélectionner un bon ordre pour les éléments structurants ; il faut encore fournir un algorithme qui détermine la représentation.

En morphologie mathématique, la démarche utilisée pour décrire une région consiste à simplifier une représentation surabondante (cf. squelette minimal). L'objectif à atteindre alors est la simplification de l'érodé de sorte à laisser inchangée l'approximation et de minimiser le cardinal de l'ensemble érodé réduit. Notre objectif est identique puisque cette condition est la plus favorable pour augmenter la performance, étant acquis que B_{pl} est fixé et que, dès lors, le numérateur du critère de performance ne change plus. Cependant l'algorithme prend un chemin inverse : introduire progressivement les positions de l'érodé jusqu'à ce que l'approximation soit la bonne. À chaque étape, la position insérée est celle qui représente le plus grand

nombre de points de contours et s'il reste plusieurs candidats, l'élément retenu est celui qui minimise le recouvrement géométrique.

Tous les éléments sont en place pour décrire l'*algorithme de représentation d'une région unique Z par une famille d'éléments structurants quelconques.*

Représentation d'un objet unique.

- *Notations*

- * Z = objet à représenter.
- * Y, Y_i, Y_{ii} = ensembles intermédiaires de calcul, relatifs aux contours. À chaque itération, Y contient les points de contours de Z non encore représentés et Y_i, Y_{ii} les points de contours qu'il est possible de représenter avec l'élément structurant en question.
- * p = priorité associée à une sous-famille $\mathcal{B}_p \subseteq \mathcal{B}$ d'éléments structurants; p varie entre 1 et P et $\bigcup_{p=1}^P \mathcal{B}_p = \mathcal{B}$.
- * $n[p]$ = nombre d'éléments structurants différents inclus dans $\mathcal{B}_p$.
- * B_{pl} = élément de la sous-famille de priorité p ; $\bigcup_{l=1}^{n[p]} B_{pl} = \mathcal{B}_p$.
- * $T[.]$ = ensemble de triplets associé à un élément structurant.

- *Algorithme de représentation de Z*

Initialisation: $Y \leftarrow \partial Z$

Pour $1 \leq p \leq P$, p décroissante et tant que $Y \neq \emptyset$ (boucle sur les priorités)

{

$\mathcal{B}'_p \leftarrow \mathcal{B}_p$

Tant que $\mathcal{B}'_p \neq \emptyset$

{

- Choix de l'élément structurant: pour chaque $B_{pl} \in \mathcal{B}'_p$,

$Y_i = Y \cap \partial(Z \circ B_{pl})$,

$T[B_{pl}] = \emptyset$

Évaluation de $n[B_{pl}]$:

★ initialisation: $Y_{ii} = Y_i$

★ le point (i, j) à choisir est celui tel que le dilaté représente le plus de points, autrement dit $(i, j) = \max_{(x,y) \in \mathcal{S}} \mathcal{A}[\partial[(x, y) \oplus B_{pl}] \cap Y_i]$

Si plusieurs positions (a, b) et (c, d) fournissent la même valeur de maximum, on choisit par exemple la première balayée ou celle qui minimise le recouvrement géométrique:

$(i, j) \leftarrow (a, b)$ si $\mathcal{A}[(Z \setminus \{(a, b) \oplus B_{pl}\}) \circ B_{pl}] \geq \mathcal{A}[(Z \setminus \{(c, d) \oplus B_{pl}\}) \circ B_{pl}]$

$(i, j) \leftarrow (c, d)$ dans le cas contraire

★ $T = T \cup \{(B_{pl}, i, j)\}$

★ Mise à jour de Y_{ii} : $Y_{ii} \leftarrow Y_{ii} \setminus (Y_{ii} \cap \partial[(i, j) \oplus B_{pl}])$

La recherche des points se poursuit jusqu'à ce que $Y_{ii} = \emptyset$

- Comparaison des performances et choix de l'élément structurant idéal:

l'élément optimal retenu B_{pl} est celui qui maximise le critère de performance B_{pl} où $l = \max_{m | B_{pm} \in \mathcal{B}'_p} F(B_{pm})$

- Mise à jour des ensembles

$Y \leftarrow Y \setminus Y_i$

$$\begin{aligned}
& \mathcal{B}'_p \leftarrow \mathcal{B}'_p \setminus B_{pl} \\
& \rho_r(Z) \leftarrow \rho_r(Z) \cup T(B_{pl}), \text{ expression de la représentation réduite} \\
& \} \\
& \}
\end{aligned}$$

Du point de vue de l'implémentation, il faut un balayage de la partie utile de l'image pour obtenir une position (i, j) . Nous utilisons pour cela l'algorithme d'érosion développé au chapitre 5 sur une image mise à zéro, sauf à l'intérieur de X et le long de la frontière où les points prennent respectivement les valeurs 1 et 2. Soit $h(n)$ la fréquence de l'histogramme en n . En tout point de l'érodé, on met zéro dans l'image si $h(0) \neq 0$ et la valeur $h(2)$ dans le cas contraire. Il suffit alors de repérer la première plus grande valeur de l'image de sortie pour déterminer la position recherchée. Ce mécanisme est illustré à la figure 7.7.

Après avoir inséré l'information (B_{pl}, i, j) dans la représentation, l'algorithme met à jour les valeurs de l'érodé *uniquement dans une fenêtre* $B_{pl} \oplus \check{B}_{pl}$ *centrée en* (i, j) .

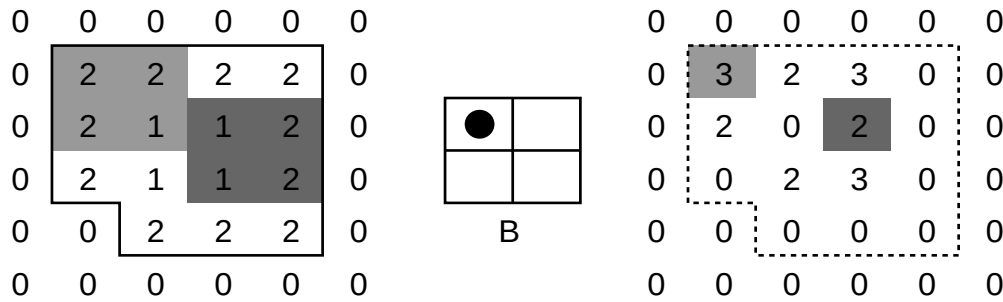


FIG. 7.7 - Comment déterminer le nombre de points frontières correspondant à chaque élément de l'érodé? Il suffit de mettre la valeur 1 à l'intérieur de l'objet et 2 le long de la frontière intérieure. On affecte ensuite la fréquence de l'histogramme pour $n = 2$ à tous les points de l'érodé.

L'algorithme décrit vise à réduire le nombre d'éléments structurants nécessaires pour la représentation d'un objet unique. Il reste d'application pour une collection d'objets dont l'ordre est fixé à l'avance.

Nous levons la contrainte d'ordre préfixé de représentation des régions au point suivant. Pour rappel, dans la mesure où le recouvrement entre régions est interdit, l'ordre de représentation des régions n'a plus guère d'importance.

2. Ordre variable de représentation des régions – Critère de performance étendu.

La représentation réduite d'un objet unique Z vient d'être étudiée en profondeur. Pour obtenir la totalité de la représentation réduite, l'algorithme est appliqué à des ensembles $P \cup X_k$ incluant région après région.

Jusqu'à présent, l'ordre de représentation des objets était supposé connu, par exemple de la plus grande région vers la plus petite ou dans l'ordre inverse pour profiter au mieux du recouvrement géométrique des régions pour la représentation des grandes régions. Pour déterminer l'ordre de manière automatique, nous avons imaginé un

critère qui s'inspire du critère de performance et considère l'organisation spatiale des régions. Cette organisation est modélisée par un graphe.

La description spatiale tirée du graphe est si importante que nous en arrivons à nous demander s'il ne serait pas bon d'inclure la totalité du graphe dans la représentation pour l'interprétation, l'efficacité de codage, et pour la réduction extrême de représentation. Contentons-nous pour le moment du graphe comme support algorithmique.

Soient X_i et X_j deux régions de l'image. Le graphe peut être caractérisé par sa *matrice d'adjacence* $A = (a_{ij})$, de dimension $M \times M$, où

$$a_{ij} = \begin{cases} 1 & \text{si } (X_i, X_j) \in R \\ 0 & \text{si } (X_i, X_j) \notin R \end{cases} \quad (7.20)$$

où R est la relation de voisinage définie à la page 166 (définition 63).

Deux régions X_k et X_l sont voisines si $a_{kl} = 1$. De plus, on prend $a_{ii} = 1$. Enfin, quand une région X_k a été traitée, on annule la colonne a_{ki} et la ligne a_{ik} qui lui correspondent. Le graphe joue un rôle capital durant la recherche algorithmique. En effet, par le choix d'un mode de représentation semi-externe, il y a le risque que tous les contours soient représentés doublement. Dans les algorithmes précédents, l'inconvénient est contourné en ne considérant que les points de contours voisins d'une région non encore représentée. La mise à jour régulière de la matrice d'adjacence, après chaque représentation d'une région, sert de support algorithmique. Nous l'utiliserons pour deux fonctions :

- (a) *Détection des régions codées implicitement.* Durant la phase de représentation d'une collection de régions, il arrive qu'une région soit entourée uniquement de régions déjà représentées, auquel cas sa forme est connue *implicitement*. Soit X_k la région en question. Dans la matrice d'adjacence, il apparaît que $a_{kk} = 1$ et $a_{ik} = a_{ki} = 0$ pour tout $i \neq k$.
- (b) *Modélisation commune à l'émetteur (phase de représentation) et au récepteur (phase de reconstruction).* Parfois, le graphe se découpe en deux sous-graphes après la représentation d'une région. Cela provient soit (i) parce que tous les contours de la région ont été représentés mais pas une partie de l'intérieur (cf. figure 7.6), ou soit (ii) lorsqu'une région se situe entre deux ensembles de régions séparés. D'autre part, durant la reconstruction, le récepteur reconstruit le graphe au fur et à mesure. Il détecte à un certain moment que le nombre de connexité de la surface non recouverte du plan a augmenté⁶. Ayant décidé une fois pour toutes de traiter d'abord l'un ou l'autre des sous-graphes, il suffit de très peu d'information pour que le récepteur interprète correctement le phénomène ; il y a trois messages possibles lors de l'apparition d'un sous-graphe supplémentaire :
 - (1) la surface est un morceau de la région précédente (partie de la surface non représentée de la région),
 - (2) il s'agit d'une région représentée implicitement et

6. La traduction immédiate de l'apparition de deux sous-graphes est bien entendu l'augmentation du nombre de connexité de la surface restante.

(3) le sous-graphe est un amalgame de plusieurs régions.

Traiter chaque sous-graphe à tour de rôle offre un avantage supplémentaire : celui de réduire l'incertitude quant à la position des éléments structurants à inclure dans la représentation. Et réduire l'information est synonyme de réduction d'entropie ou encore d'augmentation du taux de compression. On perçoit mieux à présent l'utilité du graphe. C'est un outil de modélisation très efficace et il interprète l'importante relation de voisinage à peu de frais.

Tous les éléments sont en place pour la définition d'un critère destiné à ordonner les objets. C'est le critère que nous appelons *critère de performance étendu* étant donné sa ressemblance avec le critère simple.

Définition 71 Soit $X_1, X_2, \dots, X_M$, une série de M régions à représenter. Le critère de sélection étendu sélectionne la région X_l qui maximise la fonction

$$F_e(X_l) = \frac{\mathcal{A}[\partial X_l \cap (X_l \circ \mathcal{B})]}{n[\mathcal{B}]} \quad (7.21)$$

dans laquelle la famille d'éléments structurants $\mathcal{B}$ est fixée.

Le schéma algorithmique de représentation réduite d'un ensemble de régions est à présent complet. À chaque étape, il choisit la région dont la performance est la plus élevée.

Algorithme de réduction de représentation $\rho(X)$ de plusieurs objets

- *Algorithme*

Initialisation :

$P \leftarrow S$, le bord de l'image.

$\mathcal{X} \leftarrow \{X_1, \dots, X_M\}$

Tant que $\mathcal{X} \neq \emptyset \Leftrightarrow \sum_{k=1}^M a_{kk} \neq 0$

- {

Choix de la région X_k à coder :

c'est la région telle que $a_{kk} = 1$ et $F_e(X_k)$ est maximum

- Représentation de la région X_k : par l'algorithme de représentation d'objet unique est formé l'ensemble des triplets $T(X_k)$ comprenant la position de référence et l'élément structurant choisi dans la famille $\mathcal{B}$

- Évolution de la représentation réduite :

$\rho_r(X) \leftarrow \rho_r(X) \cup T(X_k)$

- Mise à jour des ensembles et du graphe :

$P \leftarrow P \cup X_k$

$\mathcal{X} \leftarrow \mathcal{X} \setminus \{X_k\}$

$a_{kl} = a_{lk} = 0, \forall l \in \{1, \dots, M\}$

- Simplification du graphe :

s'il existe une région X_l (voire plusieurs régions) dont tous les voisins sont déjà traités ($\forall j \in \{1, \dots, M\}$ avec $j \neq l$, $a_{lj} = a_{jl} = 0$ et $a_{ll} = 1$), la région est représentée implicitement. Alors $P \leftarrow P \cup X_l$, $a_{ll} = 0$ et $\mathcal{X} \leftarrow \mathcal{X} \setminus \{X_l\}$

Avec cet algorithme, nous disposons d'une puissante représentation réduite qui conserve sa facilité d'interprétation. Pour le codage, il faut encore organiser la représentation réduite $\rho_r(X)$ de manière à diminuer l'entropie des triplets (B, i, j) . Le point suivant s'attache à diminuer l'entropie de la position des éléments structurants.

3. Réduction par diminution de l'entropie des positions.

Pour réduire davantage la représentation, nous suggérons de réorganiser complètement la position comprise dans chaque triplet. L'idée est simple et élégante : après adjonction d'un triplet, la surface représentée croît et, par la force des choses, la surface non représentée diminue. En conséquence, le nombre de positions disponibles pour le couple (i, j) s'amenuise. Pourquoi conserver une dynamique importante de positions, alors qu'au fil du temps, certaines positions deviennent interdites ?

Le mécanisme proposé consiste à *vectoriser les positions possibles et à allouer le point correspondant à (i, j) dans ce vecteur*. Si l'ordre de la représentation est respecté durant la transmission, le récepteur reconstruit sans peine le même vecteur de position et restitue la position (i, j) à partir de la position dans le vecteur. Ainsi, au fur et à mesure de la transmission, la taille du vecteur de positions possibles diminue. Cela revient à adopter une démarche de codage adaptative, qui se sert de la relation d'ordre pour réduire l'entropie de la source.

La considération de l'organisation spatiale apporte encore des améliorations substantielles. Rappelons que le graphe G doublé de sa matrice d'adjacence A traduit les relations de voisinage. L'utilisation du graphe à des fins de diminution d'entropie repose sur l'*élimination temporaire de régions non connexes à la région en cours de représentation*. Plus intuitivement, une région à représenter est entourée d'un cordon de régions déjà représentées ; les régions situées au-delà du cordon ne jouent absolument aucun rôle. En considérant ces régions comme déjà représentées, bon nombre de positions artificielles potentielles sont supprimées. Finalement, la discussion revient à dire que le graphe se partitionne progressivement en sous-graphes disjoints séparés par des cordons de régions représentées. À chaque sous-graphe correspond une vectorisation de positions pertinentes ; la représentation se charge d'indiquer clairement de quel graphe il s'agit.

Réintroduisant explicitement l'ordre dans la représentation, la nouvelle représentation réduite $\rho'_r(X)$ se déduit de $\rho_r(X)$ par le schéma suivant. Les sous-graphes sont traités dans un ordre connu de l'émetteur et du récepteur, par exemple en suivant la taille des plus grands ensembles non représentés. De plus, on suppose que la représentation a été arrangée de sorte que les positions de triplets successifs de même élément structurant soient rangées en progressant de gauche à droite et de haut en bas, l'axe vertical étant orienté vers le bas ; si (B_{kl}, a, b) et (B_{kl}, c, d) se suivent dans la représentation, alors $a + b \times hor < c + d \times hor$, où hor est la taille horizontale de l'image.

Diminution de l'entropie de position dans la représentation réduite.

- *Notations :*

- * I = partie du plan $\mathbb{R}^2$ occupée par les régions à coder.
- * P = bord placé autour de l'image.

- * $P(A, B)$ = surface de l'image étendue, sachant que A est l'ensemble traité et B l'élément structurant. Concrètement, $P(A, B) = A \oplus \check{B}$.
 - * $X_1, \dots, X_M$ = les régions à représenter dans l'ordre, $\bigcup_{k=1}^M X_k = I$.
 - * $G(X_l)$ = le sous-graphe contenant la région X_l .
 - * $nG(X_l)$ = nombre de sous-graphes disjoints avant d'avoir représenté la région X_l .
 - * $Test$ = variable booléenne indiquant si le triplet précédent concerne le même élément structurant.
 - * Y, U_k = ensembles intermédiaires de calcul.
 - * Z = l'ensemble des positions pertinentes possibles.
 - * $B_l(k)$ = élément structurant l intervenant dans la représentation de X_k .
 - * $P_v[C, (i, j)]$ = l'opérateur qui vectorise les éléments de l'ensemble C en parcourant les positions de gauche à droite et de haut en bas. L'opérateur renvoie la position relative x correspondant à (i, j) .
- *Initialisation :*
- $X_0 \leftarrow \emptyset,$
 $nG(X_0) \leftarrow nG(X_1) + 1$
- *Cœur de l'algorithme :*
- Pour $k \leftarrow 1$ à $k = M$ (Boucle sur les régions)
- {
- Calculs propres à la région X_k
- $U_k = \bigcup \{X_n \mid X_n \in G(X_k)\}$
- Si $nG(X_k) > nG(X_{k-1})$, on indique un éventuel changement de sous-graphe,
 $Y = U_k$
- Boucle sur les éléments de représentation de la région X_k
- Pour $l \leftarrow 1$ à $l = L$
- {
- Détermination de la valeur de la variable $Test$ à partir du triplet (u, v, B) précédent
- Si ($Test = non$)
- {
- $Z = P(Y, B_l(k)) = Y \oplus B_l(k)$
- $\rho'_r(X) \leftarrow \rho'_r(X) \cup \{(B_l(k), P_v[Z, (i, j)])\}$
- }
- Sinon ($Test = oui$)
- {
- $Z = P(Y, B_l(k))$
- $\rho'_r(X) \leftarrow \rho'_r(X) \cup \{(B_l(k), P_v[Z, (i, j)] - P_v[Z, (u, v)])\}$
- }
- }
- }

Il n'est pas possible de chiffrer analytiquement le gain apporté par cette réduction d'entropie. En tout état de cause, le gain sera appréciable parce que le nombre de

positions pertinentes décroît rapidement dans le schéma de représentation. Si la représentation a quelque peu perdu de sa simplicité d'interprétation, elle convient davantage à un codage. En effet, l'élimination de positions inutiles correspond à un accroissement de l'information a priori.

7.5.4 Résumé de la méthode et résultats

Méthode. Avant de parler des résultats de codage, le moment est venu de retracer les lignes principales intervenant dans la recherche d'une représentation d'une image de labels. Plusieurs éléments se mêlent dans sa description :

- *Représentation d'une région unique.*
La représentation d'une région unique s'obtient en plaçant des éléments structurants à l'intérieur de manière à obtenir *in fine* le même contour.
- *Représentation d'une collection de régions.*
Pour représenter une collection de régions, il suffit d'appliquer la procédure précédente région après région. Pour du codage, les éléments structurants servant à représenter une région pourront recouvrir les régions déjà traitées. Dès lors que le recouvrement est licite, il y a en contrepartie création d'un concept d'ordre de codage des régions. Il s'accompagne de l'impératif besoin d'un procédé algorithmique de détermination de l'ordre. C'est pourquoi nous avons défini un critère de performance.
- *Définition d'un critère de performance.*
Au vu de la représentation utilisée, les paramètres révélateurs de l'efficacité sont : le nombre de points de contour propres à la région et le nombre d'éléments structurants employés pour sa description. Leur rapport, appelé *critère de performance étendu*, évalue la performance de la représentation de chaque région. À chaque étape, l'algorithme choisit la région ayant la meilleure performance. Après représentation d'une région, il faut recalculer les performances des régions voisines puisqu'une partie de leur contour est représentée indirectement par la région en question.
- *Utilisation d'un graphe comme support de modélisation de l'organisation spatiale des régions.*
L'évaluation des performances exige la connaissance des relations de voisinage entre les régions. Pour traduire l'agencement spatial des régions, nous avons doublé la représentation d'un graphe de voisinage de régions. Plus qu'un simple modèle, il s'agit d'un outil algorithmique puissant dont la gestion ne coûte que bien peu de bits.
- *Réduction entropique des positions grâce au remplissage progressif du plan.* Au fur et à mesure que la représentation se fige, le nombre de positions possibles pour tout ajout d'élément structurant diminue. Alors, les positions restantes, associées à un élément structurant, sont transformées en un équivalent linéaire en suivant un balayage horizontal partant du coin supérieur gauche.

Résultats de codage

Dans un premier temps et avec l'aide de Benoît MACQ et de Benoît SIMON, nous avons tenté de coder les points du squelette des régions d'une image par la méthode de FREEMAN, après avoir élargi le nombre de directions pour permettre des sauts entre les branches du squelette. Les résultats de ces essais furent décevants ; le coût de codage est au moins cinq fois plus élevé que l'estimation de 1,3 bit par point de contour.

Malgré toutes les tentatives effectuées, la technique de codage proposée dans ce chapitre ne descend malheureusement pas sous cette estimation. Plutôt que de relater toutes les alternatives essayées pour coder l'image de labels, nous décrivons la méthode qui semble la plus efficace. Les résultats d'une réalisation complète sont fournis après avoir précisé les modalités du codage entropique.

Le codage entropique. La représentation réduite contient une liste de paires ordonnées (N, v) où N est le numéro de l'élément structurant B considéré et v la position obtenue après vectorisation. Les données sont confiées à un codeur entropique dans les termes suivants :

- *Codage des numéros des éléments structurants.*

Pour chaque région, les paires sont ordonnées en commençant par la plus petite valeur de B et suivant un balayage horizontal pour v . Comme la représentation englobe plusieurs régions, la seule difficulté est le passage d'une région à une autre que l'évolution des valeurs de B ne permet pas toujours de détecter. Illustrons cette ambiguïté sur un exemple. Soient une palette $\mathcal{B}$ composée de 4 éléments structurants et la représentation réduite $\rho_r(X) = \{(2, u), (3, v), (3, r), (1, s), \dots\}$. Alors qu'en passant de 3 à 1, il est clair qu'un changement de région a eu lieu, on ne peut tirer une conclusion semblable pour le passage de 2 à 3. Comme parade, nous avons imaginé d'ajouter la valeur du cardinal de $\mathcal{B}$ lorsqu'il y avait un doute possible et de coder les numéros par différence. Ainsi, $\rho_r(X)$ devient $\{(2, u), (5, v), (0, r), (2, s), \dots\}$ ou $\{(2, u), (1, v), (0, r), (2, s), \dots\}$ suivant qu'il y a eu ou non un changement de régions. Les valeurs $(2, 5, 0, 2, \dots)$ ou $(2, 1, 0, 2, \dots)$ sont mises à l'entrée d'un codeur arithmétique et envoyées au décodeur avant la liste des positions de sorte que le décodeur puisse interpréter correctement le flot de données relatif aux positions.

- *Codage des positions.*

Dès lors que le récepteur a reçu la liste des éléments structurants, il est capable de retracer tout l'historique, c'est-à-dire qu'il effectue la même vectorisation qu'à l'émetteur et traite les mêmes sous-graphes. Pour réduire encore l'entropie des positions, nous transmettons le plus grand rectangle qui circonscrit la région dès qu'un changement de région a été détecté. L'émetteur et le récepteur se servent alors du rectangle comme masque.

Les positions $(u, v, r, s, \dots)$, complétées des coordonnées du rectangle, sont également mises à l'entrée d'un codeur arithmétique.

La seule information manquante est liée à la modélisation des régions par un graphe. Toute apparition d'un sous-graphe supplémentaire donne lieu à des indications qui précisent

Éléments structurants	Numéro	Priorité	Nombre d'occurrences
Carré 4x4	1	2	368
Losange de taille 7	2	2	214
Carré 8x8	3	2	48
Carré 16x16	4	2	25
Carré 32x32	5	2	5
Carré 64x64	6	2	3
Carré 2x2	7	1	119
Carré 1x1	8	1	33

TAB. 7.1 - *Description d'une famille d'éléments structurants et nombre d'occurrences de chacun d'eux dans la représentation réduite de l'essai.*

Opération effectuée	Coût de codage
Gestion du graphe	111 bits
Codage des éléments structurants	586 bits
Codage des positions	10984 bits
Total	11681 bits

TAB. 7.2 - *Coût de codage des composantes de notre représentation.*

s'il provient d'une représentation interne et, le cas échéant, si le sous-graphe contient plusieurs régions. 1 bit suffit pour chacune d'elles.

Un codage complet a été réalisé sur l'image de labels représentée à la figure 6.8 (a) (cf. page 157) avec la palette $\mathcal{B}$ détaillée dans le tableau 7.1. Il est bon de remarquer que les éléments de même priorité sont classés par ordre de taille croissante. Dans cette réalisation où les éléments ne sont pas choisis par familles entières, cet ordre favorise le choix d'un petit élément à égalité de performance ; c'est préférable car le nombre de positions possibles pour ce dernier est plus faible.

Le coût de codage des composantes de la représentation réduite est mentionné dans le tableau 7.2. Il apparaît clairement que le codage des positions contribue majoritairement au débit. Diminuer le coût de codage des positions est donc une nécessité si l'on veut concurrencer les autres techniques de codage. Par comparaison, l'estimation du codage par 1,3 bit par point de contour donne 8102 bits.

Conclusions du chapitre

Ce chapitre a décrit une représentation de forme et les transformations à effectuer pour un codage. Le modèle servant à exprimer la position relative des régions est présenté au début du chapitre. Il s'agit d'un graphe bâti sur la relation de voisinage. Ensuite, nous avons présenté une représentation très générale, qui peut convenir à toutes sortes de traitements

d'image par objet. Comme la représentation est très redondante, nous l'avons simplifiée. Différents procédés de simplification ont été proposés pour y parvenir. Il y a par exemple la possibilité de recourir à la notion de polymorphisme pour obtenir un recouvrement optimal. Enfin, dans une phase appelée *réduction*, nous avons réalisé plusieurs modifications de la représentation, sans affecter pour autant la qualité de l'approximation, en vue d'un codage. Ces opérations s'appuient sur plusieurs éléments nouveaux (critère de performance, recherche de sous-graphe, diminution de l'entropie des positions, etc) applicables à d'autres techniques de codage que celles étudiées.

En l'absence de résultats de codage probants, les simulations de l'algorithme de représentation suscitent des réflexions assez nombreuses où questions et perspectives s'entremêlent :

Détermination d'une palette optimale d'éléments structurants.

Il est envisageable de déterminer adaptativement la palette optimale d'éléments structurants; mais comment faire et surtout quel critère utiliser? La détermination d'une palette optimale apporterait une souplesse supplémentaire très agréable pour le codage. Quelques filières se dessinent: recherche exhaustive avec contrainte pour limiter le temps de recherche, analyse des propriétés de convexité des régions et des éléments structurants, ... Le mécanisme d'un codage avec palette optimale comprend deux étapes: transmission de la palette d'éléments structurants et envoi de la représentation réduite et codée.

Description à résolution variable.

Par la séparation de la famille $\mathcal{B}$ en sous-familles de priorités distinctes, la représentation répond à une description de résolution variable. Il suffit de construire des sous-familles composées d'éléments structurants dont les tailles sont des multiples du facteur de sous-échantillonnage pour éviter les difficultés. Face à une description hiérarchique, il y a une autre attitude: sous-échantillonner la représentation de formes, transmettre ce signal et reconstituer la pleine résolution par l'ajout d'un signal détail. Une première méthode consiste par exemple à appliquer l'algorithme après un sous-échantillonnage. Une autre effectue un sous-échantillonnage puis élimine certaines composantes de $\rho'_r(X)$.

Modification partielle du procédé de réduction.

Le procédé de réduction réalise une étape de réduction d'entropie des positions par une linéarisation des positions possibles. Or la forme des régions suggère par exemple de décrire toutes les positions possibles non pas en balayant horizontalement l'image mais en cercles centrés sur les parties de l'objet déjà représentées. Un exemple simple illustre l'avantage de cette description des positions pour un élément structurant. Soit à placer un élément structurant qui est le pixel unitaire. Comme le pixel fait partie de la même région que les éléments déjà transmis auparavant, il viendra obligatoirement se placer sur un point de contour de l'approximation précédente. Il est alors préférable de rassembler les positions potentielles dans une structure en suivant le contour plutôt que de partir du coin supérieur gauche. D'autres considérations de ce genre entraîneront très certainement une diminution de l'entropie des positions possibles. Les quelques essais menés indiquaient une entropie des positions proche

de l'entropie maximale. Quelques règles de connexité connues de l'émetteur et du récepteur, comme celle précitée, pourraient apporter des gains substantiels.

Une perspective : la représentation par squelette propagé

Notre approche de représentation des formes consiste à décrire les surfaces des régions à la manière du squelette morphologique. Le principal défaut du squelette morphologique est sa difficulté d'interprétation. La description qu'il fournit est complète, mais comment coder l'information qu'il recèle ? D'autre part, l'utilisation du squelette morphologique impose une simplification préalable, parce que son contenu est surabondant.

En explorant des nouveaux algorithmes d'ouverture morphologique, nous avons imaginé de représenter la forme d'une région en suivant la trace de la propagation de l'élément structurant à l'intérieur de la région, comme s'il s'agissait de remplir les branches du squelette. Ainsi, on détecte un ou plusieurs homothétiques ancrés à l'intérieur de chaque région à représenter, puis on propage ces éléments dans les régions. Les images de la figure 7.8 montrent les résultats d'une telle approche. La distinction majeure par rapport à l'algorithme d'ouverture par propagation est l'initialisation. Dans un procédé de représentation, il est plus intéressant de commencer par le plus grand rectangle alors que, pour une simple ouverture, un homothétique quelconque suffit. Les rectangles ayant servi d'éléments de départ sont entourés d'un rectangle blanc dans les images de la figure 7.8. On y voit aussi la trace du point milieu de chaque front de propagation horizontal ou vertical, et la trace des coins des propagations en diagonale. Pour l'image du haut, nous avons utilisé le pixel comme élément structurant. L'image du bas a été obtenue en propageant un carré 2x2. Une telle propagation ne permet pas une reconstruction parfaite de la forme des régions mais elle atténue la sensibilité aux irrégularités de forme. En effet, les pattes de la pince sont décrites avec moins de lignes différentes dans le second cas. Pour le codage, on commence par transmettre la position d'un rectangle et ensuite l'information relative au front de propagation. Comme la largeur du front ne peut que décroître au long d'une propagation dans une direction horizontale ou verticale et qu'elle reste confinée dans un cône si la direction est diagonale, il suffit de transmettre la largeur initiale et son évolution. Il est aussi possible d'approximer les lignes par des segments de droite, ce qui conduira à diminuer fortement le coût de codage.

Les éléments de la modélisation des régions développés dans ce chapitre (graphe de voisinage, critère de performance où le nombre de fronts propagés remplace le nombre d'éléments structurants) s'intègrent aisément dans une représentation par propagation d'éléments structurants. Autrement dit, les principaux algorithmes de ce chapitre s'adaptent à cette nouvelle description.

En l'absence d'estimation du coût de codage, il n'est guère possible d'estimer l'efficacité d'une représentation par squelette propagé. Le concept est néanmoins intéressant parce que la propagation procède à un remplissage progressif des régions. À chaque étape, il y a alors une forte réduction d'entropie due à l'occupation spatiale des régions codées et à la possibilité de prédire des extensions maximales pour les fronts. Signalons aussi qu'une station de travail NeXT calcule cette représentation en une seconde pour une image de taille 256x256.

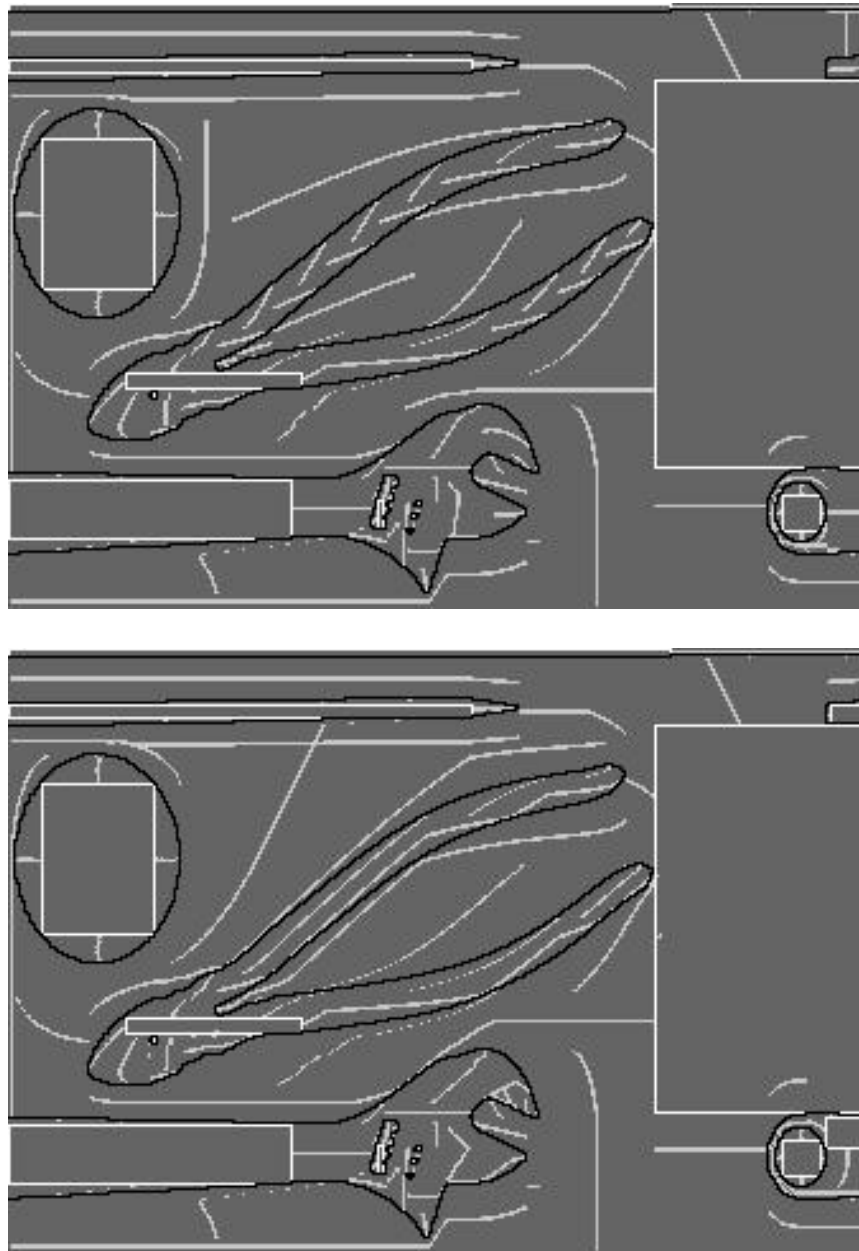


FIG. 7.8 - Représentation de la forme par squelette propagé. La première image résulte d'une propagation avec un pixel; en conséquence, il n'y a aucune déformation des contours. La seconde image a été obtenue en propageant un carré 2x2, ce qui induit une légère imprécision dans la représentation des contours. La signification des couleurs est la suivante: (1) les contours des objets sont représentés en noir, (2) en gris foncé, la surface recouverte par propagation, (3) le squelette propagé est dessiné en clair et (4) les éléments de départ sont les rectangles entourés d'un trait blanc.

Conclusions *

Le développement d'outils de traitement d'image articulés sur la notion d'objet est relativement récent. Dans le domaine du codage apparaissent des techniques de codage par objet, toutes destinées à des fins de compression élevée. Ce travail s'inscrit dans ce courant.

Le premier chapitre a décrit une notation et introduit le *concept de représentation d'image*, cadre général pour tous les traitements d'image. Dans une description par des régions, la représentation se dédouble en composantes que sont la représentation de forme $\rho(X)$ et la représentation de texture $\rho(T)$. Au terme d'une discussion formelle, *nous avons proposé une définition de texture* qui, au moyen d'un procédé d'extrapolation, sépare l'information de texture de son domaine de définition.

Avant d'aborder en longueur la segmentation, nous avons décrit quelques outils de la morphologie mathématique au chapitre 2 et *examiné les contraintes de réalisation*. Cette théorie permet de traiter la forme d'un objet ou d'une fonction en se référant à une forme fixée. Les opérateurs morphologiques interviennent à plusieurs endroits de la chaîne de codage, ce qui justifiait l'étude des opérateurs importants.

Les procédés de représentation spectrale s'adaptent difficilement aux textures parce que le support des régions n'est pas un domaine régulier. Pour pallier cet inconvénient et en conformité avec notre notion de texture, *nous avons imaginé d'étendre les signaux de texture en développant un opérateur généralisé d'extrapolation spectrale*. La convergence et l'unicité de l'opérateur sont établies dans l'annexe A. Outre l'amélioration de l'interprétation spectrale des régions qui en résulte, le procédé d'extrapolation remplit plusieurs fonctions intéressantes : (1) il décorrèle les signaux de contour et de texture, (2) il enrichit le rendu de textures reconstruites avec un nombre limité de raies spectrales, (3) avec un procédé d'extrapolation de textures, transmettre un simple morceau suffit à synthétiser toute la texture, et (4) les textures étendues pourront servir de prédiction des zones découvertes dans une séquence d'images. *Ces différentes utilisations ont été étudiées dans le chapitre 3 sous un angle à la fois théorique et pratique*. Nous avons ensuite chiffré le taux de compression de signaux texturés. Il avoisine 50 pour des grandes régions texturées.

Avec les outils de la morphologie mathématique et celui de l'extrapolation de textures, *nous avons construit un nouveau schéma de segmentation*, à partir d'un modèle d'image basé sur la signature spectrale locale des régions. Le schéma comprend une *phase d'extraction de textures par croissance de germes*, une détection des régions actives et une *segmentation des régions à variations lentes*, toutes trois cohérentes avec le procédé de codage des textures. L'originalité tient tant aux composantes du procédé de segmentation qu'à l'idée d'intégrer des méthodes adaptées à différents types de régions. Le chapitre 4 a

*. Les parties du texte écrites en italique correspondent à des travaux originaux.

analysé toutes les composantes du schéma et fourni les résultats complets de la segmentation. Nous y avons aussi montré comment établir une correspondance entre deux textures dans des images successives d'une séquence.

La seconde partie du document est consacrée au problème de la représentation de la forme. Pour l'application choisie, c'est-à-dire la compression de l'information des objets, toutes les représentations ne sont pas équivalentes tant du point de vue des performances que des facilités d'interprétation et d'intégration dans un schéma de codage complet des contours. En toute généralité, trois éléments composent un codeur de contours : un filtrage, une réduction de représentation, et une phase de codage entropique.

En développant des algorithmes de filtrage, il était devenu indispensable de trouver des algorithmes de calcul rapides. *Une étude complète de quelques idées algorithmiques, développées dans un cadre très général, a été menée au chapitre 5. Nous avons tout d'abord étendu le domaine d'application des méthodes recourant à l'histogramme local et proposé un algorithme capable de calculer l'ouvert d'une image en un seul balayage.* Pour le traitement d'images de labels, nous avons mis au point une nouvelle technique de calcul de l'ouverture par propagation de l'élément structurant. *La description des algorithmes a été complétée d'une réflexion fondamentale sur l'implémentation des opérations morphologiques élémentaires.* Elle a débouché sur quelques propriétés théoriques intéressantes, notamment concernant le fermé de l'ouvert.

En lissant les contours des régions, on espère faciliter leur représentation. Les travaux rapportés au chapitre 6 détaillent *deux familles de procédés de lissage convenant à une image de labels. Elles mettent en œuvre un même principe, aboutissant par ailleurs à des procédés auto-duaux : définir un critère géométrique local de régularité et effectuer des échanges de points entre régions en vue d'augmenter cette régularité pour toute l'image.* La première famille s'articule sur des opérateurs de la morphologie à seuillage, définis sous une nouvelle forme et dont l'annexe B reprend méthodiquement les propriétés. Le nouveau concept de polymorphisme, sorte d'extension de l'ouverture, est au cœur de la seconde famille de méthodes de filtrage. Les propriétés théoriques de cet opérateur justifient son usage tant comme mesure des effets d'un filtrage qu'au niveau du codage, et il y a fort à parier qu'on puisse l'utiliser dans d'autres applications encore.

Le chapitre 7 a exposé une représentation de forme par éléments de surface *qui inclut une modélisation de l'organisation des régions par un graphe.* Cette représentation est suffisamment générale pour servir de langage commun dans toutes les étapes d'un traitement d'image par objet et c'est là son principal intérêt. *Par le biais de critères de performance, nous avons simplifié son expression et, finalement, nous nous en sommes servi pour coder les contours.* Si les essais montrent que cette représentation n'est pas très efficace pour le codage des contours, il ne faut pas en conclure qu'elle ne présente pas d'intérêt car bon nombre de paramètres peuvent être optimisés.

Un sujet étendu comme le codage d'image orienté par la notion d'objet ne peut être bouclé en une thèse, aussi complète soit telle. Afin de s'en convaincre, il suffit de dresser la liste des outils auxquels nous avons fait appel : morphologie mathématique, extrapolation de textures, filtres de Gabor, etc. Nous avons toutefois tenu à les développer dans un cadre homogène, préservant et exploitant de la sorte les étroites liaisons entre les parties du procédé de codage. Puisse cette recherche en susciter bien d'autres !

Troisième partie

Annexes

Annexe A

Démonstration de la convergence de l'algorithme itératif général de déconvolution

Avant d'entreprendre la démonstration de la convergence, il est utile d'introduire quelques concepts topologiques.

A.1 Espaces métriques, applications et points fixes

Une *topologie* sur E est une famille $\mathcal{T}$ de parties de E comprenant l'ensemble vide $\emptyset$ et E , stable pour l'intersection finie et la réunion infinie de parties de $\mathcal{T}$. Le lecteur désireux d'approfondir ces notions est convié à lire par exemple l'ouvrage de LIPSCHUTZ [53].

Définition 72 Soit d une distance définie sur un ensemble non vide E . La topologie $\mathcal{T}$ sur E engendrée par la famille des boules ouvertes de E est appelée *topologie métrique* (ou *topologie induite par la métrique d*). De plus, l'ensemble E , ainsi que la topologie $\mathcal{T}$ induite par la distance métrique d , est appelé *espace métrique* et noté (E, d) .

Concrètement, nous travaillerons dans l'*espace euclidien* $(\mathbb{R}^N, d)$, c'est-à-dire dans l'ensemble $\mathbb{R}^N$ muni de la norme euclidienne :

$$d(p, q) = \sqrt{\sum_{n=0}^{N-1} [p(n) - q(n)]^2} \quad (\text{A.1})$$

Un *point fixe* d'une application $O : A \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ est un point $p^* \in A$ invariant pour O , ce qui signifie que $O(p^*) = p^*$. Puisque toutes les applications O ne possèdent pas de point fixe, les conditions d'existence d'un tel point sont d'importance capitale pour quantité de situations pratiques. En effet, la connaissance de l'existence d'un point fixe associé à O permet d'envisager une solution itérative à un problème. Il est également intéressant d'assurer l'unicité du point fixe, sans quoi différentes valeurs initiales ne conduisent pas toutes au même résultat. Pour obtenir l'existence d'un point fixe unique, il faut munir

O et l'espace contenant A des propriétés adéquates. La plus connue est la propriété de contraction.

Définition 73 Soit l'espace métrique $(\mathbb{R}^N, d)$. Une application $O : \mathbb{R}^N \rightarrow \mathbb{R}^N$ est dite contractante s'il existe un nombre réel α , $0 \leq \alpha < 1$ tel que pour tout $p, q \in \mathbb{R}^N$, $d(O(p), O(q)) \leq \alpha d(p, q)$.

Ainsi dans une application contractante, la distance des images de deux points quelconques est inférieure à la distance de deux points. Le concept d'application contractante est à la base du théorème du point fixe encore appelé *théorème de Banach*. Nous n'utiliserons pas ce théorème mais plutôt des versions assouplies après avoir défini une application non extensive.

Définition 74 Une application O de $\mathbb{R}^N$ dans $\mathbb{R}^N$ est non extensive si $\forall p, q \in \mathbb{R}^N : d(O(p), O(q)) \leq d(p, q)$. Une application non extensive O est stricte si l'égalité est obtenue uniquement quand $p = q$.

Contrairement à une application contractante, une application non extensive peut avoir plusieurs points fixes. Ainsi, il y a plusieurs points fixes pour l'application qui annule la première coordonnée alors qu'une application de translation non nulle n'en possède aucun. La non-extensivité stricte limite le nombre de points fixes à 1 mais il faut caractériser davantage l'image de A pour garantir l'existence d'un tel point. Les théorèmes suivants serviront à établir la convergence et l'unicité :

Théorème 75 [100] Soit une application $O : A \subseteq \mathbb{R}^N \rightarrow \mathbb{R}^N$ non extensive stricte qui applique le sous-ensemble A de l'espace métrique complet $(\mathbb{R}^N, d)$ sur A . Si l'image de A par O est un ensemble compact¹, alors O possède un point fixe $p^* \in A$ unique. Qui plus est, la séquence $p_{i+1} = O(p_i)$ converge vers p^* pour toute valeur initiale $p_0 \in A$.

Théorème 76 [100] Si une application O est non extensive et applique le sous-ensemble compact et convexe² A de l'espace métrique $(\mathbb{R}^N, d)$ sur A , alors O possède un point fixe contenu dans A . En outre, $\forall \lambda \in]0, 1[$ et $p_0 \in A$, la séquence

$$p_{i+1} = (1 - \lambda)p_i + \lambda O(p_i) \tag{A.2}$$

converge vers un point fixe de O dans A .

Contrairement au théorème 75, le théorème 76 ne garantit pas l'unicité du point fixe, comme l'illustre l'application $O : [0, 1] \rightarrow [0, 1] : p \rightarrow O(p) = p$. Ici, la valeur d'initialisation p_0 est toujours un point fixe.

Nous disposons dorénavant des outils nécessaires à la démonstration du théorème.

1. Un sous-ensemble de $\mathbb{R}^N$ est *compact* si et seulement s'il est fermé et borné.

2. Un ensemble A est *convexe* si $\forall p, q \in A$ et $\alpha \in [0, 1]$, le point $(1 - \alpha)p + \alpha q$ appartient également à A .

A.2 Thèse

Théorème 77 *Quel que soit le vecteur de données initial p_0 , l'itération $p_{i+1} = Gp_i$ converge vers une solution p^* , pas toujours unique, avec*

$$G = TDM^{-1}\Pi_2M[\uparrow 2]M^{-1}SM[\downarrow 2]M^{-1}\Pi_1M \quad (\text{A.3})$$

Les opérateurs matriciels intervenant dans cette relation sont décrits page 51 et suivantes.

A.3 Démonstration

La démonstration s'effectue en phases successives :

1. Montrons tout d'abord que *la composition d'applications de deux opérateurs non extensifs fournit une application non extensive elle aussi.*

Soient O, P deux applications non extensives. $d(PO(p), PO(q)) \leq d(O(p), O(q)) \leq d(p, q)$ par définition de la non-extensivité respective de P et de O ; dès lors PO est non extensif. De plus, si l'une d'elles est non extensive stricte, la composition est non extensive stricte.

La démarche de démonstration consiste à prouver que tous les opérateurs intervenant dans l'expression de G sont non extensifs et qu'il existe éventuellement au moins un opérateur non extensif strict.

2. *Vérification de la non-extensivité des opérateurs composant G .*

- (a) Le filtrage idéal est une opération non extensive.

Montrons en premier lieu que les opérateurs de modification des coefficients transformés sont extensifs dans le cas idéal où la réponse impulsionnelle vaut 1 dans la bande utile et 0 ailleurs, les positions des fréquences discrètes non nulles étant indicées par les éléments d'un ensemble noté E_{BP} . Pour toute paire de vecteurs transformés, éventuellement complexes, $\mathcal{P}, \mathcal{Q}$, le carré de la distance vaut

$$d^2(\mathcal{P}, \mathcal{Q}) = \sum_{k \in E_{BP}} |\mathcal{P}(k) - \mathcal{Q}(k)|^2 + \sum_{k \notin E_{BP}} |\mathcal{P}(k) - \mathcal{Q}(k)|^2 \quad (\text{A.4})$$

Grâce au partitionnement de l'ensemble des indices, la relation se ramène à une inégalité dont le second terme représente en l'occurrence le carré de la distance entre les vecteurs traités par filtrage :

$$d^2(\mathcal{P}, \mathcal{Q}) \geq \sum_{k \in E_{BP}} |\mathcal{P}(k) - \mathcal{Q}(k)|^2 = d^2(\Pi\mathcal{P}, \Pi\mathcal{Q}) \quad (\text{A.5})$$

Pour des valeurs de coefficient de filtrage Π supérieures à 1, il n'est pas possible de prouver en toute généralité que la relation précédente est vérifiée. La vérification s'effectuera au cas par cas.

Après retour dans le domaine de départ, il y a conservation du caractère non extensif. En effet,

$$d^2(\mathcal{P}, \mathcal{Q}) = \sum_{k=0}^{N-1} |\mathcal{P}(k) - \mathcal{Q}(k)|^2 \geq \sum_{k \in E_{BP}} |\mathcal{P}(k) - \mathcal{Q}(k)|^2 \quad (\text{A.6})$$

et par la version discrète du théorème de Parseval,

$$\sum_{k \in E_{BP}} |\mathcal{P}(k) - \mathcal{Q}(k)|^2 = Nd^2(M^{-1}\Pi M p, M^{-1}\Pi M q) \quad (\text{A.7})$$

En conclusion, des filtres $M^{-1}\Pi M$ et $M^{-1}SM$ non adaptatifs sont des opérateurs non extensifs.

- (b) Les opérateurs de sous-échantillonnage et de sur-échantillonnage réalisent des applications de vecteurs non extensives.

Bien choisi, le filtre $M^{-1}\Pi_1 M$ prévient le repli de spectre subséquent à un sous-échantillonnage par un facteur 2. Sur le plan théorique, on peut très bien concevoir de travailler indépendamment sur deux séquences de taille N dont tous les coefficients d'indice pair ou impair sont nuls. Le sur-échantillonnage se résumera alors à sommer les deux séquences. Puisqu'elle ne modifie en rien le signal, la paire sous-échantillonnage et sur-échantillonnage est une opération non-extensive.

- (c) L'application de la fenêtre est une opération non extensive.

Soient les éléments $p, q \in \mathbb{R}^N$,

$$d^2(p, q) = \sum_{n \in D_w} [p(n) - q(n)]^2 + \sum_{n \notin D_w} [p(n) - q(n)]^2 \quad (\text{A.8})$$

En supprimant certains termes de la somme,

$$d^2(p, q) \geq \sum_{n \notin D_w} [p(n) - q(n)]^2 = d^2(Dp, Dq) \quad (\text{A.9})$$

Il y a égalité si et seulement si $p(n) = q(n)$ pour tout indice $n \notin D_w$, ce qui n'est pas suffisant pour que l'opérateur soit non extensif strict.

- (d) L'écritage réalise une transformation non extensive.

Pour s'en convaincre, il convient d'examiner, pour chaque $j \in \{0, \dots, N-1\}$, une à une les six possibilités concernant la valeur des réels $p(j)$ et $q(j)$: $p(j) < \mathcal{G}_{min}$ ou $p(j) \in \mathcal{G}$, ou encore $p(j) > \mathcal{G}_{max}$ et de même pour $q(j)$. L'examen des six situations montre que l'écritage est une opération non extensive.

3. *Dans l'enchaînement de matrices formant G , il existe parfois un opérateur non extensif strict.*

La question de la non-extensivité stricte est délicate. Lors du filtrage et de l'application de la fenêtre, nous avons conclu que la distance entre deux vecteurs était conservée avant et après application de l'opérateur si certains éléments des vecteurs étaient égaux entre eux. La correspondance ne porte malheureusement pas sur tous les éléments des vecteurs. Il en va de même pour l'écritage.

Néanmoins, moyennant certaines conditions, le filtrage suivi d'un fenêtrage opère une transformation non extensive stricte. La position réelle du filtrage dans l'expression de G n'a pas la moindre espèce d'importance car toutes les opérations sont non extensives.

Soit F , une opération de filtrage (par exemple $F = M^{-1}SM$). Le développement emprunte un *raisonnement par l'absurde*. Supposons qu'il existe $p, q \in \mathbb{R}^N$ satisfaisant les relations $p \neq q$ et $d(DFp, DFq) = d(p, q)$. En vertu de la non-extensivité des opérateurs, il résulte que

$$d(DFp, DFq) = d(Fp, Fq) = d(p, q) \quad (\text{A.10})$$

Par analogie avec les relations A.8 et A.9, l'égalité de gauche impose l'identité de tous les éléments de Fp et Fq indicés par D_w . La relation de droite traduit à son tour l'égalité des éléments des vecteurs transformés filtrés de position $j \notin E_{BP}$.

Construisons le nouveau vecteur $z = Fp - Fq$. Seules quelques composantes $z(n)$ ne sont pas nulles ; il s'agit des composantes de position $n \in D_w^c$, le complémentaire de D_w . Elles sont au nombre de N_w^c , où N_w^c est le cardinal de l'ensemble D_w^c . Par construction, le nombre de termes non nuls de la transformée de Fourier $\mathcal{Z}$ de z ne dépasse pas le cardinal de E_{BP} . Nous utilisons ensuite le théorème suivant pour établir une condition suffisante de non-extensivité stricte.

Théorème 78 [100] *Si le vecteur $h(n)$ est tel que ses composantes sont nulles pour l'ensemble des indices extérieurs à l'intervalle $[0, M - 1]$, alors M échantillons de la transformée de Fourier pris dans $[0, 2\pi[$ spécifient de manière unique le vecteur $h(n)$.*

Ainsi, la transformée de Fourier discrète d'un vecteur à N_w^c composantes non nulles est entièrement caractérisée par un nombre correspondant de valeurs de la transformée de Fourier. Ce résultat découle des propriétés de changement de base qu'induit la transformée discrète. À supposer que nous annulions plus de coefficients de E_{BP} que N_w^c , il en résulterait que $z = 0$, en vertu de la relation biunivoque mentionnée par le théorème précédent.

Soit N_{BP} le nombre de fréquences de norme non nulle. Le cas critique étant $E_{BP} = \{0, \dots, N - 1\}$, si $N - N_{BP} \geq N_w^c$, condition équivalente à $N_{BP} \leq N_w$, alors $z = 0$. Or, $z = 0 \Rightarrow Fp = Fq$, d'où $d(Fp, Fq) = d(p, q) = 0$. La dernière égalité conduit à $p = q$, ce qui contredit l'hypothèse $p \neq q$.

Nous appellerons $N_{BP} \leq N_w$, la *condition d'unicité stricte* pour des raisons qui vont bientôt apparaître. Si cette condition n'est pas satisfaite, rien ne peut être déduit quant au caractère strict de DF .

4. Conclusions.

Toutes les hypothèses du théorème 76 sont remplies :

- G est un opérateur non extensif parce que toutes les applications impliquées dans sa construction sont non extensives.
- L'ensemble de travail $[\mathcal{G}_{min}, \mathcal{G}_{max}]^N$ est convexe et compact.

Dès lors, en vertu du théorème 76, l'algorithme converge vers une valeur fixe quelle que soit la valeur initialisant le processus itératif.

Pour assurer l'unicité de ce point, il reste à obtenir la non-extensivité *stricte* de G . La plupart des auteurs se contentent de mentionner le problème ou le passent sous silence. Cette démonstration propose le nouveau *critère d'unicité stricte*, condition suffisante qui garantit l'unicité du point fixe via le théorème 75. En pratique, la condition portera sur la matrice S placée au centre de G . Remarquons que la déconvolution sélective adapte S au contenu du signal. Pour que les conclusions restent valables, il faudra effectuer quelques boucles d'itération internes jusqu'à stabilisation avant de modifier S , tout en respectant $N_{BP} \leq N_w$.

Annexe B

Étude des propriétés des opérations élémentaires de la morphologie mathématique à seuillage binaire

Introduction aux développements théoriques

Les théoriciens de la morphologie mathématique se sont principalement intéressés à l'étude des opérations partielles pour construire des ouvertures et des fermetures, tels les filtres de rang-maximum. Cette façon de faire n'est rien de plus qu'un moyen de retrouver des propriétés perdues par les opérations partielles ; par ailleurs, ces dernières n'ont jamais été examinées de manière exhaustive. Pourtant, caractériser leur comportement est indispensable pour synthétiser des filtres qui remplissent les tâches demandées. C'est pourquoi nous nous sommes longuement penché sur la caractérisation des opérations élémentaires, en ouvrant un large éventail de pistes. Sans cette étude exhaustive, nous n'aurions pu proposer les explications intuitives du chapitre 6 qui trouvent leur fondement dans cette annexe. Les développements conduits ci-après étendent et uniformisent les travaux de MARAGOS *et al.* [58], SIMON [90] et VAN DEN BOOMGAARD [101], en mettant l'accent non pas sur le comportement statistique mais sur l'aspect géométrique.

Nous consacrons la section suivante à l'étude des propriétés des opérations partielles élémentaires. Après avoir proposé des formules de décomposition pour $X \ominus^m B$ et $X \oplus^m B$, nous examinons les propriétés d'invariance en translation et de croissance. Le comportement de ces opérations sous l'intersection et l'union γ est également analysé.

Les opérations partielles les plus simples s'apparentent à des classificateurs à poids égaux puisque l'élément structurant est plan. On peut tout aussi bien imaginer manipuler des classificateurs où les poids diffèrent pour chacun des points d'un ensemble, que ce soit l'image à traiter ou l'élément structurant. Cela débouche alors sur les notions d'opération partielle *volumique*, objet de la section B.2, et de polymorphisme. Si, pour la définition des opérations partielles, un souci premier était d'assurer une indépendance de position, certaines applications requièrent la conservation de caractéristiques topologiques. Ainsi, une petite cavité contenue dans un ensemble pourrait complètement disparaître avec des opérations partielles. Or, il se peut que cette cavité soit l'information utile. Les opérations partielles volumiques, définies dans cette annexe, offrent des artifices capables de préserver

des caractéristiques locales importantes.

B.1 Étude des propriétés élémentaires des opérations partielles

Formules de décomposition

L'interprétation usuelle de l'érosion partielle est celle-ci : un point p appartient à l'érodé partiel de X par B si $X \cap B_p$ a une aire valant au moins $\mathcal{A}(B) - m$. Donc, considérant tous les sous-ensembles B^i de B ayant une aire valant au minimum $\mathcal{A}(B) - m$, l'union des érodés morphologiques $X \ominus B^i$ fournit l'érodé partiel de seuil m . C'est-à-dire que l'érodé partiel n'est rien d'autre qu'une union d'érodés morphologiques par une famille d'ensembles extraits de B , dont le nombre dépend du seuil et de la taille de l'ensemble B initial. Qu'apporte alors le concept d'érodé partiel ? Tout d'abord, il offre un formalisme général capable de traiter une collection de sous-ensembles tirés d'un ensemble B . Et surtout, par son entremise, il est possible de construire une famille intéressante de sous-ensembles de manière purement automatique, famille éventuellement énorme, avec l'avantage que l'érodé partiel est moins sensible au bruit que l'érodé morphologique. Cette interprétation peut avoir un rôle pour le traitement d'ensembles discrétisés puisqu'alors le nombre d'érodés à calculer est fini. De tout ceci, il advient qu'une définition équivalente de l'érosion partielle est

$$X \overset{m}{\ominus} B = \bigcup X \ominus B^i \text{ tel que } B^i \subseteq B \text{ et } \mathcal{A}(B^i) \geq \mathcal{A}(B) - m \quad (\text{B.1})$$

Si B^i représente la totalité des sous-ensembles qui satisfont $B^i \subseteq B$ et $\mathcal{A}(B^i) \geq \mathcal{A}(B) - m$, l'érosion partielle se décompose comme suit

Propriété 79 (Formule de décomposition de l'érosion partielle) [58, page 1173]

$$X \overset{m}{\ominus} B = \bigcup_i X \ominus B^i = \bigcup_i \bigcap_{b \in B^i} X_{-b} \quad (\text{B.2})$$

Mais pour les érosions morphologiques, il est connu que si $B^i \subseteq B^j$ alors $X \ominus B^i \supseteq X \ominus B^j$. Dès lors, il suffit de réunir les ensembles $X \ominus B^i$ tels que $B^i \subseteq B$ et $\mathcal{A}(B^i) = \mathcal{A}(B) - m$. Le nombre d'éléments structurants à considérer, c'est-à-dire aussi le nombre d'érosions à effectuer, vaut $C_{\mathcal{A}(B)}^m$, la combinaison de m par $\mathcal{A}(B)$ ¹. Ce nombre peut être énorme, par exemple avec un élément structurant de grande taille et un seuil valant plus ou moins la moitié de son aire. Dès lors, la formule de décomposition est plus une interprétation qu'un guide pour la réalisation.

L'érosion partielle est donc une union d'érosions morphologiques. Ce résultat acquis assez intuitivement a déjà été formalisé par MATHERON [59]. Ce dernier indique que tout opérateur invariant en translation et croissant – en fait, l'érosion et la dilatation partielles respectent ces propriétés – peut s'exprimer par une union d'érosions ou par une intersection

1. La combinaison de n par N , C_N^n , est définie par $\frac{N!}{(N-n)! n!}$ où $n!$ est la factorielle de n .

de dilations. Ce résultat est aussi valable pour la dilatation partielle :

Propriété 80 (Formule de décomposition de la dilatation partielle)

$$X \overset{m}{\oplus} B = \bigcap_i X \oplus B^i = \bigcap_i \bigcup_{b \in B^i} X_b \quad (\text{B.3})$$

avec, à cette fois encore, la même collection B^i de sous-ensembles.

Ce résultat s'obtient directement ou bien en exploitant la propriété 79 et le principe de dualité (relation 6.7, page 144). En voici le détail,

$$\begin{aligned} X \overset{m}{\oplus} B &= (X^c \overset{m}{\ominus} \check{B})^c \\ &= (\bigcup_i X^c \ominus \check{B}^i)^c \\ &= (\bigcup_i [X \oplus B^i]^c)^c \\ &= \bigcap_i X \oplus B^i \end{aligned}$$

L'utilisation de ces formules permet de comprendre certains faits curieux. Prenons un ensemble en forme d'échiquier dont la taille de chaque case est de 8 pixels. La dilatation morphologique de cet ensemble par un doublet de 8 pixels de large vaut le plan tout entier. Par contre, le dilaté partiel de seuil 1 est vide. Pourquoi? Parce que, conformément à la formule de décomposition (relation B.3), la dilatation partielle se ramène dans ce cas à l'intersection de deux dilations morphologiques par seulement un point du doublet. Or, les deux résultats des dilations sont complémentaires, sans intersection. On peut donner l'équivalent en termes d'érosion partielle. Par la propriété 43 (page 144), $X \overset{m}{\oplus} B = X \overset{\mathcal{A}(B)-m-1}{\ominus} \check{B}$; la dilatation de seuil 1 par le doublet se traduit en une érosion morphologique, c'est-à-dire de seuil 0, avec l'élément symétrique. À nouveau, le résultat de cette érosion est l'ensemble vide.

Cet exemple a mis en évidence la double interprétation possible de l'érosion et de la dilatation partielles en termes d'opérations morphologiques, reprise ci-après.

Propriété 81 Soit $\check{B}^j$, la famille des sous-ensembles de $\check{B}$ ayant une aire égale à $m+1$. L'érosion et la dilatation partielles s'expriment respectivement par l'intersection et l'union suivantes :

$$X \overset{m}{\ominus} B = \bigcap_j X \ominus \check{B}^j \quad (\text{B.4})$$

$$X \overset{m}{\oplus} B = \bigcup_j X \oplus \check{B}^j \quad (\text{B.5})$$

Démonstration

Établissons la première égalité. $X \overset{m}{\ominus} B = X \overset{\mathcal{A}(B)-m-1}{\oplus} \check{B}$ par la propriété 43. En vertu de la formule de décomposition de la dilatation partielle, le second terme s'exprime par $\bigcap_j X \oplus \check{B}^j$ tel que $\mathcal{A}(\check{B}^j) \geq \mathcal{A}(B) - (\mathcal{A}(B) - m - 1) = m + 1$.

La seconde relation s'obtient par dualité. ■

L'invariance en translation, la croissance des opérations partielles et propriétés associées

La translation temporelle d'un signal introduit un facteur de phase dans la transformée de Fourier, facteur qu'il est possible d'extraire et de neutraliser. Dans le même état d'esprit, les opérations morphologiques jouissent de la propriété d'invariance en translation. Il est clair qu'ainsi l'origine ne joue plus aucun rôle privilégié et que les transformations sont insensibles à la position absolue des points de l'espace. La proposition suivante décrit le comportement des opérations partielles sous la translation.

Comme elles sont des conséquences immédiates des formules de décomposition, nous fournissons les propriétés suivantes sans démonstration. On pourrait aussi les vérifier directement d'après la définition. Signalons aussi que la plupart de ces propriétés ont été énoncées précédemment par SIMON [90].

Propriété 82 (Invariance en translation)

$$X \overset{m}{\ominus} B_q = (X \overset{m}{\ominus} B)_{-q} \quad (\text{B.6})$$

$$X_q \overset{m}{\ominus} B = (X \overset{m}{\ominus} B)_q \quad (\text{B.7})$$

$$X \overset{m}{\oplus} B_q = (X \overset{m}{\oplus} B)_q \quad (\text{B.8})$$

$$X_q \overset{m}{\oplus} B = (X \overset{m}{\oplus} B)_q \quad (\text{B.9})$$

L'invariance en translation est fondamentale pour toutes les opérations de traitement du signal car elle garantit que le résultat ne dépend pas de la position absolue des points ni de l'origine. Tout aussi importante est la croissance des opérateurs; il en résulte un concept d'ordre entre les ensembles traités sous l'opérateur qui corrobore la notion simple d'inclusion entre ensembles. En d'autres termes, la structure de treillis est préservée.

Propriété 83 (Croissance des opérateurs partiels)

$$X \subseteq Y \Rightarrow X \overset{m}{\ominus} B \subseteq Y \overset{m}{\ominus} B \quad (\text{B.10})$$

$$X \subseteq Y \Rightarrow X \overset{m}{\oplus} B \subseteq Y \overset{m}{\oplus} B \quad (\text{B.11})$$

Propriété 84

$$B \subseteq C \Rightarrow X \overset{m}{\ominus} B \supseteq X \overset{m}{\ominus} C \quad (\text{B.12})$$

$$B \subseteq C \Rightarrow X \overset{m}{\oplus} B \supseteq X \overset{m}{\oplus} C \quad (\text{B.13})$$

Propriété 85 (Commutativité de la dilatation partielle)

$$X \overset{m}{\oplus} B = B \overset{m}{\oplus} X \quad (\text{B.14})$$

Propriété 86

$$m \leq n \Rightarrow X \overset{m}{\ominus} B \subseteq X \overset{n}{\ominus} B \quad (\text{B.15})$$

$$m \leq n \Rightarrow X \overset{m}{\oplus} B \supseteq X \overset{n}{\oplus} B \quad (\text{B.16})$$

En particulier, $X \ominus B \subseteq X \overset{m}{\ominus} B$ et $X \oplus B \supseteq X \overset{m}{\oplus} B$.

Développements en série. Dans ce paragraphe, nous allons montrer que les formules de décomposition permettent d'obtenir des formules de développement en série pour lesquelles une opération avec un grand élément structurant se ramène à plusieurs opérations manipulant des éléments structurants plus petits. Il va de soi que ces relations sont particulièrement utiles pour toute réalisation, tant matérielle que par logiciel. Voici, en substance, ces formules :

Propriété 87 (Formules de développement en série)

Soient $k_0 = \max(0, m - \mathcal{A}(C \setminus B))$ et $k_1 = \min(\mathcal{A}(B), m)$

$$X \overset{m}{\ominus} (B \cup C) = \bigcup_{k=k_0, \dots, k_1} [(X \overset{k}{\ominus} B) \cap (X \overset{m-k}{\ominus} (C \setminus B))] \quad (\text{B.17})$$

$$X \overset{m}{\oplus} (B \cup C) = \bigcap_{k=k_0, \dots, k_1} [(X \overset{k}{\oplus} B) \cup (X \overset{m-k}{\oplus} (C \setminus B))] \quad (\text{B.18})$$

Démonstration

Relation B.17 :

Par la propriété 79, $X \overset{m}{\ominus} (B \cup C) = \bigcup_i X \ominus [B \cup C]^i$. De plus, comme $X \ominus (D \cup E) = (X \ominus D) \cap (X \ominus E)$ (cf. [33, page 539]), $X \ominus [B \cup C]^i = (X \ominus B^i) \cap (X \ominus [C \setminus B]^i)$. C'est donc l'union de toutes les combinaisons pour lesquelles l'imprécision totale sur l'érosion morphologique vaut m , une partie de m étant associée à B^i , l'autre à l'ensemble $[C \setminus B]^i$: $X \overset{m}{\ominus} (B \cup C) = \bigcup_i [(X \ominus B^i) \cap (X \ominus [C \setminus B]^i)]$. C'est pourquoi, à supposer que $m \leq \mathcal{A}(B)$ et $m \leq \mathcal{A}(C \setminus B)$, le regroupement de certains termes mène à $X \overset{m}{\ominus} (B \cup C) = \bigcup_{k=0, \dots, m} [(X \overset{k}{\ominus} B) \cap (X \overset{m-k}{\ominus} (C \setminus B))]$.

Les contraintes sur k sont destinées à garantir que les définitions des érosions partielles intervenant dans les différentes composantes de la formule de décomposition ont du sens. Elles imposent simplement que $0 \leq k \leq \mathcal{A}(B)$ et $0 \leq m - k \leq \mathcal{A}(C \setminus B)$.

Relation B.18 :

Un raisonnement de même nature conduit à la deuxième relation. Elle s'obtient également en exploitant le principe de dualité. ■

Exemple 1 :

Traisons une fois encore le cas du doublet $B = \{0, p\}$ appliqué à un échiquier X dont les cases sont écartées d'une distance égale à la longueur du doublet. La dilatation partielle de seuil 1 de X par le doublet se décompose via la relation précédente en : $X \overset{1}{\oplus} B = [(X \overset{0}{\oplus} \{0\}) \cup (X \overset{1}{\oplus} \{p\})] \cap [(X \overset{1}{\oplus} \{0\}) \cup (X \overset{0}{\oplus} \{p\})]$. Par calcul, l'expression devient $X \overset{1}{\oplus} B = [X_0 \cup \emptyset] \cap [\emptyset \cup X_p] = X_0 \cap X_p$. Mais cet ensemble est vide par définition de l'ensemble X .

Exemple 2 :

Un exemple plus abstrait permet de comprendre l'utilité des formules de développement. Soit à calculer l'érosion partielle de X par B pour un seuil m . Retirons de B l'élément noté p . De la sorte, nous avons deux ensembles disjoints notés $B_1 (= B \setminus \{p\})$ et $\{p\}$. Par développement en série de l'érosion partielle :

$$\begin{aligned} X \overset{m}{\ominus} B &= [(X \overset{0}{\ominus} \{p\}) \cap (X \overset{m}{\ominus} B_1)] \cup [(X \overset{1}{\ominus} \{p\}) \cap (X \overset{m-1}{\ominus} B_1)] \\ &= [X_{-p} \cap (X \overset{m}{\ominus} B_1)] \cup (X \overset{m-1}{\ominus} B_1) \end{aligned}$$

Une opération avec un élément structurant de grande taille se ramène à deux opérations qui utilisent un élément structurant plus petit d'un point. Mais comme cette formule est récurrente, on peut poursuivre le raisonnement et se ramener à un élément structurant de petite taille, à condition que le seuil ne dépasse pas les limites fixées par la définition. Bien sûr, ce n'est sans doute pas la voie à suivre dans une approche par logiciel, mais pour d'autres applications, cette formule fournit un moyen de calculer malgré tout des érosions pour des éléments structurants dépassant la taille autorisée.

Un corollaire reprend des formules un peu plus générales :

Corollaire 88

$$X \overset{m}{\ominus} (B \cup C) \supseteq \bigcup_{k=k_0, \dots, k_1} [(X \overset{k}{\ominus} B) \cap (X \overset{m-k}{\ominus} C)] \quad (\text{B.19})$$

$$X \overset{m}{\oplus} (B \cup C) \subseteq \bigcap_{k=k_0, \dots, k_1} [(X \overset{k}{\oplus} B) \cup (X \overset{m-k}{\oplus} C)] \quad (\text{B.20})$$

L'égalité n'est garantie que si B et C sont mutuellement exclusifs.

Démonstration

Évident car, en vertu des relations B.12 et B.13, il est établi que $X \overset{m-k}{\ominus} (C \setminus B) \supseteq X \overset{m-k}{\ominus} C$ et $X \overset{m-k}{\oplus} (C \setminus B) \subseteq X \overset{m-k}{\oplus} C$. Ce qui, combiné avec la propriété précédente, démontre les deux relations du corollaire. ■

Comportement sous l'intersection et l'union

D'un point de vue purement théorique, l'érosion et la dilatation morphologiques se caractérisent par un comportement particulier face à l'intersection et à l'union. En effet, l'érosion distribue l'intersection et la dilatation distribue l'union. Assez naturellement et bien que ce soit regrettable, ces propriétés disparaissent pour les opérations partielles correspondantes parce que ces dernières contraignent moins le comportement. La série de propriétés à venir détaille les relations d'inclusion qui existent pour les opérations partielles.

Propriété 89

$$(X \overset{m}{\ominus} B) \cup (X \overset{m}{\ominus} C) \subseteq X \overset{m}{\ominus} (B \cap C) \quad (\text{B.21})$$

$$(X \overset{m}{\oplus} B) \cap (X \overset{m}{\oplus} C) \supseteq X \overset{m}{\oplus} (B \cup C) \quad (\text{B.22})$$

$$(X \overset{m}{\oplus} B) \cap (X \overset{m}{\oplus} C) \supseteq X \overset{m}{\oplus} (B \cap C) \quad (\text{B.23})$$

$$(X \overset{m}{\ominus} B) \cup (X \overset{m}{\ominus} C) \subseteq X \overset{m}{\ominus} (B \cup C) \quad (\text{B.24})$$

Démonstration

Toutes ces relations sont des conséquences immédiates des relations B.12 et B.13. Démontrons par exemple la relation B.23.

Puisque $B \cap C \subseteq B$ et $B \cap C \subseteq C$, par la relation B.13, $X \overset{m}{\oplus} (B \cap C) \subseteq X \overset{m}{\oplus} B$ et $X \overset{m}{\oplus} (B \cap C) \subseteq X \overset{m}{\oplus} C$. ■

Les relations suivantes examinent l'effet d'une somme de seuils ou, à l'inverse, lorsqu'il s'agit de séparer un seuil en une somme de seuils. Partout nous supposons que les seuils prennent des valeurs permises par la définition des opérations partielles.

Propriété 90

$$X \overset{m+n}{\ominus} B \supseteq (X \overset{m}{\ominus} B) \cup (X \overset{n}{\ominus} B) \quad (\text{B.25})$$

$$X \overset{m+n}{\oplus} B \subseteq (X \overset{m}{\oplus} B) \cap (X \overset{n}{\oplus} B) \quad (\text{B.26})$$

Démonstration

Relation B.25:

Par la relation B.15, $X \overset{m+n}{\ominus} B \supseteq X \overset{m}{\ominus} B$. De même, $X \overset{m+n}{\ominus} B \supseteq X \overset{n}{\ominus} B$. D'où, in fine, $X \overset{m+n}{\ominus} B \supseteq (X \overset{m}{\ominus} B) \cup (X \overset{n}{\ominus} B)$.

Relation B.26:

On tient un raisonnement similaire pour la relation B.26. ■

Propriété 91

$$X \overset{m+n}{\ominus} (B \cup C) \supseteq (X \overset{m}{\ominus} B) \cap (X \overset{n}{\ominus} C) \quad (\text{B.27})$$

$$X \overset{m+n}{\oplus} (B \cup C) \subseteq (X \overset{m}{\oplus} B) \cup (X \overset{n}{\oplus} C) \quad (\text{B.28})$$

Démonstration

Relation B.27:

Soit un point p appartenant à $(X \overset{m}{\ominus} B) \cap (X \overset{n}{\ominus} C)$. p est donc contenu dans les deux ensembles, ce qui signifie que $\mathcal{A}(X \cap B_p) \geq \mathcal{A}(B) - m$ et $\mathcal{A}(X \cap C_p) \geq \mathcal{A}(C) - n$. La somme de ces deux relations produit l'inégalité $\mathcal{A}(X \cap B_p) + \mathcal{A}(X \cap C_p) \geq \mathcal{A}(B) + \mathcal{A}(C) - m - n$. D'autre part, si $q \in X \overset{m+n}{\ominus} (B \cup C)$, cela équivaut à dire que $\mathcal{A}(X \cap (B \cup C)_q) \geq \mathcal{A}(B \cup C) - m - n$. En décomposant le premier terme de l'inégalité :

$$\begin{aligned} \mathcal{A}((X \cap B_q) \cup (X \cap C_q)) &\geq \mathcal{A}(B \cup C) - m - n \\ \mathcal{A}(X \cap B_q) + \mathcal{A}(X \cap C_q) - \mathcal{A}(X \cap B_q \cap C_q) &\geq \mathcal{A}(B) + \mathcal{A}(C) - \mathcal{A}(B \cap C) - m - n \\ \mathcal{A}(X \cap B_q) + \mathcal{A}(X \cap C_q) &\geq \mathcal{A}(B) + \mathcal{A}(C) - [\mathcal{A}(B \cap C) - \mathcal{A}(X \cap B_q \cap C_q)] - m - n \end{aligned}$$

Or la différence $[\mathcal{A}(B \cap C) - \mathcal{A}(X \cap B_q \cap C_q)]$ est toujours positive. D'où si $p \in (X \overset{m}{\ominus} B) \cap (X \overset{n}{\ominus} C)$ alors $p \in X \overset{m+n}{\ominus} (B \cup C)$.

Relation B.28:

Partant de la relation précédente, $[X \overset{m+n}{\ominus} (B \cup C)]^c \subseteq [(X \overset{m}{\ominus} B) \cap (X \overset{n}{\ominus} C)]^c$. Puis au moyen du principe de dualité (relation 6.7), l'expression de l'inclusion devient successivement :

$$\begin{aligned} X^c \overset{m+n}{\oplus} (\check{B} \cup \check{C}) &\subseteq (X \overset{m}{\ominus} B)^c \cup (X \overset{n}{\ominus} C)^c \\ X^c \overset{m+n}{\oplus} (\check{B} \cup \check{C}) &\subseteq (X^c \overset{m}{\oplus} \check{B}) \cup (X^c \overset{n}{\oplus} \check{C}) \end{aligned}$$

Si nous remplaçons $X^c, \check{B}$ et $\check{C}$ respectivement par Y, D, E , la relation s'écrit $Y \overset{m+n}{\oplus} (D \cup E) \subseteq (Y \overset{m}{\oplus} D) \cup (Y \overset{n}{\oplus} E)$, c'est-à-dire la relation à démontrer. ■

Propriété 92

$$(X \cap Y) \overset{m+n}{\ominus} B \supseteq (X \overset{m}{\ominus} B) \cap (Y \overset{n}{\ominus} B) \quad (\text{B.29})$$

$$(X \cup Y) \overset{m+n}{\oplus} B \subseteq (X \overset{m}{\oplus} B) \cup (Y \overset{n}{\oplus} B) \quad (\text{B.30})$$

Démonstration

La relation B.30 est une conséquence de la commutativité de la dilatation partielle et de la relation B.28. Pour arriver à la première inclusion, il suffit de prendre le complémentaire de B.30 et de faire jouer la dualité : $[(X \cup Y) \overset{m+n}{\oplus} B]^c \supseteq [(X \overset{m}{\oplus} B) \cup (Y \overset{n}{\oplus} B)]^c \Leftrightarrow (X^c \cap Y^c) \overset{m+n}{\ominus} \check{B} \supseteq (X^c \overset{m}{\ominus} \check{B}) \cap (Y^c \overset{n}{\ominus} \check{B})$, ce qui n'est somme toute qu'une réécriture de la relation B.29. ■

Détaillons à présent les propriétés où n'intervient qu'une seule valeur de seuil. Parmi toutes les propriétés, pas une seule n'est malheureusement une égalité. Les opérations partielles ne mènent qu'à des inclusions. L'interprétation de l'érosion et de la dilatation en termes d'union d'érosions ou d'intersection de dilatations, que nous avons donnée précédemment, suffit à comprendre pourquoi le comportement des opérations partielles est moins régulier que celui des opérations morphologiques correspondantes.

Propriété 93

$$(X \overset{m}{\ominus} B) \cup (Y \overset{m}{\ominus} B) \subseteq (X \cup Y) \overset{m}{\ominus} B \quad (\text{B.31})$$

$$(X \cap Y) \overset{m}{\ominus} B \subseteq (X \overset{m}{\ominus} B) \cap (Y \overset{m}{\ominus} B) \quad (\text{B.32})$$

$$(X \overset{m}{\oplus} B) \cup (Y \overset{m}{\oplus} B) \subseteq (X \cup Y) \overset{m}{\oplus} B \quad (\text{B.33})$$

$$(X \cap Y) \overset{m}{\oplus} B \subseteq (X \overset{m}{\oplus} B) \cap (Y \overset{m}{\oplus} B) \quad (\text{B.34})$$

Démonstration

Relation B.31:

Par la croissance de l'opération d'érosion partielle, $X \overset{m}{\ominus} B \subseteq (X \cup Y) \overset{m}{\ominus} B$. De même, $Y \overset{m}{\ominus} B \subseteq (X \cup Y) \overset{m}{\ominus} B$. Donc l'union de $X \overset{m}{\ominus} B$ et de $Y \overset{m}{\ominus} B$ est comprise dans $(X \cup Y) \overset{m}{\ominus} B$.

La démonstration des autres inclusions s'articule sur le même argument. ■

Propriété 94

$$X \overset{m}{\ominus} (B \cup C) \subseteq (X \overset{m}{\ominus} B) \cap (X \overset{m}{\ominus} C) \quad (\text{B.35})$$

$$X \overset{m}{\oplus} (B \cup C) \supseteq (X \overset{m}{\oplus} B) \cup (X \overset{m}{\oplus} C) \quad (\text{B.36})$$

Démonstration

La démonstration de la première relation s'appuie sur la relation B.12. En effet, comme $B \cup C \supseteq B$, $X \overset{m}{\ominus} (B \cup C) \subseteq X \overset{m}{\ominus} B$. De même, $X \overset{m}{\ominus} (B \cup C) \subseteq X \overset{m}{\ominus} C$. Les éléments de $X \overset{m}{\ominus} (B \cup C)$ appartiennent donc à l'intersection des deux ensembles. La seconde relation s'obtient par dualité. ■

Mise en cascade d'opérations élémentaires. La propriété de mise en cascade de deux érosions morphologiques, à savoir $(X \ominus B) \ominus C = X \ominus (B \oplus C)$, est un pilier de la théorie et de la pratique. C'est grâce à elle que s'établit la notion de taille morphologique, et par suite, celle de filtres axés sur la notion de taille. Comme il apparaît dans les propriétés suivantes, il n'y a pas de relation de chaîne qui tienne pour des opérations partielles. Autrement dit, il n'est plus possible de définir une "taille partielle". En conséquence aussi, une opération avec un élément structurant qui est le dilaté d'un ensemble n'équivaut pas à deux opérations avec chacun des ensembles constituants.

Propriété 95

$$X \overset{m}{\ominus} (B \oplus C) \subseteq (X \overset{m}{\ominus} B) \ominus C \quad (\text{B.37})$$

$$X \ominus (B \overset{m}{\oplus} C) \supseteq (X \ominus B) \overset{m}{\ominus} C \quad (\text{B.38})$$

$$X \overset{m}{\oplus} (B \oplus C) \supseteq (X \overset{m}{\oplus} B) \oplus C \quad (\text{B.39})$$

$$X \oplus (B \overset{m}{\oplus} C) \subseteq (X \oplus B) \overset{m}{\oplus} C \quad (\text{B.40})$$

Démonstration

Relation B.37:

Par définition de la dilatation morphologique en termes d'union, $X \overset{m}{\ominus} (B \oplus C) = X \overset{m}{\ominus} (\bigcup_{c \in C} B_c)$. Nous utilisons ensuite la relation B.35 pour résoudre la question d'une érosion partielle par une union d'éléments structurants. Il en résulte que $X \overset{m}{\ominus} (\bigcup_{c \in C} B_c) \subseteq \bigcap_{c \in C} (X \overset{m}{\ominus} B_c)$. Mais en appliquant une propriété de translation démontrée précédemment et la définition de l'érosion morphologique au second terme de l'inclusion : $\bigcap_{c \in C} (X \overset{m}{\ominus} B_c) = \bigcap_{c \in C} (X \overset{m}{\ominus} B)_{-c}$ encore égal à $(X \overset{m}{\ominus} B) \ominus C$. De fil en aiguille, nous établissons ainsi que $X \overset{m}{\ominus} (B \oplus C) \subseteq (X \overset{m}{\ominus} B) \ominus C$.

Relation B.38:

Par la formule de décomposition de la dilatation partielle (cf. page 207), $X \ominus (B \overset{m}{\oplus} C) = X \ominus (\bigcap_i (\bigcup_{c \in C^i} B_c))$. Or, $X \ominus (B \cap C) \supseteq (X \ominus B) \cup (X \ominus C)$ et donc $X \ominus (\bigcap_i (\bigcup_{c \in C^i} B_c)) \supseteq \bigcup_i (X \ominus \bigcup_{c \in C^i} B_c)$. Mais comme $X \ominus (B \cup C) = (X \ominus B) \cap (X \ominus C)$, on a aussi $\bigcup_i (X \ominus \bigcup_{c \in C^i} B_c) = \bigcup_i (\bigcap_{c \in C^i} X \ominus B_c)$. Une propriété de translation de l'érosion partielle montre que ce dernier terme est égal à $\bigcup_i (\bigcap_{c \in C^i} (X \ominus B)_{-c})$, cette dernière expression étant précisément la décomposition de l'érosion partielle de $X \ominus B$ par C . En conclusion : $X \ominus (B \overset{m}{\oplus} C) \supseteq (X \ominus B) \overset{m}{\ominus} C$.

Relation B.39:

Par dualité de B.37 et des opérations d'érosion et de dilatation partielles :

$$\begin{aligned} [X \overset{m}{\ominus} (B \oplus C)]^c &\supseteq [(X \overset{m}{\ominus} B) \ominus C]^c \\ X^c \overset{m}{\oplus} (B \overset{m}{\oplus} C) &\supseteq (X \overset{m}{\ominus} B)^c \overset{m}{\oplus} \check{C} \\ X^c \overset{m}{\oplus} (\check{B} \oplus \check{C}) &\supseteq (X^c \overset{m}{\oplus} \check{B}) \oplus \check{C} \end{aligned}$$

C'est bien la propriété à démontrer aux notations près.

Relation B.40:

C'est à la fois une autre écriture de B.39, puisque la dilatation partielle est commutative, et le dual de B.38. ■

Propriété 96

$$(X \ominus B) \overset{m}{\oplus} C \subseteq (X \overset{m}{\oplus} C) \ominus B \quad (\text{B.41})$$

$$(X \oplus B) \overset{m}{\ominus} C \supseteq (X \overset{m}{\ominus} C) \oplus B \quad (\text{B.42})$$

Démonstration

Démontrons d'abord la première relation.

$(X \ominus B) \overset{m}{\oplus} C = (\cap_b X_{-b}) \overset{m}{\oplus} C$. Or $\forall b^*, \cap_b X_{-b} \subseteq X_{-b^*}$. Par invariance en translation et croissance de la dilatation partielle (cf. respectivement les relations B.9 et B.11), $(\cap_b X_{-b}) \overset{m}{\oplus} C \subseteq X_{-b^*} \overset{m}{\oplus} C = (X \overset{m}{\oplus} C)_{-b^*}$. Cette dernière relation étant valable pour tout $b^* \in B$, $(X \ominus B) \overset{m}{\oplus} C \subseteq \cap_{b^* \in B} (X \overset{m}{\oplus} C)_{-b^*} = (X \overset{m}{\oplus} C) \ominus B$.

La seconde relation est simplement la relation duale de la première. ■

Nous avons cherché à obtenir des règles de mise en cascade pour éventuellement simplifier des expressions complexes. Jusqu'à présent, toutes les règles sont des inclusions. Quid de $X \overset{m}{\ominus} (B \overset{m}{\oplus} C)$ et de $(X \overset{m}{\ominus} B) \overset{m}{\ominus} C$?

Corollaire 97 *Il n'existe malheureusement pas de relation de mise en cascade pour les opérations partielles comme il en existe pour les opérations d'érosion et de dilatation morphologiques. Il existe néanmoins un ensemble qui contiennent à la fois $X \overset{m}{\ominus} (B \overset{m}{\oplus} C)$ et $(X \overset{m}{\ominus} B) \overset{m}{\ominus} C$:*

$$X \overset{m}{\ominus} (B \overset{m}{\oplus} C) \supseteq (X \ominus B) \overset{m}{\ominus} C \subseteq (X \overset{m}{\ominus} B) \overset{m}{\ominus} C \quad (\text{B.43})$$

De même, par dualité,

$$X \overset{m}{\oplus} (B \overset{m}{\ominus} C) \subseteq (X \oplus B) \overset{m}{\oplus} C \supseteq (X \overset{m}{\oplus} B) \overset{m}{\oplus} C \quad (\text{B.44})$$

Démonstration

Inclusion de gauche :

Comme l'érosion partielle contient l'érosion morphologique et suivant la relation B.38 : $X \overset{m}{\ominus} (B \overset{m}{\oplus} C) \supseteq X \ominus (B \overset{m}{\oplus} C) \supseteq (X \ominus B) \overset{m}{\ominus} C$.

Inclusion de droite :

$X \ominus B \subseteq X \overset{m}{\ominus} B$ et, puisque l'érosion partielle est croissante, $(X \ominus B) \overset{m}{\ominus} C \subseteq (X \overset{m}{\ominus} B) \overset{m}{\ominus} C$. ■

Extensivité. Qu'en est-il de l'extensivité des opérations partielles élémentaires? L'anti-extensivité de l'érosion morphologique est garantie à condition que l'élément structurant contienne l'origine. Pour l'érosion partielle, ce n'est même pas le cas; elle n'est pas anti-extensive. Un contre-exemple illustre ces derniers propos. Prenons un élément structurant B carré de quatre pixels avec l'origine placée sur le coin supérieur gauche. Soit à traiter un ensemble X égal à B sauf que l'origine n'appartient plus à X . Le résultat de l'érosion partielle de seuil 1 de X par B est le singleton constitué de l'origine ce qui signifie que, malgré l'appartenance $0 \in B$, $X \overset{1}{\ominus} B \not\subseteq X$. Comment expliquer cela? Reprenons la formule B.1 qui transforme l'érosion partielle en une union d'érosions morphologiques: la famille des B^i contient tous les sous-ensembles de B amputés d'un pixel. Dans le cas présent,

c'est précisément l'élément structurant identique à X –c'est-à-dire le carré sans l'origine– qui contribue seul à l'union. En conclusion, si l'érosion partielle n'est pas anti-extensive même si l'origine appartient à l'élément structurant, c'est parce qu'elle est équivalente à une union d'érosions morphologiques par des éléments structurants qui ne contiennent pas nécessairement l'origine. Mais dès lors que tous les B^i de contribution non vide dans l'union contiennent l'origine, l'érosion partielle est anti-extensive.

Mise en cascade d'opérations avec des seuils différents. Dans une expression contenant plusieurs opérations partielles en cascade, il n'est pas possible de permuter les seuils. Ainsi par exemple : $(X \overset{m}{\ominus} B) \overset{n}{\ominus} B \neq (X \overset{n}{\ominus} B) \overset{m}{\ominus} B$, comme le montre la figure B.1. Pareillement, $(X \overset{m}{\oplus} B) \overset{n}{\oplus} B \neq (X \overset{n}{\oplus} B) \overset{m}{\oplus} B$. Signalons le résultat intéressant de SALEMBIER [84] qui montre que, si l'élément structurant est unidimensionnel, $(X \overset{m}{\oplus} B) \overset{m}{\ominus} B \subseteq (X \overset{n}{\oplus} B) \overset{n}{\ominus} B$ moyennant $m \geq n$.

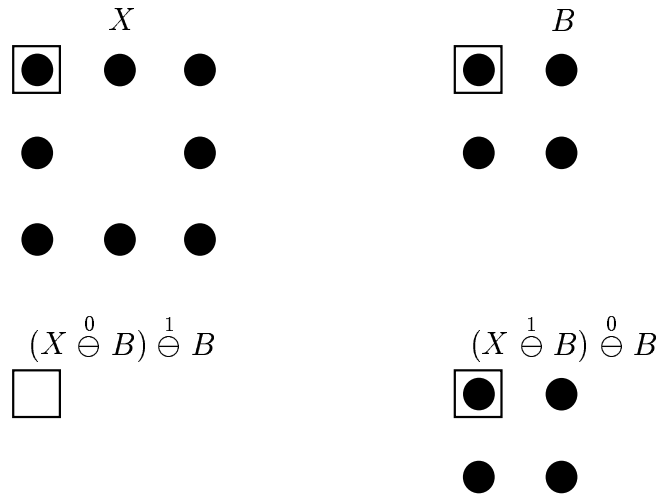


FIG. B.1 - Pour illustrer que $(X \overset{m}{\ominus} B) \overset{n}{\ominus} B \neq (X \overset{n}{\ominus} B) \overset{m}{\ominus} B$.

La mise en cascade d'opérations partielles avec des seuils identiques produit parfois des phénomènes curieux comme celui d'oscillation illustré à la figure B.2. Pour un carré de 2x2 pixels, le résultat de la dilatation de seuil 1 itérée bascule entre une croix et la carré translaté. Avec un seuil égal à 2, le résultat de $(X \overset{2}{\oplus} B) \overset{2}{\oplus} B$ est vide. Dans de telles conditions, il n'est malheureusement pas possible de définir le concept de taille morphologique partielle.

B.2 Ajout d'une mesure : la notion de volume

Telles qu'introduites précédemment, les opérations souffrent l'inconvénient –c'est parfois aussi un avantage– de ne pas considérer la position de la simplification introduite avec le seuil. Disons plutôt que, si le bât blesse, c'est surtout à cause de l'incapacité à gérer des

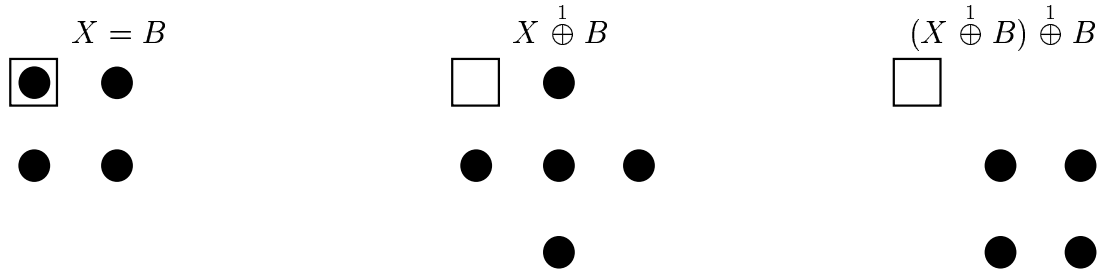


FIG. B.2 - Pourquoi le concept de ‘taille morphologique partielle’ n’est pas possible.

notions de géométrie dans les résultats des opérations partielles. Dans certaines applications, un trou à l’intérieur n’a pas la même signification qu’un pixel manquant sur le bord. Le contraire est plutôt rare. La question que suggère cette digression tient à : comment introduire une géométrie de l’erreur à partir de l’élément structurant ? C’est pour résoudre cette difficulté que nous ajoutons une mesure, appelée *volume*, à chaque ensemble. La démarche suivie consiste à définir la notion de volume d’un ensemble et à l’intégrer dans la définition des opérations partielles pour accroître les moyens de caractériser X .

Définition 98 Soit $v(x)$, un nombre réel associé à chaque élément x d’un ensemble X . Le volume de cet ensemble noté $\mathcal{V}(X)$ est le réel obtenu par sommation du poids de ses éléments :

$$\mathcal{V}(X) = \sum_{x \in X} v(x) \quad (\text{B.45})$$

À présent, l’attribution des poids est laissée à la libre appréciation du praticien. Par un mécanisme judicieux et propre à son application, il choisit un poids pour chaque élément d’un ensemble, comme par exemple ceux de la fonction distance. Le volume peut être introduit dans chacune des opérations partielles précédentes moyennant certaines modifications. Auparavant, il faut définir le volume d’une intersection d’ensembles, ou ce qui revient au même, de l’union d’ensembles. Mais la tâche est ardue. En effet, de nombreuses propriétés démontrées pour des définitions avec la notion d’aire utilisent l’importante relation que voici : $\mathcal{A}(X \cup Y) = \mathcal{A}(X) + \mathcal{A}(Y) - \mathcal{A}(X \cap Y)$. Il existe donc une correspondance bi-univoque entre la relation purement ensembliste et l’aire des ensembles. Pour le volume, ce n’est pas si simple. Quoi qu’il en soit, définir l’intersection va forcément de pair avec une définition pour l’union. En effet, imposant que la relation précédente soit également respectée, l’union sera définie via la notion d’intersection. Pour le concept de volume d’une intersection de deux ensembles, nous utiliserons :

Définition 99 Soient deux ensembles X, Y et v, w , les jeux de poids respectifs. Le volume de l’intersection de ces deux ensembles vaut

$$\mathcal{V}(X \cap Y) = \sum_{p \in X \cap Y} v(p)w(p) \quad (\text{B.46})$$

La difficulté majeure de cette définition consistera à comparer le volume d’une intersection d’ensembles avec le volume d’un de ces deux ensembles à cause du biais qu’introduit l’autre volume.

Définition de l'érosion partielle volumique

Dans la morphologie mathématique, l'ensemble X et l'élément structurant B ne jouent pas le même rôle. Nous cherchons à tirer le maximum d'information sur X en utilisant B comme forme d'analyse. B est donc l'outil et X l'objet à caractériser. Nous pouvons alors introduire une différence fondamentale entre X et B dans les définitions mêmes des opérations partielles volumiques. Elle est résumée par l'hypothèse suivante : *l'ensemble X à caractériser a un jeu de poids valant 1 à l'intérieur et 0 à l'extérieur*. Cette dissymétrie est naturelle parce que l'élément structurant est une caractéristique connue a priori et donc accessible à l'utilisateur ; il peut maîtriser son usage. Mais au niveau du principe, l'inverse est possible.

Après cette précision importante, nous définissons l'*érosion partielle volumique*.

Définition 100 *L'érosion partielle volumique de seuil α d'un ensemble X est définie comme l'ensemble des points p pour lesquels le volume de l'intersection de X par B_p est supérieur à une fraction $(1 - \alpha)$ du volume de B :*

$$X \stackrel{\alpha}{\ominus} B = \{p \in \mathbb{Z}^2 \mid \mathcal{V}(X \cap B_p) \geq (1 - \alpha)\mathcal{V}(B)\} \quad (\text{B.47})$$

avec $\alpha \in [0, 1]$.

Définition de la dilatation partielle volumique

La dilatation partielle volumique se définit par analogie avec l'érosion partielle volumique.

Définition 101 *La dilatation partielle volumique de seuil α d'un ensemble X est définie comme l'ensemble des points p pour lesquels le volume de l'intersection de X par $\check{B}_p$ est strictement supérieur à une fraction α du volume de B , c'est-à-dire*

$$X \stackrel{\alpha}{\oplus} B = \{p \in \mathbb{Z}^2 \mid \mathcal{V}(X \cap \check{B}_p) > \alpha\mathcal{V}(B)\} \quad (\text{B.48})$$

avec $\alpha \in [0, 1]$.

Propriété 102 *Pour un élément structurant dont tous les poids valent 1,*

$$X \stackrel{\alpha}{\ominus} B = X \stackrel{\alpha}{\oplus} B \quad (\text{B.49})$$

où $X \stackrel{\alpha}{\ominus} B = \{p \in \mathbb{Z}^2 \mid \mathcal{A}(X \cap B_p) \geq (1 - \alpha)\mathcal{A}(B)\}$.

Démonstration

Évident parce que si tous les poids de B valent 1, la notion d'aire est équivalente à celle de volume. ■

Pseudo-ouverture et pseudo-fermeture volumiques

Définition 103 La pseudo-ouverture volumique et la pseudo-fermeture volumique de seuils (α, β) d'un ensemble X par l'élément structurant B sont définies respectivement par

$$X \overset{\alpha, \beta}{\underset{\circ}{\ominus}} B = (X \overset{\alpha}{\underset{\circ}{\ominus}} B) \overset{\beta}{\underset{\oplus}{\oplus}} B \quad (\text{B.50})$$

et

$$X \overset{\alpha, \beta}{\underset{\bullet}{\oplus}} B = (X \overset{\alpha}{\underset{\bullet}{\oplus}} B) \overset{\beta}{\underset{\ominus}{\ominus}} B \quad (\text{B.51})$$

Si α est choisi égal à β , les opérations seront notées par $X \overset{\alpha}{\underset{\circ}{\ominus}} B$ et $X \overset{\alpha}{\underset{\bullet}{\oplus}} B$.

La figure B.3 met en œuvre une opération partielle volumique. Supposons que nous désirions effectuer la pseudo-ouverture par un carré 3x3 avec un seuil de 1 à l'érosion mais telle que le coin inférieur gauche ne soit jamais absent. Par l'entremise des opérations volumiques, nous sommes capable de réaliser cette transformation. La solution consiste à attribuer un poids supérieur au coin en question et à adapter le seuil de sorte qu'une configuration qui ne ne contiendrait pas ce point soit rejetée. Dans cet exemple, l'opération est la combinaison d'une érosion partielle volumique suivie d'une dilatation morphologique : $(X \overset{0,1}{\underset{\circ}{\ominus}} B) \oplus B$. Le disque noir représenté dans la dernière image montre bien qu'il n'y a aucune configuration où ce point est le seul élément manquant de B .

Nous n'avons pas à proprement parler utilisé les opérations volumiques parce que les opérations partielles ont suffi. Il n'empêche que l'idée des opérations volumiques ouvre d'intéressantes perspectives de recherche.

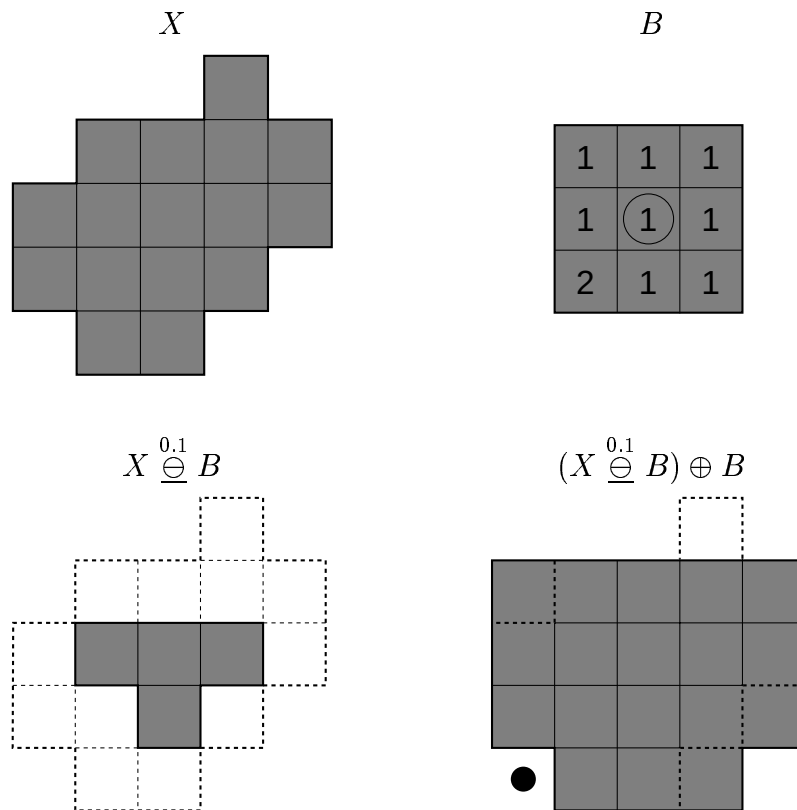


FIG. B.3 - Illustration de l'effet d'une érosion partielle volumique suivie d'une dilatation morphologique.

Bibliographie

- [1] S. Beucher. *Segmentation d'images et morphologie mathématique*. Thèse de doctorat, École nationale supérieure des mines de Paris, Juin 1990.
- [2] M. Biggar, O. Morris, and A. Constantinides. Segmented-image coding: performance comparison with the discrete cosine transform. *IEE Proceedings F, Radar and Signal Processing*, 135(2):121–132, April 1988.
- [3] H. Blum. An associative machine for dealing with the visual field and some of its biological implications. In E. Bernard and M. Kare, editors, *Biological prototypes and synthetic systems*, volume 1, pages 244–260. Plenum Press, New-York, 1962.
- [4] L. Bouchard, J. Cassas, F. Meyer, C. Oddou, L. Torres, and M. Van Droogenbroeck. *Coding of inside*. Deliverable D7 of Race Project Morpheco R2053, 51 pages, December 1993.
- [5] A. Bovik, M. Clark, and W. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, January 1990.
- [6] J. Brandt, A. Jain, and R. Algazi. Medial axis representation and encoding of scanned documents. *Journal of Visual Communication and Image Representation*, 2(2):151–165, June 1991.
- [7] J. Breen and P. Soille. Generalization of Van Herk recursive erosion/dilation algorithm to lines at arbitrary angles. In *Digital Image Computing: Techniques and Applications*, pages 549–555, Sydney, December 1993.
- [8] J. Candy. *Signal processing: The modern approach*. Mc Graw-Hill, 1988.
- [9] B. Chaudhuri. An efficient algorithm for running window pel gray level ranking in 2-D images. *Pattern Recognition Letters*, 11(2):77–80, February 1990.
- [10] S. Comes, S. Maes, and M. Van Droogenbroeck. Recent and prospective decorrelation techniques for image processing. *Annales des Télécommunications*, 48(7-8):390–403, July-August 1993.
- [11] M. Coster and J.-L. Chermant. *Précis d'analyse d'images*. Éditions du CNRS, Paris, 1985.
- [12] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169, July 1985.

- [13] P. Delogne. Estimation spectrale de séquences aléatoires. Chapitre 7 du cours de traitement numérique du signal, Université catholique de Louvain, 40 pages, Mai 1994.
- [14] P. Delogne and B. Macq. Universal variable length coding for an integrated approach to image coding. *Annales des Télécommunications*, 46(7-8):452–459, 1991.
- [15] D. Docampo-Amoedo and A. Figueiras-Vidal. A deconvolution approach to harmonic signal extrapolation. In *Int. Conf. on Acoustics, Speech and Signal Processing*, pages 2345–2348. IEEE, May 1989.
- [16] E. Dougherty. *An introduction to morphological image processing*, volume TT9 of *Tutorial Texts Series*. Spie Optical Engineering Press, Bellingham, 1992.
- [17] J. du Buf and P. Heitkämper. Texture features based on Gabor phase. *Signal Processing*, 23(3):227–244, June 1991.
- [18] B. Dubuisson. *Diagnostic et reconnaissance de formes*. Traité des nouvelles technologies. Hermès, Paris, 1990.
- [19] R. Duda and P. Hart. *Pattern classification and scene analysis*. Wiley-Interscience, 1973.
- [20] J. Dunham. Optimum uniform piecewise linear approximation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):67–75, January 1986.
- [21] P. Duvaut. *Traitement du signal: concepts et applications*. Hermès, Paris, 1991.
- [22] M. Eden and M. Kocher. On the performance of a contour coding algorithm in the context of image coding; part I: Contour segment coding. *Signal Processing*, 8(4):381–386, July 1985.
- [23] R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38(157):181–200, January 1982.
- [24] U. Franke. Selective deconvolution: A new approach to extrapolation and spectral analysis of discrete signals. In *Int. Conf. on Acoustics, Speech and Signal Processing*, pages 30.3.1–30.3.4. IEEE, May 1987.
- [25] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput.*, 10:260–268, June 1961.
- [26] K. Fu. *Digital pattern recognition*, volume 10 of *Communication and Cybernetics*. Springer-Verlag, Berlin, 1980.
- [27] R. Gerchberg. Super-resolution through error energy reduction. *Optica Acta*, 21(9):709–720, 1974.
- [28] J. Gil and M. Werman. Computing 2-D min, median, and max filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):504–507, May 1993.

- [29] M. Gilge, T. Engelhardt, and R. Mehlan. Coding of arbitrarily shaped image segments based on a generalized orthogonal transform. *Signal Processing: Image Communication*, 1(2):153–180, October 1989.
- [30] C. Gratin. *De la représentation des images au traitement morphologique d'images tridimensionnelles*. Thèse de doctorat, École nationale supérieure des mines de Paris, Janvier 1993.
- [31] M. Grimaud. A new measure of contrast: the dynamics. In *Image Algebra and Morphological Image Processing III, Vol. 1769*, pages 292–305, San Diego, July 1992. SPIE.
- [32] R. Haralick. Statistical and structural approaches to texture. *Proceedings of IEEE*, 67(5):786–804, May 1979.
- [33] R. Haralick, S. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):532–550, July 1987.
- [34] H. Heijmans and C. Ronse. The algebraic basis of mathematical morphology; I. Dilations and erosions. *Computer Graphics and Image Processing*, 50:245–295, 1990.
- [35] J. Hsiao and A. Sawchuk. Supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1279–1292, December 1989.
- [36] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(1):13–18, February 1979.
- [37] B. Hunt. Nonstationary statistical image models (and their application to image data compression). *Computer Graphics and Image Processing*, 12:173–186, 1980.
- [38] H. Jagadish and A. Bruckstein. On sequential shape descriptions. *Pattern Recognition*, 25(2):165–172, 1992.
- [39] A. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [40] B. Julesz. Textons, the elements of texture perception. *Nature*, 290:91–97, 1981.
- [41] R. Kresch and D. Malah. Morphological multi-structuring-element skeleton and its applications. In *International Symposium on Signals, Systems and Electronics*, pages 166–169, Paris, September 1992. URSI.
- [42] M. Kunt. *Techniques modernes de traitement numérique des signaux*, Volume 1 de *Traitement de l'information*. Presses polytechniques et universitaires romandes, Lausanne, 1991.
- [43] M. Kunt, M. Bénard, and R. Leonardi. Recent results in high-compression image coding. *IEEE Transactions on Circuits and Systems*, 34(11):1306–1336, November 1987.

- [44] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second generation image coding techniques. *Proceedings of IEEE*, 73:549–574, April 1985.
- [45] V. Lacroix. Pixel labelling in a second-order Markov mesh. *Signal Processing*, 12(1):59–82, January 1987.
- [46] C. Lantuéjoul. *La squelettisation et son application aux mesures topologiques de mosaïques polycristallines*. Thèse de doctorat, École nationale supérieure des mines de Paris, 1978.
- [47] B. Laÿ. Une méthode rapide pour trouver les érodés successifs d'une image binaire. Document interne au Centre de Morphologie Mathématique, N-920, 17 pages, Fontainebleau, Septembre 1984.
- [48] B. Laÿ. Recursive algorithms in mathematical morphology. In *Acta Stereologica*, Vol. 6, pages 691–696, Caen, 1987. ICS.
- [49] P. Lemaire and M. Van Droogenbroeck. A methodology for a universal contour-texture coding. Working document, Cost 211-bis; Simulation Subgroup, Martlesham, May 1990.
- [50] P. Lemaire and M. Van Droogenbroeck. *Traitement d'images: codage des contours et de la texture*. Travail de fin d'études, Université catholique de Louvain, Juin 1990.
- [51] F.-C. Leou and Y.-C. Chen. A contour-based image coding technique with its texture information reconstructed by polyline representation. *Signal Processing*, 3(1):81–89, October 1991.
- [52] M. Liou. Overview of the px64 kbit/s video coding standard. *Communications of the ACM*, 34(4):59–63, April 1991.
- [53] S. Lipschutz. *Topologie: cours et problèmes*. Série Schaum. Mc Graw-Hill, Paris, 1988.
- [54] K. Liu. *On chain coding of line drawings*. Delft University Press, 1992.
- [55] B. Macq. *Perceptual Transforms and Universal Entropy Coding for an Integrated Approach to Picture Coding*. PhD thesis, Université catholique de Louvain, September 1989.
- [56] S. Maes and M. Van Droogenbroeck. Edge characterization in hierarchical subband coding schemes. In *Fourth International Conference on Image Processing and its Applications*, pages 598–601, Maastricht, May 1992. IEE.
- [57] P. Maragos and R. Schafer. Morphological skeleton representation and coding of binary images. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(5):1228–1244, October 1986.
- [58] P. Maragos and R. Schafer. Morphological filters – Part II: their relations to median, order-statistic, and stack filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(8):1170–1184, August 1987.

- [59] G. Matheron. *Random sets and integral geometry*. John Wiley and Sons, New York, 1975.
- [60] G. Matheron. Examples of topological properties of skeletons. In J. Serra, editor, *Image analysis and mathematical morphology*, volume 2: Theoretical advances, chapter 11, pages 217–238. Academic Press, 1988.
- [61] G. Matheron. On the negligibility of the skeleton and the absolute continuity of erosions. In J. Serra, editor, *Image analysis and mathematical morphology*, volume 2: Theoretical advances, chapter 12, pages 239–256. Academic Press, 1988.
- [62] J. Mazille. Mathematical morphology and convolutions. *Journal of Microscopy*, 156:3–13, October 1989.
- [63] F. Meyer. Algorithmes ordonnés de ligne de partage des eaux. Document interne au Centre de Morphologie Mathématique, N-06/90/MM, Fontainebleau, Février 1992.
- [64] F. Meyer. Un algorithme optimal de ligne de partage des eaux. In *Proc. 8ème Congrès Reconnaissance des Formes et Intelligence artificielle*, pages 847–857, Paris, Novembre 1993. AFCET.
- [65] T. Minami and K. Shinohara. Encoding of line drawings with a multiple grid chain code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):269–276, March 1986.
- [66] M. Nelson. *La compression de données*. Dunod, Paris, 1993.
- [67] S. Pal and A. Gosh. Index of area coverage of fuzzy image subsets and object extraction. *Pattern Recognition Letters*, 11(12):831–841, December 1990.
- [68] A. Papoulis. A new algorithm in spectral analysis and band-limited extrapolation. *IEEE Transactions on Circuits and Systems*, 22(9):735–742, September 1975.
- [69] A. Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, 1984.
- [70] T. Pavlidis. *Structural pattern recognition*. Springer-Verlag, 1977.
- [71] J. Pecht. Speeding up successive Minkowski operations. *Pattern Recognition Letters*, 3(2):113–117, 1985.
- [72] W. Pennebaker and J. Mitchell. *JPEG Still image data compression standard*. Van Nostrand Reinhold, New York, 1993.
- [73] W. Pennebaker, J. Mitchell, G. Langdon, and R. Arps. An overview of the basic principles of the Q-Coder adaptive binary arithmetic coder. *IBM Journal of Research and Development*, 32(6):717–726, November 1988.
- [74] W. Philips. A fast algorithm for the generation of orthogonal base functions on a arbitrarily shaped region. In *Int. Conf. on Acoustics, Speech and Signal Processing*, pages 421–424, San Francisco, March 1992. IEEE.

- [75] I. Pitas and A. Venetsanopoulos. Morphological shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):38–45, January 1990.
- [76] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.
- [77] M. Rabbani and P. Jones. *Digital image compression techniques*, volume TT7 of *Tutorial Texts Series*. Spie Optical Engineering Press, Bellingham, 1991.
- [78] J.-F. Rivest. Xlim 3.1: Manuel de l'utilisateur. Document interne au Centre de Morphologie Mathématique, Fontainebleau, Juin 1992.
- [79] C. Ronse. Extraction of narrow peaks and ridges in images by combination of local low rank and Max filters: implementation and applications to clinical angiography. Philips Research Laboratory Belgium, Working Document WD47, 1988.
- [80] C. Ronse. Toggles of openings, and a new family of idempotent operators on partially ordered sets. *Applicable Algebra in Engineering, Communication and Computing*, 3:99–128, 1992.
- [81] C. Ronse and H. Heijmans. The algebraic basis of mathematical morphology; II. openings and closings. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54(1):74–97, July 1991.
- [82] C. Ronse and B. Macq. Morphological shape and region description. *Signal Processing*, 3(1):91–105, October 1991.
- [83] M. Sabri and W. Steenaart. An approach to band-limited signal extrapolation: the extrapolation matrix. *IEEE Transactions on Circuits and Systems*, 25(2):74–78, February 1978.
- [84] P. Salembier. *Multiscale image analysis and modeling using rank order filters – Application to defect detection*. PhD thesis, École polytechnique fédérale de Lausanne, 1991.
- [85] M. Schmitt. *Des algorithmes morphologiques à l'intelligence artificielle*. Thèse de doctorat, École nationale supérieure des mines de Paris, Février 1989.
- [86] M. Schmitt et J. Mattioli. *Morphologie mathématique*. Masson, Paris, 1993.
- [87] J. Serra. *Image analysis and mathematical morphology*. Academic Press, New York, 1982.
- [88] J. Serra (editor). *Image analysis and mathematical morphology*, volume 2: Theoretical advances. Academic Press, 1988.
- [89] B. Simon. *Morphologie mathématique appliquée au codage d'images par objets*. Travail de fin d'études, Université catholique de Louvain, Juin 1992.
- [90] B. Simon, B. Macq, and M. Van Droogenbroeck. Morphological image sketch coding. In *Workshop on Circuits, Systems and Signal Processing*, pages 251–256, Houthalen, March 1993. ProRISC IEEE Benelux.

- [91] D. Sinha and E. Dougherty. Characterization of fuzzy Minkowski algebra. In *Image Algebra and Morphological Image Processing III, Vol. 1769*, pages 59–69, San Diego, July 1992. SPIE.
- [92] D. Sinha and E. Dougherty. Fuzzy mathematical morphology. *Journal of Visual Communication and Image Representation*, 3(3):286–302, September 1992.
- [93] P. Soille. *Morphologie mathématique: du relief à la dimensionalité – Algorithmes et méthodes*. Thèse de doctorat, Université catholique de Louvain, Février 1992.
- [94] P. Soille and L. Vincent. Determining watersheds in digital pictures via flooding simulations. In *Visual Communications and Image Processing, Vol. 1360*, pages 240–250, Lausanne, October 1990. SPIE.
- [95] S. Sternberg. Grayscale morphology. *Computer Vision, Graphics, and Image Processing*, 35:333–355, 1986.
- [96] P. Strobach. Quadtree-structured recursive plane decomposition coding of images. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 39(6):1380–1397, June 1991.
- [97] M. Suk and S.-M. Chung. A new image segmentation technique based on partition mode test. *Pattern Recognition Letters*, 16(5):469–480, 1983.
- [98] H. Talbot. *Analyse morphologique de fibres minérales d’isolation*. Thèse de doctorat, École nationale supérieure des mines de Paris, Octobre 1993.
- [99] H. Tanaka and A. Leon-Garcia. Efficient run-length encodings. *IEEE Transactions on Information Theory*, 28(6):880–890, November 1982.
- [100] V. Tom, T. Quatieri, M. Hayes, and J. McClellan. Convergence of iterative nonexpansive signal reconstruction algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(5):1052–1058, October 1981.
- [101] R. van den Boomgaard. *Mathematical morphology: Extensions towards computer vision*. PhD thesis, Amsterdam University, March 1992.
- [102] M. Van Droogenbroeck. Rapport d’activités du premier mandat. Document interne au laboratoire de Télécommunications et Télédétection, Université catholique de Louvain, 150 pages, Septembre 1991.
- [103] M. Van Droogenbroeck. Rapport d’activités du deuxième mandat. Document interne au laboratoire de Télécommunications et Télédétection, Université catholique de Louvain, 158 pages, Août 1992.
- [104] M. Van Droogenbroeck. Algorithmes d’ouverture morphologique. Document interne au Centre de Morphologie Mathématique, N-17/93/MM, 22 pages, Fontainebleau, Mars 1993.
- [105] M. Van Droogenbroeck and L. Vandendorpe. Integration of the selective deconvolution algorithm into a multiresolution scheme. In *Image Processing: Algorithms and Techniques III, Vol. 1657*, pages 225–235, San Jose, February 1992. SPIE.

- [106] M. Van Droogenbroeck (éditeur). Xlim3d: Un logiciel de traitement d'image. Document interne au Centre de Morphologie Mathématique, N-33/93/MM, 252 pages, Fontainebleau, Décembre 1993.
- [107] M. van Herk. A fast algorithm for local minimum and maximum filters on rectangular and octogonal kernels. *Pattern Recognition Letters*, 13(7):517–521, July 1992.
- [108] L. van Vliet and B. Verwer. A contour processing method for fast binary neighbourhood operations. *Pattern Recognition Letters*, 7:27–36, 1988.
- [109] L. Vandendorpe. *Hierarchical coding of moving video signals*. PhD thesis, Université catholique de Louvain, October 1991.
- [110] L. Vincent. *Algorithmes morphologiques à base de files d'attente et de lacets, extension aux graphes*. Thèse de doctorat, École nationale supérieure des mines de Paris, Mai 1990.
- [111] L. Vincent. Morphological transformations of binary images with arbitrary structuring elements. *Signal Processing*, 22(1):3–23, January 1991.
- [112] G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):31–44, April 1991.
- [113] X. Zhuang and R. Haralick. Morphological structuring element decomposition. *Computer Vision, Graphics, and Image Processing*, 35(35):370–382, 1986.