



HAL
open science

Segmentation supervisée d'images texturées par régularisation de graphes

Cyrille Fauchaux

► **To cite this version:**

Cyrille Fauchaux. Segmentation supervisée d'images texturées par régularisation de graphes. Traitement des images [eess.IV]. Université François-Rabelais de Tours, France, 2013. Français. NNT : . tel-01131729

HAL Id: tel-01131729

<https://hal.science/tel-01131729>

Submitted on 15 Mar 2015

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

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

Remerciements

Merci à :

- Romuald Boné, Julien Olivier et Pascal Makris pour m’avoir offert l’occasion de réaliser cette thèse et m’avoir encadré durant ces quatre années, y compris depuis l’autre bout du monde ou sur le départ pour la maternité.
- Jean-Christophe Pittet qui, dans le cadre de ma bourse CIFRE, m’a accueilli pendant plus de trois ans au sein de son entreprise Cosmo Laboratoire et m’a permis de découvrir le merveilleux monde de la cosmétique.
- Messieurs Mohamed Daoudi et Olivier Lézoray pour avoir accepté la lourde tâche de rapporter cette thèse, ainsi que messieurs Atila Baskurt et Laurent Najman pour avoir participé à mon jury.
- Toute l’équipe pédagogique du Département Informatique de Polytech Tours pour tout le savoir qu’ils m’ont transmis et pour m’avoir accueilli pendant tant d’années, ainsi qu’à Alexandre Lissy pour son génie de l’informatique dont il a bien voulu me faire profiter.
- Betty, les Karines, Julie, Jessica et Charles pour leur bonne humeur constante et pour avoir été des collègues admirables.
- Tous mes amis les Oufs avec qui j’ai passé de si bons moments.
- Et bien entendu mes parents, sans qui tout cela n’aurait jamais été possible.

REMERCIEMENTS

Résumé

Une méthode de traitement d'images est qualifiée de non-locale lorsqu'elle permet de traiter une image avec un plus haut niveau d'abstraction que celui des pixels qui la composent. Alors que la complexité des problèmes de vision par ordinateurs ne cesse d'augmenter, il est devenu clair que la non-localité est une solution aux limitations de certaines méthodes.

Les travaux présentés dans cette thèse s'inscrivent dans le cadre de la segmentation d'images. Ils améliorent une récente méthode non-locale de segmentation [Ta, 2009] dont la particularité est de représenter les images sous forme de graphes de similarité. L'intérêt de ce type de structure de données est d'autoriser l'établissement de relations entre des pixels non-adjacents et ainsi de mettre en évidence certaines structures ou motifs dans les images.

Cette méthode de segmentation se base sur un processus de régularisation de graphes. La segmentation est représentée par une fonction indicatrice dont la qualité est évaluée à l'aide d'une énergie selon deux critères : la régularité et la précision (pour ce dernier, on parle de critère d'attache aux données). La segmentation optimale correspond à celle qui établit un compromis entre ces deux critères et minimise l'énergie associée. Sous certaines conditions, ce problème de minimisation correspond à un problème d'optimisation globale et peut être approximé à l'aide d'un algorithme itératif.

La méthode définie par Ta se limite cependant à un critère d'attache aux données basé sur l'homogénéité des niveaux de gris. Or, en présence de bruit ou de régions texturées, le niveau de gris constitue une information trop bas niveau pour permettre une bonne identification de la classe d'appartenance de chacun des pixels. Afin de remédier à cela, nous proposons de la combiner à une méthode de caractérisation de textures, et ainsi tirer parti de deux approches non-locales.

Nous présentons ainsi une première extension de la méthode de segmentation par régularisation de graphe. En prolongeant des travaux de Chan et Vese [Chan *et al.*, 2000], nous définissons un nouveau terme d'attache aux données capable de mesurer l'homogénéité d'un ensemble de caractéristiques de textures. Évaluée sur des images échographiques 2D et 3D, les résultats obtenus montrent une amélioration significative de la qualité de segmentation.

Notre première méthode est cependant sensible à la pertinence des caractéristiques de textures, et une pondération manuelle de chacune d'entre elles serait nécessaire afin d'obtenir les meilleurs résultats. Nous proposons donc une seconde approche dans laquelle l'attache aux données est désormais évaluée à l'aide d'un classificateur supervisé (implémenté sous la forme de réseaux de neurones artificiels) permettant ainsi une modélisation automatique

RÉSUMÉ

des textures à identifier. Nous profitons par ailleurs de la présence de ce classificateur pour permettre à notre méthode de réaliser une segmentation multiclass. Évaluée sur des données issues de différentes techniques d'imagerie médicale, cette méthode présente une précision de segmentation accrue tout en offrant un paramétrage simplifié.

Mots clés : Segmentation, analyse de textures, graphe, régularisation, classificateur supervisé, réseau de neurones artificiels, analyse d'images médicales, images 3D.

Abstract

An image processing method is said to be non-local if it processes the image with a higher level of abstraction than the pixels. As the complexity of the computer vision problems is increasing, it becomes clear that the non-locality will help to overcome some traditional algorithms limitations.

The work presented in this thesis deals with image segmentation. It is based on a recent non-local image segmentation approach [Ta, 2009] whose particularity is to represent images with similarity graphs. The advantage of this type of data structure is to efficiently express relations between non-adjacent pixels, allowing to highlight patterns in the images.

This segmentation method is performed by a graph regularization process. Segmentation is expressed by an indicator function whose quality is evaluated by an energy according to two criteria: the regularity and the fidelity to the data. The optimal segmentation satisfies both and minimizes the energy. Under some conditions, this minimization problem is a global optimization one, and the solution can be approximated with an iterative algorithm.

One drawback of this method is linked to its data fidelity criteria being based on the homogeneity of grayscale values. When working with textured or noisy images, the information represented by a pixel intensity might not be enough to allow the identification of the pixel's class. In order to address this issue, we propose to combine this non-local image processing approach with a second one: the texture features.

First, we propose to extend the graph regularization method. Based on the work of Chan and Vese [Chan *et al.*, 2000], we define a fidelity criteria which is able to measure the homogeneity of a set of texture features. Evaluated on 2D and 3D ultrasound images, our results show an improvement of the quality of segmentation.

Our first method is, however, sensitive to the relevancy of the features. The user is required to manually set several parameters in order to obtain the best results. Therefore, we propose a second approach in which data fidelity is evaluated by a supervised classifier (currently implemented as a neural network). The classifier being trained to identify the different textures in an image, the most relevant features are automatically identified. Our method is also extended to support any number of classes. This method is evaluated on data from several image acquisition techniques, highlighting its increased generality and accuracy.

Keywords: Segmentation, textures analysis, graph, regularization, supervised classifier, artificial neural network, medical images analysis, 3D images.

ABSTRACT

Publications de l'auteur

C. Fauchaux, J. Olivier, R. Boné et P. Makris. Segmentation d'images échographique 2D et 3D par régularisation de graphe basée texture. *In COMpression et REprésentation des Signaux Audiovisuels*, pages 127–132. 2012.

C. Fauchaux, J. Olivier, R. Boné et P. Makris. Texture-based graph regularization process for 2D and 3D ultrasound image segmentation. *In Proceedings of the International Conference on Image Processing*, pages 2333–2336. IEEE. 2012.

C. Fauchaux, J. Olivier et R. Boné. Graph-Based Regularization of Binary Classifiers for Texture Segmentation. *In Computer Analysis of Images and Patterns*, volume 8048 de *Lecture Notes in Computer Science*, pages 310–318. Springer Berlin Heidelberg. 2013.

Table des matières

Introduction	21
1 Méthodes de segmentation d'images à base de graphes	25
1.1 Introduction	25
1.2 Généralités sur les graphes, définitions et notations	26
1.3 Graphes de similarité	27
1.3.1 Sélection de l'ensemble des nœuds	27
1.3.2 Sélection de l'ensemble des arêtes	28
1.3.3 Sélection de la mesure de similarité	30
1.4 Méthodes à base d'arbres de couverture minimale	31
1.5 Méthodes à base de coupes minimales et approches spectrales	33
1.6 Marche aléatoire	36
1.7 Segmentation par régularisation de graphes	37
1.7.1 Opérateurs différentiels définis sur les graphes	37
1.7.2 Fonctionnelles de régularisation	39
1.7.3 Première classe de modèles variationnels	40
1.7.4 Seconde classe de modèles variationnels	42
1.8 Conclusion	46
2 Segmentation d'images texturées par régularisation de graphes	47
2.1 État de l'art des méthodes de caractérisation de textures	47
2.1.1 Caractéristiques d'Haralick	48
2.1.2 Modèles auto-régressifs	50
2.1.3 Caractéristiques de textures dérivées de méthodes de traitement du signal	51
2.1.4 Modèle binaire local	60
2.1.5 Champs de Markov aléatoires	61
2.1.6 Bilan	63
2.2 Contribution	65

TABLE DES MATIÈRES

2.3	Résultats	68
2.3.1	Segmentation d'images multispectrales	68
2.3.2	Segmentation d'images texturées artificielles	70
2.3.3	Segmentation d'images échographiques 2D et 3D	78
2.4	Conclusion	83
3	Segmentation d'images texturées par régularisation de classificateurs	85
3.1	Classification supervisée de données	86
3.1.1	Méthode des k plus proches voisins	88
3.1.2	Réseaux de neurones artificiels	89
3.1.3	Machines à vecteurs supports	91
3.1.4	Bilan des méthodes de classification supervisée	93
3.2	Contribution	94
3.2.1	Segmentation biclasse	94
3.2.2	Segmentation multiclasse	95
3.2.3	Architecture et entraînement des réseaux de neurones	96
3.2.4	Choix de l'algorithme de régularisation	97
3.2.5	Application à la segmentation automatique d'images	98
3.3	Résultats	99
3.3.1	Segmentation d'images texturées artificielles	99
3.3.2	Segmentation d'images médicales 2D	103
3.3.3	Segmentation d'images IRM 3D	106
3.4	Conclusion	112
	Conclusion générale et perspectives	113
	Annexes	119
A	Caractéristiques d'Haralick	119
A.1	Notations et calculs préliminaires	119
A.2	Propriétés utiles	120
A.3	Caractéristiques de textures	121
A.3.1	<i>Angular second moment</i>	122
A.3.2	<i>Contrast</i>	122
A.3.3	<i>Correlation</i>	123
A.3.4	<i>Sum of squares: Variance</i>	124
A.3.5	<i>Inverse difference moment</i>	124
A.3.6	<i>Sum average</i>	125

TABLE DES MATIÈRES

A.3.7	<i>Sum variance</i>	125
A.3.8	<i>Sum entropy</i>	126
A.3.9	<i>Entropy</i>	126
A.3.10	<i>Difference variance</i>	127
A.3.11	<i>Difference entropy</i>	128
A.3.12	Autres mesures de corrélation	128

TABLE DES MATIÈRES

Liste des tableaux

2.1	Évaluation de l'influence des paramètres λ_1 et λ_2 de la méthode de segmentation par régularisation de graphe : distance générique d'anomalie.	74
2.2	Évaluation de l'influence des paramètres λ_1 et λ_2 de la méthode de segmentation par régularisation de graphe : distance de Hausdorff.	74
2.3	Influence des paramètres λ_1 et λ_2 de la méthode de segmentation par régularisation de graphe sur la vitesse de convergence.	76
2.4	Évaluation de la capacité à gérer des caractéristiques de faible pertinence : qualité du résultat.	77
2.5	Évaluation de la capacité à gérer des caractéristiques de faible pertinence : vitesse de convergence.	78
2.6	Segmentation basée texture d'images échographiques 2D par régularisation de graphes : distance générique d'anomalie.	79
3.1	Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur la série d'images texturées artificielles.	100
3.2	Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur des images échographiques.	103
3.3	Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur des images de microscopie confocale.	105
3.4	Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur une image 3D dans un contexte multiclasse.	108

LISTE DES TABLEAUX

Table des figures

1.1	Exemple de graphe orienté des k plus proches voisins.	30
1.2	Exemple d'arbre de couverture minimale.	32
1.3	Utilisation d'un arbre de couverture minimale pour la segmentation de données.	32
1.4	Exemple de segmentation obtenue par la méthode des coupes normalisées minimales.	36
2.1	Exemples de textures visuellement différentes mais dont les statistiques de premier ordre concernant les niveaux de gris sont identiques. Dans chacune des deux régions, le niveau de gris moyen vaut 192.	48
2.2	Exemple de matrice de co-occurrences.	50
2.3	Exemples de textures issues de la base d'images Brodatz et leur spectre de puissance.	53
2.4	Régions du spectre de puissance permettant l'identification de la fréquence et de l'orientation des composantes fréquentielles d'une texture.	53
2.5	Décomposition en ondelettes d'un signal 1D.	55
2.6	Décomposition en ondelettes d'un signal 2D.	56
2.7	Exemples de filtres de Gabor.	58
2.8	Filtres de Schmid.	58
2.9	Filtres de Leung-Malik.	59
2.10	Banc de filtres MR4.	60
2.11	Méthode de calcul d'un motif binaire local.	61
2.12	Ordre du voisinage pour un site d'un champ de Markov.	62
2.13	Cliques de second ordre utilisées lors de la modélisation d'un phénomène par champs de Markov.	62
2.14	Exemple d'image multispectrale.	69
2.15	Résultats obtenus après exécution de l'algorithme de régularisation de graphe sur une image multi-spectrale artificielle.	71
2.16	Exemple d'image texturée artificielle.	71
2.17	Structures typiques des matrices de co-occurrences des deux régions de l'image texturée artificielle.	72

TABLE DES FIGURES

2.18	Représentation normalisée des caractéristiques d’Haralick obtenues pour chaque pixel de l’image 2.16a.	73
2.19	Régularisation d’un contour de forte courbure.	74
2.20	Illustration de l’influence de la régularisation sur les contours de forte courbure. 75	
2.21	Comparaison des résultats obtenus par différentes techniques de segmentation sur l’image échographique 2D N° 1.	80
2.22	Comparaison des résultats obtenus par différentes techniques de segmentation sur l’image échographique 2D N° 5.	81
2.23	Illustration des résultats obtenus sur une image échographique 3D.	82
3.1	Illustration de la notion d’apprentissage semi-supervisé dans un graphe. . .	87
3.2	Schéma d’un neurone formel j de type perceptron.	90
3.3	Structure d’un perceptron multicouche.	91
3.4	Principe de marge maximale.	92
3.5	Exemple de problème qui n’est linéairement séparable que lorsque les données sont projetées dans un espace de plus grande dimension.	93
3.6	Processus de segmentation biclasse par régularisation de classificateurs. . .	95
3.7	Processus de segmentation multiclasse par régularisation de classificateurs. .	96
3.8	Exemple d’image présentant un objet texturé de forme aléatoire.	99
3.9	Segmentations obtenues par les différentes méthodes sur l’image texturée artificielle 1.	101
3.10	Segmentations obtenues par les différentes méthodes sur l’image texturée artificielle 7.	102
3.11	Segmentations obtenues par les différentes méthodes sur l’image texturée artificielle 14.	102
3.12	Image échographique utilisée pour entraîner le classificateur.	104
3.13	Résultats obtenus sur l’image échographique 2D n° 2 par différentes méthodes de segmentation.	104
3.14	Résultats obtenus sur l’image échographique 2D n° 3 par différentes méthodes de segmentation.	105
3.15	Résultat obtenus par la méthode de segmentation par régularisation de classificateurs sur une image de microscopie confocale.	105
3.16	Exemple d’image de cerveau acquise en imagerie par résonance magnétique. 106	
3.17	Segmentation obtenue par la méthode de régularisation de classificateurs sur une coupe d’image IRM 3D.	107
3.18	Segmentation du cerveau obtenue par la méthode de régularisation de classificateurs sur une image IRM 3D : substance grise.	109
3.19	Segmentation du cerveau obtenue par la méthode de régularisation de classificateurs sur une image IRM 3D : substance blanche.	110

TABLE DES FIGURES

- 3.20 Segmentation du cerveau obtenue par la méthode de régularisation de classificateurs sur une image IRM 3D : substance blanche et substance grise. 111

TABLE DES FIGURES

Introduction

L'une des problématiques des réseaux sociaux consiste à identifier au sein d'une population les individus qui pourraient se connaître. Naïvement, cette problématique est relativement simple à résoudre. Considérons que nous disposons pour chaque personne d'une liste potentiellement incomplète de ses connaissances. N'importe quel informaticien est capable d'entrer ces individus et les relations qui les lient dans une base de données relationnelle, et d'effectuer des requêtes afin d'identifier les amis des amis, les amis des amis des amis... D'après la théorie des six degrés de séparation¹, quelques jointures suffisent à identifier et proposer à tout individu une liste de ses connaissances potentielles.

Dans un autre registre, un moteur de recherche peut être construit aussi facilement (pour les besoins de la démonstration, nous supposons être en possession d'un oracle capable d'associer des mots clés aux pages et de mesurer la similarité des sujets traités dans chaque paire de pages Web). Les pages Web sont alors indexées à l'aide de mots-clés et des relations sont établies entre des pages Web « similaires ».

Les choses se compliquent cependant lors du passage à l'échelle. Les jointures nécessaires à l'exécution de ces requêtes font en effet exploser l'aspect combinatoire du problème, et un tel système deviendra inutilisable au-delà d'un certain nombre d'individus. Il en est de même lorsque l'on souhaite obtenir toutes les pages Web traitant d'une certaine thématique.

Fin 2012, Facebook annonçait pourtant avoir atteint le milliard d'utilisateurs actifs, et Google les 50 milliards de pages Web indexées. Par quelles méthodes arrivent-ils à traiter de si grands volumes de données ?

Une solution consiste à ne pas utiliser de bases de données relationnelles, mais à représenter les données à l'aide de graphes. Ces problèmes s'y prêtent naturellement : dans un réseau social, les individus sont représentés par les nœuds et les relations par les arêtes. L'identification des connaissances potentielles d'un individu revient alors simplement à effectuer un parcours en largeur du graphe. Il en est de même pour le moteur de recherche, à la différence qu'il est en plus possible de pondérer les relations entre les pages en fonction d'un éventuel degré de similarité.

Des méthodes issues de la théorie des graphes peuvent par ailleurs être appliquées sur ces données. Citons par exemple les algorithmes de détection de communautés, lesquels peuvent servir à identifier automatiquement des groupes d'âges dans un réseau social [Pons, 2005], ou encore des thèmes importants dans un moteur de recherche [Véronis, 2003].

1. La théorie des six degrés de séparation énonce que toute personne sur le globe peut être reliée à n'importe quelle autre au travers d'une chaîne de relations comprenant au plus cinq autres maillons.

Les graphes permettent naturellement de représenter tous types de données discrètes, car même s'il n'existe pas de relations naturelles entre des données, il est toujours envisageable de se baser sur une mesure de similarité ou de proximité afin de refléter une certaine forme d'organisation.

Le sujet auquel nous nous intéressons dans cette thèse concerne l'utilisation en vision par ordinateur de méthodes d'analyse de grands volumes de données basées sur les graphes.

La vision par ordinateur est une branche de l'intelligence artificielle qui vise à produire des machines et méthodes capables d'identifier les différents éléments d'une scène perçue au travers d'appareils d'imagerie numérique. Cette problématique est aujourd'hui plus importante que jamais car les récentes avancées en matière d'imagerie numérique, particulièrement en imagerie médicale, permettent aujourd'hui d'acquérir des contenus multimédias de très grandes tailles : les photos obtenues par un appareil grand public sont composées de plus de dix millions de pixels, tandis qu'un microscope confocal est en mesure de produire des images 3D contenant jusqu'à cent-cinquante millions de pixels. Combiné à une réduction des coûts de certains matériels d'imagerie, il nous est désormais possible de produire plus d'images que ce que des experts humains peuvent analyser.

Ce n'est cependant pas la taille de ces images qui nous pousse à nous intéresser à ces approches. En effet, de nombreuses méthodes de traitement d'images commencent aujourd'hui à montrer leurs limites et ne sont pas capables de traiter des problèmes de vision aussi complexes que nous le souhaiterions. Les méthodes d'analyse de données basées sur les graphes ayant donné un nouveau souffle aux problématiques de traitement de grands volumes de données, nous pensons pertinent de les appliquer au traitement d'images.

Les graphes disposent en particulier d'une propriété intéressante : ils permettent aisément d'établir des relations entre des pixels non-adjacents, et ainsi de traiter les images avec un plus haut niveau d'abstraction que celui des pixels qui les composent. Cette propriété est appelée « non-localité » et est pressentie par la communauté scientifique comme une des solutions aux limitations précédemment évoquées.

La vision par ordinateur a toujours été un domaine de recherche très actif. Au-delà de l'aspect utile, permettre à une machine de comprendre ce qu'elle voit représente un énorme défi. Le système de vision humain (et animal) est en effet utilisé instinctivement pour résoudre des problèmes complexes de vision, mais son fonctionnement est encore mal compris.

Il a tout de même été démontré que pour comprendre une scène, le système visuel humain fonctionne par étapes : les objets sont tout d'abord identifiés individuellement puis sont comparés à des représentations stockées en mémoire afin de leur donner une signification. Cette thèse s'inscrit dans le cadre de l'identification des objets, couramment appelée « segmentation d'images ».

Des expériences ont par ailleurs démontré que pendant cette phase de segmentation, certaines zones du cortex visuel ont des rôles bien précis [Daugman, 1987, DiCarlo *et al.*, 2012]. Ces rôles concernent notamment la caractérisation de l'orientation, de la couleur ou encore de la texture des objets.

Le mécanisme de caractérisation de textures du cerveau est lui aussi mal compris.

Nous disposons cependant en traitement d'images de plusieurs modèles mathématiques, statistiques et structurels qui, sans être parfaits, permettent de caractériser une texture selon différents critères tels que l'orientation, la fréquence, le contraste, la régularité, la finesse...

Ces modèles constituent eux aussi des outils non-locaux. Lorsque la texture d'un pixel est caractérisée, il est en effet nécessaire de tenir compte du niveau de gris des pixels voisins. Il s'agit donc de données de plus haut niveau d'abstraction que le pixel.

L'objectif de nos travaux est de permettre à une méthode de segmentation par régularisation de graphes [Ta, 2009] de tirer parti de caractéristiques de textures afin de mieux identifier les différentes régions des images.

Le premier chapitre sera consacré aux graphes. Nous verrons tout d'abord quelles sont les différentes façons de représenter une image à l'aide d'un graphe, puis enchaînerons sur un état de l'art des différentes méthodes de segmentation d'images à base de graphes.

Le second chapitre débutera par un état de l'art des méthodes de caractérisation de textures. Nous enchaînerons ensuite sur une présentation de notre première méthode de segmentation d'images texturées par régularisation de graphes. Celle-ci prolonge les travaux de Chan et Vese [Chan *et al.*, 2000] et s'appuie sur un critère de segmentation capable d'évaluer l'homogénéité des caractéristiques de textures à l'intérieur de chaque région. Nous analyserons ensuite son comportement sur des images texturées synthétiques et mettrons en évidence le rôle de ses différents paramètres. Nous évaluerons ensuite ses performances sur des images échographiques 2D et 3D.

Dans le troisième chapitre, nous présenterons une seconde méthode de segmentation d'images texturées. Afin de simplifier le paramétrage de la première, nous remplaçons le critère d'homogénéité par un classificateur supervisé (un état de l'art des principaux classificateurs sera préalablement réalisé). L'objectif de ce classificateur est de permettre l'identification des caractéristiques de textures les plus pertinentes afin d'offrir une meilleure modélisation du problème. Nous proposons par ailleurs d'étendre cette méthode à la problématique de segmentation multiclasse. Les performances de cette méthode seront évaluées sur des images synthétiques, des images échographiques, des images de microscopie confocale et enfin sur des images d'IRM 3D multiclassées.

Nous terminerons par une conclusion dans laquelle nous établirons un bilan de nos méthodes et discuterons de plusieurs perspectives visant à améliorer leur précision.

Le lecteur intéressé trouvera en annexe une liste détaillée des différentes caractéristiques d'Haralick dans laquelle nous proposons une évaluation de leurs domaines de valeurs ainsi qu'une interprétation de leur signification. Nous discutons enfin de la stabilité numérique de certaines d'entre elles.

Chapitre 1

Méthodes de segmentation d'images à base de graphes

1.1 Introduction

Un reproche qui peut être fait aux algorithmes traditionnels de traitement d'images est le manque de flexibilité des structures de données impliquées. L'organisation sous forme de simple grille génère en effet d'importantes contraintes vis-à-vis de leur exécution, limitant leur potentiel. Cela s'illustre particulièrement pour des problématiques liées à la « forme » des données. Par exemple, de nombreux algorithmes sont – sur le principe – capables de traiter des données de plus grande dimension, telles que des images tri-dimensionnelles ou encore des vidéos. Cependant, si un tel algorithme n'a pas été dès l'origine implémenté de façon à supporter des données de plus haute dimension, il est relativement complexe de le faire évoluer dans ce sens, et des développements spécifiques sont souvent requis¹.

De par leur structure, les graphes sont capables de représenter tout domaine discret. Il est ainsi possible de construire une représentation d'une image à l'aide d'un graphe en associant un nœud du graphe à chaque pixel. Par la suite, en établissant des relations entre les nœuds correspondants à des pixels voisins dans l'image, il est possible d'obtenir un graphe dont la topologie reflète l'organisation initiale des pixels, notamment les relations de voisinage.

Cette représentation est beaucoup plus souple. Étant donné qu'il n'y a plus de notion de « bord de l'image », exécuter un algorithme sur une région d'intérêt – qu'elle soit rectangulaire ou non – peut simplement revenir à travailler sur un sous-graphe composé uniquement des nœuds appartenant à cette région. Par ailleurs, la structure de graphe faisant abstraction des dimensions de l'image, il est envisageable de traiter des données bi et tridimensionnelles exactement de la même manière ; seule l'étape de construction du graphe nécessite d'être adaptée aux données manipulées.

Comme nous le verrons plus loin dans ce chapitre, de nombreux outils de traitement d'images basés sur les graphes reposent sur des algorithmes issus d'autres domaines, princi-

1. Nous pouvons arriver à une conclusion similaire en considérant la problématique de traitement de région d'intérêt de formes arbitraires, en particulier non rectangulaires.

palement de la théorie des graphes. Il devient alors possible de concevoir des algorithmes de traitement d'images à partir d'approches conçues pour des problématiques totalement différentes. Réciproquement, les graphes rendent possible d'appliquer des algorithmes de traitement d'images à des données qui ne sont pas des images. Ainsi, un algorithme de débruitage peut, par exemple, servir à lisser un maillage [Elmoataz *et al.*, 2008].

Ce chapitre constitue un état de l'art des méthodes de traitement d'images à base de graphes. Nous réaliserons tout d'abord un rappel de notions usuelles de la théorie des graphes, puis enchaînerons sur les différentes méthodes permettant de représenter une image à l'aide d'un graphe. Nous terminerons ce chapitre par une revue des différentes familles de méthodes de segmentation d'images basées sur les graphes.

1.2 Généralités sur les graphes, définitions et notations

Un graphe est une représentation d'un ensemble d'éléments dans laquelle des relations peuvent être établies entre toutes paires d'éléments. Ceux-ci sont représentés par des nœuds, et les relations par des arêtes. Un graphe $G = (V, E)$ est ainsi composé d'un ensemble $V = \{V_1 \dots V_N\}$ de N nœuds et d'un ensemble $E \subset V \times V$ d'arêtes. Dans ce manuscrit, nous ne nous intéressons qu'aux graphes simples (il n'existe pas d'arête reliant un nœud à lui-même et il y a au plus une arête reliant deux nœuds) et non-orientés (si une relation existe entre deux nœuds, celle-ci est symétrique).

Deux nœuds $u, v \in V$ reliés par une arête $(u, v) \in E$ sont dit adjacents ou encore voisins. Cette relation est notée $u \sim v$. Réciproquement, l'arête (u, v) est dite incidente aux nœuds u et v .

Le voisinage $\mathcal{N}(u)$ d'un nœud u correspond quant à lui à l'ensemble des nœuds adjacents à u :

$$\mathcal{N}(u) = \{v \in V \text{ tel que } (u, v) \in E\}$$

On note $\mathcal{H}(V)$ l'espace de Hilbert des fonctions scalaires à valeurs réelles $f : V \rightarrow \mathbb{R}$ définies sur l'ensemble des nœuds et $\mathcal{H}(E)$ l'espace de Hilbert des fonctions scalaires à valeurs réelles $w : E \rightarrow \mathbb{R}$ définies sur l'ensemble des arêtes. Ces ensembles sont munis des produits scalaires suivants :

$$\langle f, h \rangle_{\mathcal{H}(V)} = \sum_{u \in V} f(u) \cdot h(u) \quad \text{avec } f, h : V \rightarrow \mathbb{R} \quad (1.1)$$

$$\langle F, H \rangle_{\mathcal{H}(E)} = \sum_{(u,v) \in E} F(u, v) \cdot H(u, v) \quad \text{avec } F, H : E \rightarrow \mathbb{R} \quad (1.2)$$

Une fonction $w \in \mathcal{H}(E)$ associe à chaque arête $(u, v) \in E$ une valeur réelle $w(u, v) \in \mathbb{R}$ (aussi notée w_{uv}). Une telle fonction est généralement appelée fonction de pondération, et permet de définir un graphe pondéré noté $G = (V, E, w)$.

Nous noterons par ailleurs $\mathcal{H}^m(V)$ l'espace de Hilbert des fonctions vectorielles à valeurs réelles $f : V \rightarrow \mathbb{R}^m$ définies sur l'ensemble des nœuds. Ces fonctions sont généralement utilisées pour représenter les données associées à chaque nœud, sous forme de vecteur d'attributs. La notation indicée $f_i(\cdot)$, $i \in \{1 \dots m\}$ permet d'accéder à chacune des composantes du vecteur.

1.3 Graphes de similarité

La plupart des méthodes de segmentation d'images à base de graphes utilisent des graphes de similarité. Leur rôle est de permettre la représentation de relations entre les composantes de l'image possédant des caractéristiques semblables, la fonction de pondération servant alors à moduler ces relations en représentant un « degré de similarité ».

Ces composantes peuvent être de différents types. Comme évoqué précédemment, avec une approche bas niveau, chaque nœud représente un pixel de l'image. Nous verrons plus loin qu'il est également possible de considérer des éléments de plus haut niveau, tels que des régions de l'image, permettant ainsi de diminuer la taille du graphe et d'accélérer son traitement.

Le point commun entre les différents types de composantes est qu'il leur est associé un ou plusieurs attributs les caractérisant, représentés par des fonctions de $\mathcal{H}(V)$ ou $\mathcal{H}^m(V)$. Ceux-ci correspondent aux données à traiter et peuvent, là encore, être de types très différents : coordonnées [Bougleux *et al.*, 2007], intensité du niveau de gris [Felzenszwalb et Huttenlocher, 2004, Grady, 2006], caractéristiques colorimétriques [Shi et Malik, 2000, Lezoray *et al.*, 2007], caractéristiques de textures [Shi et Malik, 2000], patches d'images [Lezoray *et al.*, 2008]... Nous reviendrons dans un prochain chapitre sur les principales méthodes de caractérisation de textures.

La construction d'un graphe de similarité peut être découpée en trois étapes, que nous allons détailler dans les sections suivantes :

- la sélection de l'ensemble des nœuds,
- la sélection de l'ensemble des arêtes,
- la sélection de la mesure de similarité.

1.3.1 Sélection de l'ensemble des nœuds

En ce qui concerne la sélection de l'ensemble des nœuds, l'approche la plus naturelle consiste à représenter chaque donnée par un nœud du graphe [Chan *et al.*, 2001, Bougleux, 2007, Lezoray *et al.*, 2008]. L'intérêt est ici d'avoir une représentation la plus fine possible des interactions entre les données.

Malheureusement, le nombre d'arêtes potentielles augmentant de façon quadratique avec le nombre de nœuds, ce niveau de détails peut avoir un coût non négligeable. Cela est particulièrement vrai dans le cadre du traitement d'images où les données manipulées sont les pixels, dont le nombre peut être relativement important. Nous verrons dans la section suivante les solutions qui permettent de conserver les problèmes de segmentation solvables tout en maintenant un maximum d'informations utiles.

L'alternative consiste à réduire le nombre de nœuds en employant une approche de type « superpixel ». Cela revient à appliquer un algorithme de *clustering* comme prétraitement de l'image dans le but de produire une sur-segmentation de celle-ci. La partition ainsi obtenue est constituée de régions (ou superpixels) présentant une très faible variabilité intrarégion. Le but de cette étape de *clustering* n'étant pas d'anticiper sur la segmentation finale², cette propriété est extrêmement importante. En effet, c'est elle qui permet de

2. La capacité de détection des contours des objets aura tout de même une influence certaine sur la

regrouper des pixels dont la dissociation – du fait de leur très forte ressemblance – n’aurait fourni aucune information supplémentaire à l’algorithme de segmentation qui sera par la suite appliqué.

Cette approche a notamment été utilisée dans [Ta, 2009], où l’auteur utilise l’algorithme de partitions d’énergie [Arbeláez et Cohen, 2004] pour effectuer une sur-segmentation des images et créer ses graphes, réduisant ainsi le nombre de nœuds à moins de 3 % du nombre initial de pixels.

Le lecteur intéressé pourra se référer à [Achanta *et al.*, 2012] pour un état de l’art des méthodes de « superpixellisation ».

1.3.2 Sélection de l’ensemble des arêtes

Choisir l’ensemble des arêtes consiste à définir la topologie du graphe, en particulier par la spécification du voisinage de chaque nœud. Avant d’entrer dans le détail des méthodes existantes, il faut garder à l’esprit qu’il est difficile de prédire quelle topologie de graphe permettra d’obtenir les meilleurs résultats. Le choix des heuristiques sous-jacentes est dépendant de l’algorithme utilisé et des données à traiter [Maier *et al.*, 2008].

Les méthodes de construction que nous allons décrire se basent généralement sur une mesure de distance permettant d’évaluer la proximité de deux nœuds. Cette proximité n’est cependant pas nécessairement géométrique ; elle peut tout à fait se baser sur une notion de similarité des caractéristiques associées aux nœuds, ou encore combiner distance géométrique et caractéristique. De nouveau, il n’existe pas de règle permettant de définir la mesure la plus adaptée ; ce choix devra être fait en fonction de l’objectif recherché et des données à traiter.

Soient $A \subset \mathbb{R}^m$ et $p, q \in A$. Les trois mesures de distance les plus utilisées sont :

- la norme L_1 , ou distance de Manhattan :

$$d(p, q) = \sum_{i=1}^m |p_i - q_i| \quad , \quad (1.3)$$

- la norme L_2 , ou distance Euclidienne :

$$d(p, q) = \sum_{i=1}^m (p_i - q_i)^2 \quad , \quad (1.4)$$

- la norme L_∞ , ou distance de Chebychev :

$$d(p, q) = \max_i |p_i - q_i| \quad . \quad (1.5)$$

Dans l’absolu, la majorité des algorithmes de segmentation par graphes travaille à partir de graphes complets, c’est à dire dans lesquels une arête existe entre chaque paire de nœuds. Ce type de graphe possède l’avantage de mettre en interaction l’ensemble des nœuds, mais constitue aussi le graphe de similarité le plus complexe qui puisse être construit.

qualité du résultat final. Il est donc important que l’algorithme de *clustering* considère correctement les caractéristiques des pixels.

1.3. GRAPHE DE SIMILARITÉ

Son exhaustivité rend cette technique inapplicable dès que le nombre de nœuds devient important, ce qui est souvent le cas pour les images.

En réalité, il est généralement inutile de recourir à un tel niveau de détails car deux pixels géométriquement proches ont plus de chances de partager des caractéristiques communes – et donc d’avoir une influence l’un sur l’autre – que deux pixels géométriquement opposés. Dans un graphe complet, la relation qui existe entre des pixels géométriquement distants sera généralement inhibée, car associée à un poids de faible valeur relative. Il est donc courant de ne pas mettre en relation deux nœuds si ceux-ci ne satisfont pas un certain critère de similarité. Cela amène à la notion de voisinage, et donc à la mesure de distance évoquée précédemment.

Les deux types de graphes de voisinage généralement utilisés sont le graphe de ε -voisinage et le graphe des k plus proches voisins [Felzenszwalb et Huttenlocher, 2004, Lezoray *et al.*, 2007, von Luxburg, 2007, Ta *et al.*, 2009].

Dans un graphe de ε -voisinage, deux nœuds ne sont mis en relation que si la distance les séparant ne dépasse pas un seuil préalablement fixé. Soit $G_\varepsilon = (V, E)$ un graphe de ε -voisinage, et $\mu : V \times V \rightarrow \mathbb{R}$ une mesure de distance. Le voisinage d’un nœud u est défini par :

$$\mathcal{N}_\varepsilon(u) = \{v \in V \setminus \{u\} \text{ tel que } \mu(u, v) \leq \varepsilon\} \quad . \quad (1.6)$$

Cette méthode permet notamment de reproduire les structures en 4 et 8-voisinage selon lesquelles les images sont traditionnellement traitées. En se basant sur les coordonnées des pixels et en fixant $\varepsilon = 1$, la distance de Manhattan permet d’obtenir un graphe grille en 4-voisinage et la distance de Chebychev, un graphe grille en 8-voisinage. Appliquées à des images 3D, ces mêmes définitions restent valables, et permettent de construire respectivement des 6 et 26-voisinages.

Dans un graphe des k plus proches voisins, le voisinage $\mathcal{N}_{kNN}(u)$ d’un nœud u est constitué des k nœuds les plus proches de u du point de vue de la mesure de distance μ utilisée. Cependant, telle qu’illustrée figure 1.1, cette heuristique produit un graphe orienté : un nœud v peut faire partie des k plus proches voisins d’un nœud u sans que u ne fasse partie du voisinage de v . Deux graphes non-orientés peuvent être dérivés à partir du graphe des k plus proches voisins [von Luxburg, 2007, Maier *et al.*, 2009] :

- le graphe mutuel des k plus proches voisins, dans lequel deux nœuds u et v sont connectés si $v \in N_{kNN}(u)$ **ou** si $u \in N_{kNN}(v)$. Chaque nœud possède alors au moins k voisins,
- le graphe symétrique des k plus proches voisins, dans lequel deux nœuds u et v sont connectés si $v \in N_{kNN}(u)$ **et** si $u \in N_{kNN}(v)$. Chaque nœud possède alors au plus k voisins.

La version mutuelle est généralement préférée car elle assure un certain degré de connectivité à chaque nœud [Elmoataz *et al.*, 2008, Ta, 2009].

Les graphes de ε -voisinage sont principalement utilisés lorsque les données font partie d’un domaine organisé – tel que les images – pour lequel la proximité géométrique reste un critère de similarité important. Les graphes des k plus proches voisins sont, quant à eux, plus régulièrement employés pour traiter des données issues de domaines non-organisés, pour lesquels il n’existe pas de notion de proximité géométrique.

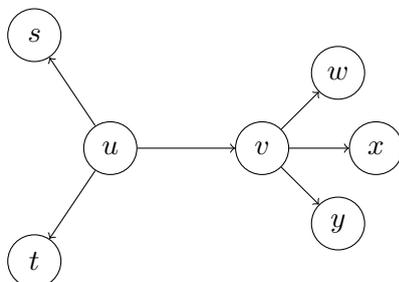


FIGURE 1.1 – Exemple de graphe orienté des 3 plus proches voisins (seuls les voisinages des nœuds u et v sont illustrés). La longueur des arêtes reflète la distance entre les nœuds. Le nœud v fait partie des 3 plus proches voisins du nœud u , mais le nœud u ne fait pas partie des 3 plus proches voisins du nœud v . Dans le graphe mutuel, le voisinage de u sera composé des nœuds s , t et v , tandis que celui de v sera composé de u , w , x et y . Dans le graphe symétrique, les nœuds u et v ne sont pas voisins.

Quelques travaux ont tenté d'évaluer l'influence du type de graphe, ou encore de la taille du voisinage. Dans [Felzenszwalb et Huttenlocher, 2004], le graphe des k plus proches voisins a été comparé au graphe de ε -voisinage. Un même algorithme de segmentation était alors exécuté dans ces deux configurations, et les performances comparées. Les résultats obtenus étaient relativement équivalents. Cependant, comme cela est confirmé dans [Grady et Jolly, 2008], l'utilisation de voisinages de taille trop réduite ne permet pas de tirer pleinement partie de l'information contenue dans les relations de voisinage. De fortes connectivités semblent donc être préférables, au détriment évident des performances de l'algorithme en matière de temps de calcul ainsi que d'occupation mémoire.

1.3.3 Sélection de la mesure de similarité

La similarité de deux nœuds est généralement inversement proportionnelle à la distance les séparant. À ce titre, il est tout à fait envisageable de pondérer la relation entre deux nœuds u et v à l'aide de la mesure de distance définie précédemment, en fixant par exemple $w_{uv} = (\mu(u, v) + \varepsilon)^{-1}$, avec ε un réel positif proche de zéro permettant d'éviter toute instabilité numérique.

Cependant, en particulier pour les domaines organisés tels que les images, une approche différente est généralement employée : la topologie du graphe est construite à l'aide d'heuristiques purement géométriques – permettant ainsi de limiter l'étendue des voisinages – tandis que la mesure de similarité est calculée à partir des données. Une mesure de distance, telle que celles présentées précédemment, est alors pondérée à l'aide d'une fonction de transfert décroissante, comme une fonction inverse :

$$w_{uv} = (\sigma \cdot \mu(u, v)^2 + \varepsilon)^{-1} , \quad (1.7)$$

ou encore une gaussienne :

$$w_{uv} = \exp\left(\frac{-\mu(u, v)^2}{\sigma}\right) , \quad (1.8)$$

avec $\sigma > 0$ un paramètre dont le rôle est de régler la vitesse de décroissance de ces fonctions.

Il a également été suggéré d'utiliser des mesures anisotropiques, tenant à la fois compte de la proximité géométrique et de la similarité des caractéristiques :

$$w_{uv} = \frac{1}{d(u, v)} \cdot \frac{1}{\sigma \cdot \mu(u, v)^2 + \varepsilon} \quad , \quad (1.9)$$

$$w_{uv} = \frac{1}{d(u, v)} \cdot \exp\left(\frac{-\mu(u, v)^2}{\sigma}\right) \quad , \quad (1.10)$$

avec $d(u, v)$ une mesure de la distance géométrique entre les deux nœuds u et v . Les tests réalisés dans [Grady et Jolly, 2008] indiquent que la mesure de similarité basée sur la fonction inverse permet une meilleure pondération des relations.

Enfin, il faut noter que certains algorithmes peuvent se passer de mesure de similarité, la topologie du graphe fournissant déjà une information suffisante pour mener à bien la tâche associée. Il est alors utile de fixer l'ensemble des mesures de similarité à une valeur neutre et constante, permettant un traitement équitable de l'ensemble des voisins. Dans ce cas, $\forall (u, v) \in E, w_{uv} = 1$.

Ces trois étapes permettent de définir une structure de données utilisée par la plupart des méthodes de traitement d'images à base de graphes. Dans la suite de ce chapitre, nous allons détailler quatre familles de méthodes de segmentation.

1.4 Méthodes à base d'arbres de couverture minimale

En théorie des graphes, un graphe est un arbre s'il est simple, connexe et acyclique, c'est à dire s'il n'existe qu'un unique chemin entre chaque paire de nœuds.

Soit $G = (V, E, w)$ un graphe. On suppose que G est connexe (il existe un chemin entre toute paire de nœud). L'arbre de couverture minimale³ de G correspond au graphe connexe $G' = (V, E', w)$ avec $E' \subset E$, tel que la somme des poids associés aux arêtes de l'ensemble E' est minimale. La figure 1.2 illustre cette notion.

Il faut noter que, à l'inverse de l'ensemble des algorithmes présentés dans ce chapitre, les algorithmes dont il est question dans cette section travaillent à partir de graphes de dissimilarité, dans lesquels la fonction de pondération mesure la « distance » entre les nœuds.

Dans un arbre, la suppression de k arêtes entraîne la création de $k + 1$ composantes connexe. Afin de réaliser le partitionnement d'un ensemble de données, l'idée générale est de supprimer les arêtes de l'arbre de couverture minimale présentant les poids les plus importants. Les composantes connexes résultante représentent alors le résultat de la segmentation. La figure 1.3 illustre ce principe.

3. Ou MST, de l'anglais *Minimum Spanning Tree*.

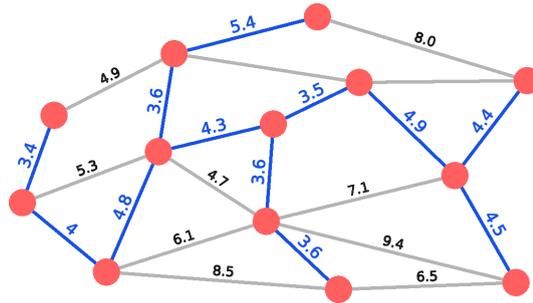


FIGURE 1.2 – Exemple d’arbre de couverture minimale. Le poids associé aux arêtes correspond à la distance Euclidienne entre chaque paire de nœuds. L’arbre de couverture minimale (en bleu) connecte l’ensemble des nœuds du graphe et minimise la somme des poids associés aux arêtes.

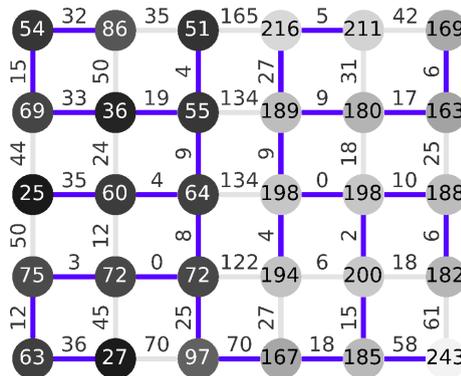


FIGURE 1.3 – Utilisation d’un arbre de couverture minimale pour la segmentation de données. La valeur associée à chaque nœud représente son niveau de gris, tandis que celle associée à chaque arête correspond à la différence de niveaux de gris (en valeur absolue) des nœuds qu’elle relie. Dans l’arbre de couverture minimale (représenté par les arêtes en bleu), l’arête de poids 70 (qui relie les deux régions au centre du graphe) est celle de poids maximal. Sa suppression entraîne la création de deux composantes connexes, chacune identifiant une région du graphe dans laquelle les niveaux de gris sont homogènes.

[Zahn, 1971] a été le premier à appliquer ce concept au partitionnement de données. Dans un arbre de couverture minimale, il définit comme « inconsistante » une arête dont le poids associé est supérieur au poids moyen des arêtes des composantes connexes situées à ses extrémités. Il propose ainsi un algorithme de partitionnement dans lequel les arêtes inconsistantes sont supprimées du MST.

Les travaux de Zahn ont inspiré à [Felzenszwalb et Huttenlocher, 2004] un algorithme de segmentation d'images non-supervisé dont le principe s'inspire du plus célèbre algorithme de construction d'arbres de couverture minimale, l'algorithme de Kruskal [Kruskal, 1956].

L'algorithme de Kruskal considère deux graphes : $G = (V, E, w)$, qui représente les données initiales et $G_{MST} = (V, E_s, w)$ l'arbre de couverture minimale. Notons que l'ensemble de nœuds ainsi que la fonction de pondération de ces deux graphes sont identiques, et que $E_s \subset E$. À l'initialisation, $E_s = \emptyset$. Chaque arête de E est ensuite considérée par ordre croissant de poids. Si les composantes connexes de G situées aux deux extrémités d'une arête sont différentes, alors celle-ci est ajoutée au graphe G_{MST} . L'algorithme s'arrête lorsque toutes les arêtes ont été considérées.

Une caractéristique importante de cet algorithme, qu'exploite Felzenszwalb et Huttenlocher, est que l'arbre de couverture minimale n'est pas construit à partir d'une unique entité, mais résulte de la fusion d'un ensemble de composantes connexes⁴. Étant donné que les arêtes sont considérées par ordre croissant de poids, chaque arête qui est ajoutée à l'arbre de couverture minimale joint deux nœuds qui sont plus « distants » que l'ensemble des paires de nœuds jusqu'alors considérées.

Felzenszwalb et Huttenlocher proposent donc de mesurer l'homogénéité d'une composante connexe à partir de sa « différence interne », définie comme égale au poids le plus important de son arbre de couverture minimale, et de conditionner la fusion de deux composantes à l'aide de cette mesure. Deux composantes ne sont ainsi fusionnées que si cela n'entraîne pas la création d'une composante moins homogène, c'est à dire si il n'existe aucune arête entre ces deux composantes dont le poids serait supérieur à la différence interne de chacune des composantes.

Une version hiérarchique de cette approche est également proposée dans [Haxhimusa et Kropatsch, 2004].

1.5 Méthodes à base de coupes minimales et approches spectrales

Une coupe d'un graphe $G = (V, E)$ est définie comme étant la partition de celui-ci en deux sous-graphes disjoints S et T . Le coût – ou valeur – de cette coupe correspond au nombre d'arêtes joignant ces deux sous-graphes :

$$\text{cut}(S, T) = \text{Card}\{(u, v) \in E \text{ tel que } u \in S \text{ et } v \in T\} \quad . \quad (1.11)$$

Lorsqu'un graphe $G = (V, E, w)$ est pondéré, ce coût est alors égal à la somme des

4. À l'initialisation, chaque nœud définit une composante connexe.

poids des arêtes :

$$\text{cut}(S, T) = \sum_{\substack{(u,v) \in E \\ u \in S \\ v \in T}} w_{uv} \quad . \quad (1.12)$$

À partir de cette notion, il est possible de définir le problème de calcul de coupe minimale : étant donné un graphe $G = (V, E)$ et deux nœuds $s, t \in V$, trouver les régions S et T telles que $s \in S$, $t \in T$ et $S \cap T = \emptyset$ dont la coupe associée est de valeur minimale (la définition de ce problème est identique pour les graphes pondérés).

Lorsqu'il est associé à la notion de graphe de similarité, il est aisé de transformer ce problème de calcul de coupe minimale en un problème de segmentation d'images : étant donné le graphe de similarité associé à une image et deux pixels représentatifs de deux régions à segmenter, calculer la coupe minimale séparant ces deux pixels permettra d'identifier les arêtes de plus faible poids, c'est à dire de « couper » entre les pixels présentant un minimum de similarité.

Cette approche a notamment été appliquée par Wu et Leahy [Wu et Leahy, 1993] qui emploient l'algorithme de Gomory-Hu [Gomory et Hu, 1961] afin de construire un arbre identifiant l'ensemble des coupes séparant toute paire de nœuds du graphe. La segmentation de l'image en K classes est alors réalisée en identifiant les $K - 1$ coupes de plus petites valeurs.

Malheureusement, l'algorithme de Gomory-Hu présente une complexité relativement importante ($\mathcal{O}(|V|^4)$, avec $|V|$ le nombre de nœuds du graphe), ce qui le rend inapplicable à de grands graphes. Partant du constat que la majorité des coupes identifiées n'est pas utilisée – car de valeur trop importante – Wu et Leahy ont proposé une implémentation hiérarchique de l'algorithme initial. Celle-ci vise à identifier des sous-graphes ne pouvant pas influencer sur le calcul des coupes minimales afin de les condenser⁵. La taille du graphe sur lequel l'algorithme de Gomory-Hu est ensuite exécuté peut ainsi être réduite, permettant le traitement de graphes de grandes tailles. Les expérimentations alors menées ont permis de constater une amélioration notable du temps de calcul, celui-ci étant alors divisé par 72.

Cette approche est cependant biaisée. La valeur d'une coupe étant globalement proportionnelle au nombre d'arêtes « coupées », celles de très petites tailles sont favorisées, sans qu'elles n'aient de réelle signification du point de vue de la segmentation.

Afin de compenser ce biais, plusieurs travaux ont suggéré une mesure « normalisée » du coût d'une coupe, dans laquelle la taille de cette dernière est prise en compte.

Dans [Wang et Siskind, 2001], la notion de coupe moyenne est ainsi proposée. La normalisation du coût d'une coupe est réalisée en divisant le coût initial, tel que formulé par l'équation (1.12), par le nombre d'arêtes concernées. L'algorithme de calcul de coupe moyenne minimale – emprunté lui aussi à la théorie des graphes – se limite cependant à des graphes planaires⁶ pour une résolution en temps polynomial, le problème étant NP-difficile pour des graphes de topologies arbitraires.

La méthode des coupes normalisées [Shi et Malik, 2000] est, quant à elle, radicalement

5. La condensation d'un sous-graphe revient à le représenter par un nœud unique.

6. Un graphe est qualifié de planaire s'il peut être représenté dans le plan sans qu'aucune arête n'en croise une autre.

1.5. MÉTHODES À BASE DE COUPES MINIMALES ET APPROCHES SPECTRALES

différente. Le coût d'une coupe y est exprimé comme une proportion du poids des arêtes concernées par la coupe :

$$\text{Ncut}(S, T) = \frac{\text{cut}(S, T)}{\text{cut}(S, V)} + \frac{\text{cut}(S, T)}{\text{cut}(T, V)} \quad , \quad (1.13)$$

avec V l'ensemble des nœuds. L'originalité de cette approche réside dans la méthode de résolution du problème de calcul de coupe minimale. Celle-ci fait en effet partie du domaine de la théorie spectrale des graphes, c'est à dire qu'elle se base sur une analyse des valeurs et vecteurs propres de la matrice Laplacienne du graphe, laquelle est définie par :

$$M_{uv} = \begin{cases} \sum_{s \in V} w_{us} & \text{si } u = v \\ -w_{uv} & \text{sinon} \end{cases} \quad \forall u, v \in V \quad . \quad (1.14)$$

La propriété qui nous intéresse dans le cadre de la segmentation d'images est que le spectre du graphe, c'est à dire la liste des valeurs propres de sa matrice Laplacienne normalisée, est indicatif du nombre de composantes connexes du graphe. En particulier, si la matrice admet i valeurs propres nulles, cela signifie que le graphe est composé de $i + 1$ composantes connexes, que les vecteurs propres associés permettent de caractériser. Cette propriété est étendue aux valeurs propres non-nulles : l'intensité d'une valeur propre est inversement proportionnelle à la connexité de la composante associée avec le reste du graphe. Ainsi, en considérant les i plus petites valeurs propres, il est possible d'identifier $i + 1$ clusters de nœuds présentant une forte connexité interne et une faible connexité avec le reste du graphe.

Cette propriété peut être combinée à la formule du coût d'une coupe normalisée $\text{Ncut}(S, T)$ sous la forme d'un problème aux valeurs propres généralisé. L'identification des $k - 1$ plus petites valeurs propres issues de la solution du problème permet alors de segmenter l'image en k régions. La relaxation de certaines contraintes reste nécessaire afin de permettre la résolution du problème, celui-ci étant sinon NP-complet.

D'un point de vue purement algorithmique, la plus grosse difficulté de cette méthode revient à résoudre le problème aux valeurs propres généralisé, ce qui requiert actuellement $\mathcal{O}(n^3)$ opérations, avec n le nombre de nœuds du graphe. Il est évident que cette approche est impraticable, y compris pour des images de petite taille, ne serait-ce que pour stocker en mémoire la matrice Laplacienne. Heureusement, plusieurs propriétés permettent d'optimiser la résolution de cet algorithme. D'une part, en fonction de la méthode de construction du graphe, il se peut que la matrice d'adjacence (et donc la matrice Laplacienne) soit essentiellement composée de valeurs nulles⁷. Cette propriété peut être exploitée par de nombreuses implémentations de calcul matriciel afin d'optimiser l'espace mémoire occupé par ces matrices, mais aussi en réduisant le nombre d'opérations liées à la résolution du problème aux valeurs propres généralisé. D'autre part, le résultat de la segmentation n'est généré qu'à partir d'un nombre réduit de vecteurs propres, pour lesquels il n'est pas nécessaire d'avoir une grande précision. Ces deux dernières propriétés peuvent être exploitées par l'algorithme Lanczos [Golub et Van Loan, 1996], qui permet de réduire

7. On parle alors de matrice creuse.

la complexité du calcul des vecteurs propres en $\mathcal{O}(n^{3/2})$, ce qui rend cette approche exploitable pour des images de taille modérée. Il reste cependant impossible de l'appliquer à des images de grande taille ou encore à des images 3D.

Malheureusement, la relaxation et les différentes approximations nécessaires à la résolution du problème rendent le résultat de la segmentation souvent difficile à interpréter ou à exploiter, comme illustré figure 1.4. Il est actuellement compliqué d'influer sur l'algorithme pour corriger ce genre de comportement car le seul élément sur lequel l'utilisateur a la possibilité intervenir est le graphe lui-même.

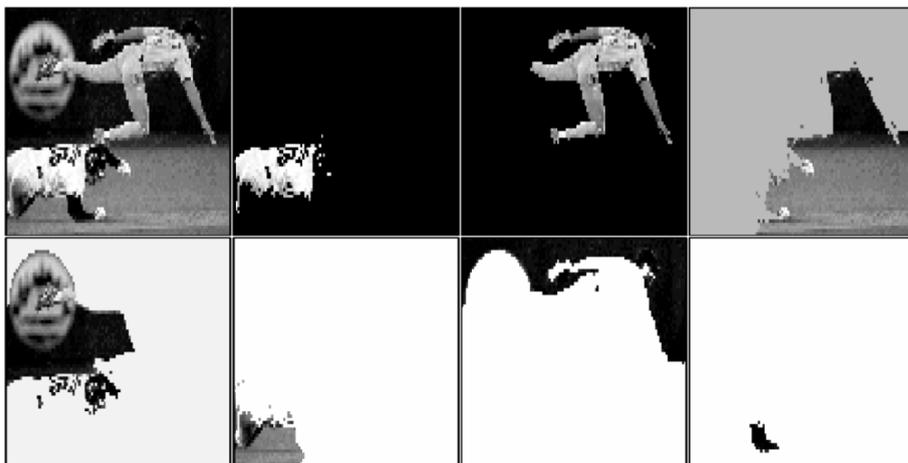


FIGURE 1.4 – Exemple de segmentation obtenue par la méthode des coupes normalisées minimales : l'image d'origine (en haut à gauche) et les partitions générées à partir des sept plus petites valeurs propres solutions du problème.

D'autres méthodes exploitant différentes matrices Laplaciennes ont également été explorées [Weiss, 1999, von Luxburg, 2007].

1.6 Marche aléatoire

La marche aléatoire correspond à la modélisation mathématique d'un processus dynamique discret – le « marcheur » – dont chaque évolution est définie à l'aide d'un processus stochastique de type chaîne de Markov : à chaque étape, le marcheur passe aléatoirement d'une position (ou état) à une autre selon une certaine probabilité qui ne dépend pas des positions passées mais uniquement de la position actuelle.

L'analyse du comportement d'un tel processus permet de mettre en évidence des relations entre différentes positions ou états auxquels le processus peut accéder. Un célèbre exemple d'application reposant sur ce principe est le robot d'indexation du moteur de recherche de Google, lequel a notamment pour tâche d'évaluer la popularité d'une page Web compte tenu de la popularité de celles pointant dessus.

Dans [Grady, 2006], l'auteur propose de traiter le problème de k -partition d'une image à l'aide d'un processus de marche aléatoire. Étant donné un graphe de similarité et un ensemble de marqueurs manuellement positionnés, le problème revient à identifier sur quel

marqueur un marcheur aléatoire au départ d'un pixel a le plus de chance d'arriver en premier, la mesure de similarité du graphe étant interprétée comme une probabilité de déplacement.

La résolution de ce problème par la simulation d'un marcheur aléatoire est bien entendu inenvisageable. Cependant, il a été démontré que la probabilité qu'un marcheur aléatoire atteigne un marqueur correspond à la solution d'un problème de Dirichlet [Grady, 2006], qui consiste à calculer la fonction harmonique – c'est à dire dont la dérivée seconde est nulle en tout point – prolongeant les marqueurs. Ce problème peut être résolu à l'aide d'un système d'équations basé sur la matrice Laplacienne du graphe. Bien que la résolution d'un tel système d'équation soit relativement lourde et complexe, des optimisations tenant compte de la densité des matrices peuvent être appliquées.

1.7 Segmentation par régularisation de graphes

La régularisation d'un problème consiste à introduire dans sa formulation un ensemble de contraintes dans le but de le rendre soluble ou de prévenir l'obtention de solutions non-exploitable ou trop complexes. Cette technique est notamment utilisée dans le domaine de l'apprentissage automatique pour éviter le sur-apprentissage [Larsen et Hansen, 1994]. En traitement d'images, les contraintes utilisées concernent essentiellement la régularité de la solution, et permettent d'éviter une trop grande sensibilité au bruit.

Dans ses travaux, Ta propose d'aborder plusieurs problèmes de traitement d'images (colorisation, débruitage, segmentation) à l'aide d'une méthode de régularisation de graphes [Ta, 2009]. Pour cela, il s'appuie sur deux classes de modèles variationnels discrets qui permettent de transformer le problème initial en un problème de minimisation d'énergie. Celui-ci prend la forme suivante :

$$\min_{f \in \mathcal{H}(V)} \{E(f) = E_r(f) + \lambda \cdot E_d(f)\} \quad , \quad (1.15)$$

avec $E_d(f)$ le terme d'attache aux données lié à la pertinence de la solution f par rapport à l'objectif initial, $E_r(f)$ la fonctionnelle de régularisation qui vise à évaluer la régularité de la solution et λ le paramètre permettant d'exprimer un compromis entre la précision et la régularité de la solution.

Avant d'entrer dans le détail de ces modèles variationnels et d'en présenter les applications, il est nécessaire de définir la façon dont la fonctionnelle de régularisation évalue la régularité d'une fonction $f \in \mathcal{H}(V)$. Comme dans le domaine continu, la régularité d'une fonction peut être évaluée à l'aide d'opérateurs différentiels définis sur les graphes.

1.7.1 Opérateurs différentiels définis sur les graphes

Les travaux de Ta s'inspirent de méthodes variationnelles définies dans le domaine continu et basées sur les équations aux dérivées partielles. Ils reposent donc sur un ensemble d'opérateurs différentiels qui correspondent à des versions adaptées pour les graphes pondérés des opérateurs traditionnels d'analyse différentielle [Bougleux *et al.*, 2007, Elmoataz *et al.*, 2008].

Afin de présenter ces opérateurs, nous considérons un graphe pondéré $G = (V, E, w)$, auquel sont associées les fonctions scalaires à valeurs réelles $f \in \mathcal{H}(V)$ et $F \in \mathcal{H}(E)$.

1.7.1.1 Opérateur de différence pondérée et opérateur adjoint

Les deux opérateurs de base sont l'opérateur de différence pondérée et son adjoint [Bougleux *et al.*, 2007, Elmoataz *et al.*, 2008].

L'opérateur de différence pondérée correspond à l'opérateur de dérivée du domaine continu. Il s'exprime sous la forme d'une application $d_w : \mathcal{H}(V) \rightarrow \mathcal{H}(E)$ définie selon chaque arête uv par :

$$(d_w f)(u, v) = \frac{f(v) - f(u)}{h_{uv}} \quad , \quad (1.16)$$

avec h_{uv} le pas d'échantillonnage permettant de prendre en compte la « distance » entre u et v . Dans le cas des graphes, ce pas permet de faire intervenir la mesure de similarité. Il est ainsi relativement courant de prendre $h_{uv} = 1/\sqrt{w_{uv}}$ [Bougleux *et al.*, 2007, Elmoataz *et al.*, 2008, Ta, 2009]. L'opérateur de différence pondérée s'exprime alors par la formule suivante⁸ :

$$(d_w f)(u, v) = \sqrt{w_{uv}}(f(v) - f(u)) \quad . \quad (1.17)$$

L'opérateur de différence pondérée est lié à son adjoint par la relation suivante :

$$\langle d_w f, F \rangle = \langle f, d_w^* F \rangle \quad , \quad (1.18)$$

avec $\langle \cdot, \cdot \rangle$ l'opérateur produit scalaire (voir equations (1.1) et (1.2)).

L'opérateur adjoint de l'opérateur de différence pondérée $d_w^* : \mathcal{H}(E) \rightarrow \mathcal{H}(V)$ est ainsi défini pour chaque nœud u par :

$$(d_w^* F)(u) = \sum_{v \sim u} \sqrt{w_{uv}}(F(v, u) - F(u, v)) \quad . \quad (1.19)$$

Il est alors possible de définir l'opérateur de divergence $-d_w^*$, lequel mesure le flot sortant en chaque nœud $u \in V$ d'une fonction $F \in \mathcal{H}(E)$.

1.7.1.2 Opérateur gradient pondéré

L'opérateur gradient pondéré $\nabla_w f : \mathcal{H}(V) \rightarrow \mathbb{R}^m$ permet d'évaluer la régularité d'une fonction $f \in \mathcal{H}(V)$. Il est défini en chaque nœud $u \in V$ comme étant le vecteur des différences pondérées pour l'ensemble des arêtes incidentes à u :

$$(\nabla_w f)(u) = ((d_w f)(u, v))_{v \sim u}^T \quad . \quad (1.20)$$

Par la suite, c'est essentiellement la norme L_q de ce gradient qui sera utilisée :

$$\|(\nabla_w f)(u)\|_q = \left[\sum_{v \sim u} (\sqrt{w_{uv}} |f(v) - f(u)|)^q \right]^{1/q} \quad . \quad (1.21)$$

8. La notation de l'équation (1.16) est utilisée par analogie avec la définition de l'opérateur traditionnel de dérivée, dont l'opérateur de différence pondérée est le pendant lorsque appliqué à des graphes.

1.7.1.3 Opérateur p-Laplacien pondéré isotrope et anisotrope

L'opérateur p-Laplacien pondéré est un opérateur de second ordre dont le rôle est de mesurer la régularité des variations d'une fonction $f \in \mathcal{H}(V)$. Il est défini pour tout réel p tel que $0 < p < +\infty$, paramètre dont le rôle est de pénaliser les fortes variations de f .

Dans sa version isotrope, le p-Laplacien pondéré $\Delta_{w,p}^i : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ est défini par :

$$\Delta_{w,p}^i f = d_w^* \left(\|\nabla_w f\|_2^{p-2} d_w f \right) . \quad (1.22)$$

Le p-Laplacien pondéré anisotrope $\Delta_{w,p}^a : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ est quant à lui défini par :

$$\Delta_{w,p}^a f = d_w^* \left(|d_w f|^{p-2} d_w f \right) . \quad (1.23)$$

En un nœud $u \in V$, ces opérateurs s'expriment par :

$$\begin{aligned} (\Delta_{w,p}^i f)(u) &= \sum_{v \sim u} \sqrt{w_{uv}} \left(\frac{(d_w f)(vu)}{\|\nabla_w f(v)\|_2^{2-p}} - \frac{(d_w f)(uv)}{\|\nabla_w f(u)\|_2^{2-p}} \right) , \\ &= \sum_{v \sim u} w_{uv} \left(\frac{f(u) - f(v)}{\|\nabla_w f(v)\|_2^{2-p}} - \frac{f(v) - f(u)}{\|\nabla_w f(u)\|_2^{2-p}} \right) , \\ &= \sum_{v \sim u} w_{uv} \left(\|\nabla_w f(v)\|_2^{p-2} + \|\nabla_w f(u)\|_2^{p-2} \right) (f(u) - f(v)) , \end{aligned} \quad (1.24)$$

et :

$$\begin{aligned} (\Delta_{w,p}^a f)(u) &= \sum_{v \sim u} \sqrt{w_{uv}} \left(\frac{(d_w f)(vu)}{|(d_w f)(vu)|^{2-p}} - \frac{(d_w f)(uv)}{|(d_w f)(uv)|^{2-p}} \right) , \\ &= \sum_{v \sim u} w_{uv}^{p/2} \left(\frac{f(u) - f(v)}{|f(u) - f(v)|^{2-p}} - \frac{f(v) - f(u)}{|f(v) - f(u)|^{2-p}} \right) , \\ &= 2 \sum_{v \sim u} w_{uv}^{p/2} |f(u) - f(v)|^{p-2} (f(u) - f(v)) . \end{aligned} \quad (1.25)$$

1.7.2 Fonctionnelles de régularisation

Comme évoqué plus tôt, le rôle d'une fonctionnelle de régularisation est de mesurer la régularité d'une fonction. Dans ses travaux, Ta reprend la famille de fonctionnelles isotropes proposée dans [Bougleux et Elmoataz, 2005, Elmoataz *et al.*, 2008]. Celles-ci sont définies sur les graphes et utilisent la norme L_2 du gradient pondéré (1.21) afin de proposer une mesure inspirée de la variation totale⁹.

La variation totale correspond à l'amplitude cumulée de l'ensemble des variations d'une fonction à valeur réelles $g : \mathbb{R}^m \rightarrow \mathbb{R}$. Elle est définie sur un intervalle borné $\Omega \in \mathbb{R}^m$ par :

$$TV_\Omega(g) = \int_\Omega |g'(x)| dx . \quad (1.26)$$

9. Le principe de variation totale sera par la suite abrégé par TV, de l'anglais *Total Variation*.

Elle peut être adaptée aux graphes de similarité en considérant la norme du gradient en chaque nœud. Soit $G = (V, E, w)$ un graphe pondéré et $f \in \mathcal{H}(V)$ une fonction à valeur réelles définie sur l'ensemble des nœuds. Cette famille de fonctionnelles s'exprime alors pour tout $0 < p < +\infty$ par :

$$\begin{aligned} \mathcal{R}_{w,p}^i &= \sum_{u \in V} \|(\nabla_w f)(u)\|_2^p \quad , \\ &= \sum_{u \in V} \left[\sum_{v \sim u} w_{uv} (f(v) - f(u))^2 \right]^{p/2} \quad . \end{aligned} \quad (1.27)$$

[Ta, 2009] propose une nouvelle famille de fonctionnelles anisotropes de régularisation, basée sur la norme L_q du gradient pondéré. Elle s'exprime pour $0 < p < +\infty$ et $q = p$ par :

$$\begin{aligned} \mathcal{R}_{w,p}^a &= \sum_{u \in V} \|(\nabla_w f)(u)\|_p^p \quad , \\ &= \sum_{u \in V} \sum_{v \sim u} w_{uv}^{p/2} |f(v) - f(u)|^p \quad . \end{aligned} \quad (1.28)$$

Ces deux familles de fonctionnelles peuvent s'exprimer de manière générale par :

$$\mathcal{R}_{w,q,\phi}^* = \sum_{u \in V} \phi(\|(\nabla_w f)(u)\|_q) \quad , \quad (1.29)$$

avec $\phi(\cdot)$ le noyau de régularisation permettant de pénaliser les fortes variations de la fonction f au voisinage d'un nœud.

1.7.3 Première classe de modèles variationnels

À partir des éléments présentés précédemment, une première classe de modèles variationnels peut être définie [Ta *et al.*, 2007a, Ta *et al.*, 2007b]. Le critère de régularisation utilisé est inspiré de la variation totale. Cette méthode est notamment appliquée à plusieurs problèmes d'interpolation. Pour rappel, en reconnaissance de formes, l'interpolation consiste à attribuer à chacune des données une valeur à partir d'un ensemble de marqueurs initiaux. Un exemple typique de problème d'interpolation – que cet algorithme est capable de traiter – est la colorisation d'images, où le but est d'attribuer un code couleur à chacun des pixels à partir de marqueurs colorimétriques [Lezoray *et al.*, 2008, Ta, 2009].

1.7.3.1 Formulation

On considère un graphe de similarité $G = (V, E, w)$ associé à l'image à traiter, et $f_0 \in \mathcal{H}(V)$ la fonction initiale représentative des marqueurs initiaux.

Cette première classe de modèles s'exprime par :

$$\min_{f: V \rightarrow \mathbb{R}} \left\{ \mathcal{E}_1(f, f_0, \lambda) = \mathcal{R}_{w,q,\phi}^* + \sum_{u \in V} \frac{\lambda}{2} \|f(u) - f_0(u)\|_2^2 \right\} \quad , \quad (1.30)$$

avec $f \in \mathcal{H}(V)$ la solution, résultat de l'interpolation.

Suivant les valeurs prises par le paramètre q de la fonctionnelle de régularisation ainsi que l'expression du noyau de régularisation $\phi(\cdot)$ utilisé, deux modèles de régularisation peuvent être générés à partir de cette classe. En particulier, en utilisant $\phi(s) = \frac{1}{p}s^p$ et $q = 2$, le modèle de régularisation obtenu est appelé p-TV isotrope, tandis qu'avec $\phi(s) = \frac{1}{2p}s^p$ et $p = q$, le modèle est p-TV anisotrope.

Le problème de minimisation (1.30) associé à cette classe de modèles est convexe si $p \geq 1$. Il admet donc un minimum global. La dérivée partielle de l'énergie $\mathcal{E}_1(f, f_0, \lambda)$ par rapport à f est alors nulle en tout sommet $u \in V$:

$$\frac{\partial \mathcal{E}_1(f, f^0, \lambda)}{\partial f(u)} = 0 \quad , \quad (1.31)$$

équation que l'on peut réécrire :

$$\frac{\partial \mathcal{E}_1(f, f^0, \lambda)}{\partial f(u)} = \sum_{v \sim u} \alpha_{uv}^{q, \phi, f} (f(u) - f(v)) \cdot |f(u) - f(v)|^{q-2} + \lambda(f(u) - f^0(u)) = 0 \quad , \quad (1.32)$$

avec :

$$\alpha_{uv}^{q, \phi, f} = w_{uv}^{q/2} \left(\frac{\phi'(\|(\nabla_w f)(u)\|_q)}{\|(\nabla_w f)(u)\|_q^{q-1}} + \frac{\phi'(\|(\nabla_w f)(v)\|_q)}{\|(\nabla_w f)(v)\|_q^{q-1}} \right) \quad . \quad (1.33)$$

Pour des raisons de stabilité numérique, il convient d'utiliser des versions régularisées de la norme du gradient et de la valeur absolue. Les opérateurs utilisés par la suite sont donc :

$$\|(\nabla_w f)(u)\|_q = \sqrt{\|(\nabla_w f)(u)\|_q + \varepsilon} \quad , \quad (1.34)$$

$$|f(u) - f(v)| = |f(u) - f(v)| + \varepsilon \quad , \quad (1.35)$$

avec ε une valeur positive proche de zéro, mais la notation initiale est conservée par mesure de clarté.

La solution du problème de minimisation peut être approximée par un algorithme itératif¹⁰. En réécrivant l'équation (1.32) sous la forme suivante :

$$f(u) = \frac{\lambda f_0(u) + \sum_{v \sim u} \alpha_{uv}^{q, \phi, f} |f(u) - f(v)|^{q-2} f(v)}{\lambda + \sum_{v \sim u} \alpha_{uv}^{q, \phi, f} |f(u) - f(v)|^{q-2}} \quad \forall u \in V \quad , \quad (1.36)$$

il est possible d'appliquer l'algorithme itératif de Gauss-Jacobi :

$$\begin{cases} f^0(u) = f_0(u) \\ f^{n+1}(u) = \frac{\lambda f_0(u) + \sum_{v \sim u} \alpha_{uv}^{q, \phi, f^n} |f^n(u) - f^n(v)|^{q-2} f(v)}{\lambda + \sum_{v \sim u} \alpha_{uv}^{q, \phi, f^n} |f^n(u) - f^n(v)|^{q-2}} \quad \forall u \in V. \end{cases} \quad (1.37)$$

La condition d'arrêt de l'algorithme peut être un nombre prédéfini d'itérations, ou encore l'atteinte d'un certain niveau de convergence, avec $\|f^{n+1} - f^n\| < \tau$.

10. D'autres méthodes sont également applicables. Nous pouvons notamment citer les approches à base de coupes de graphes [Kolmogorov et Zabih, 2004] ou encore celles basées sur une expression duale du problème de minimisation [Hidane *et al.*, 2012, Couprie *et al.*, 2013].

1.7.3.2 Application à la segmentation semi-supervisée

Ta propose de traiter des problèmes de classification et de segmentation de données semi-supervisées à partir de cet algorithme d'interpolation. L'objectif est alors d'attribuer à chacune des données non plus une valeur numérique mais une classe (ou un label) à partir de marqueurs positionnés sur un ensemble d'éléments représentatifs de chacune des classes à identifier.

On suppose que l'image est constituée de c régions à identifier, nombre connu *a priori*. Soit C_i , $i = 1 \dots c$, l'ensemble des nœuds initialement marqués comme appartenant à la classe i , et $V_0 = \cup C_i$ l'ensemble des nœuds disposant d'un marqueur initial. La fonction initiale $f_0 \in \mathcal{H}^c(V)$ associe à chaque nœud un vecteur composé de c valeurs réelles : $f_0 = (f_{0,i})_{i=1\dots c}$, avec :

$$f_{0,i}(u) = \begin{cases} 1 & \text{si } u \in C_i \\ 0 & \text{sinon} \end{cases} . \quad (1.38)$$

Afin de permettre la reconnaissance de c classes, c processus de régularisation de graphe basés sur l'équation (1.37) sont exécutés en parallèle, chacun cherchant à calculer une fonction $f_i : V \rightarrow [0, 1]$ solution du problème de minimisation (1.30) étant donné $f_{0,i}$.

Nous obtenons ainsi pour le modèle p-TV isotrope :

$$\begin{cases} f_i^0(u) = f_{0,i}(u) \\ f_i^{n+1}(u) = \frac{\lambda f_{i,0}(u) + \sum_{v \sim u} w_{uv} \left(\|\nabla_w f_i^n(u)\|_2^{p-2} + \|\nabla_w f_i^n(v)\|_2^{p-2} \right) f_i^n(v)}{\lambda + \sum_{v \sim u} w_{uv} \left(\|\nabla_w f_i^n(u)\|_2^{p-2} + \|\nabla_w f_i^n(v)\|_2^{p-2} \right)} \quad \forall u \in V, \end{cases} \quad (1.39)$$

et :

$$\begin{cases} f_i^0(u) = f_{0,i}(u) \\ f_i^{n+1}(u) = \frac{\lambda f_{i,0}(u) + \sum_{v \sim u} w_{uv}^{p/2} |f_i^n(u) - f_i^n(v)|^{p-2} f_i^n(v)}{\lambda + \sum_{v \sim u} w_{uv}^{p/2} |f_i^n(u) - f_i^n(v)|^{p-2}} \quad \forall u \in V, \end{cases} \quad (1.40)$$

pour le modèle p-TV anisotrope.

Enfin, le résultat final de la segmentation est calculé par vote majoritaire, les valeurs représentées par les différents f_i pouvant être interprétées comme un potentiel d'appartenance de chaque nœud à la classe i .

1.7.4 Seconde classe de modèles variationnels

Dans [Ta, 2009], l'auteur propose une seconde classe de modèles variationnels. Elle se distingue de la précédente de deux façons. D'une part, il s'agit d'une classe dédiée à la segmentation de données. La solution ainsi recherchée est une fonction indicatrice de la segmentation qui est donc binaire. D'autre part, le critère d'attache aux données n'est plus limité à la notion de variation totale. Ta propose ainsi d'utiliser deux implémentations différentes de cette classe de modèles variationnels, basées respectivement sur les modèles

de Rudin, Osher et Fatemi¹¹ [Rudin *et al.*, 1992] et de Chan et Vese¹² [Chan et Vese, 2001], implémentations que nous détaillerons par la suite. La forme générale de cette classe de problèmes s'exprime par :

$$\min_{f:V \rightarrow \{0,1\}} \left\{ \mathcal{E}_2(f, f_0, \lambda) = \mathcal{R}_{w,q,\phi}^* + \sum_{u \in V} \lambda g(f_0)(u) f(u) \right\} , \quad (1.41)$$

avec $g(\cdot)$ la fonction jouant le rôle de terme d'attache aux données.

Afin de pouvoir aboutir à un algorithme itératif similaire à celui utilisé pour résoudre la première classe de problème, Ta propose une version relaxée du problème de minimisation (1.41), dans laquelle les solutions ne sont plus binaires, mais définies dans un domaine continu, à savoir $[0, 1]$:

$$\min_{f:V \rightarrow [0,1]} \left\{ \tilde{\mathcal{E}}_2(f, f_0, \lambda) = \mathcal{R}_{w,q,\phi}^* + \sum_{u \in V} \lambda g(f_0)(u) f(u) \right\} . \quad (1.42)$$

Ta démontre, à l'aide de la formule de la co-aire [Chambolle, 2005] que, dans le cas où $\phi(s) = \frac{1}{p}s^p$ et $p = q = 1$, le problème de minimisation (1.42) est convexe et que la solution du problème relaxé est aussi solution du problème initial (1.41). Il est ainsi possible d'approximer une solution du problème de segmentation à l'aide du même algorithme itératif utilisé pour la première classe, basé sur l'annulation du gradient :

$$\begin{cases} f^0(u) = m(u) \\ f^{n+1}(u) = \frac{\sum_{v \sim u} \sqrt{w_{uv}} |f^n(u) - f^n(v)|^{-1} f^n(v) - \lambda g(f_0)(u)}{\sum_{v \sim u} \sqrt{w_{uv}} |f^n(u) - f^n(v)|^{-1} f^n(v)} \quad \forall u \in V \end{cases} , \quad (1.43)$$

avec $m : V \rightarrow \{0, 1\}$ le marqueur initial, une segmentation grossière que l'algorithme a pour rôle d'affiner au fil des itérations.

Du fait de la relaxation du problème, il est cependant nécessaire de projeter la solution calculée après chaque itération dans l'intervalle $[0, 1]$ tel que :

$$f^n(u) = \begin{cases} 0 & \text{si } f^n(u) < 0 \\ 1 & \text{si } f^n(u) > 1 \\ f^n(u) & \text{sinon} \end{cases} . \quad (1.44)$$

1.7.4.1 Modèle de Rudin, Osher et Fatemi

Le modèle de Rudin, Osher et Fatemi s'inspire d'une méthode de débruitage d'images qui se base sur une mesure de la variation totale.

Cette mesure part du principe qu'un pixel de bruit présente un niveau de gris très différent de ses voisins. Elle est ainsi définie comme la somme des gradients d'intensité en chaque point de l'image. Sa minimisation entraîne une régularisation de la fonction qui associe à chaque pixel son niveau de gris, et donc un débruitage de l'image.

11. Ce modèle sera pas la suite dénommé « modèle de ROF ».

12. Ce modèle sera pas la suite dénommé « modèle de CV ».

Initialement proposée dans [Rudin *et al.*, 1992], cette méthode s'exprime dans le domaine continu par :

$$\mathcal{J}_{ROF}(f, f_0, \lambda) = \int_{\Omega} \|\nabla f\|_1 + \lambda \int_{\Omega} (f(x) - f_0(x))^2 dx \quad , \quad (1.45)$$

avec $f_0 : \Omega \subset \mathbb{R}^m \rightarrow [0, 1]$ la fonction initiale représentant l'image, et f la solution recherchée.

Afin de transformer cette fonctionnelle d'énergie en un problème de minimisation appartenant à la seconde classe de modèles variationnels, il est nécessaire de procéder à une reformulation du terme d'attache aux données. En se restreignant à une fonction f binaire, celui-ci peut se réécrire :

$$(f - f_0)^2 = f^2 - f f_0 + f_0^2 = f - f f_0 + f_0^2 = (1 - f_0)f + f_0^2 \quad , \quad (1.46)$$

ce qui permet de reformuler la fonctionnelle de la façon suivante :

$$\mathcal{J}_{ROF}(f, f_0, \lambda) = \int_{\Omega} \|\nabla f\|_1 + \lambda \int_{\Omega} \left((1 - f_0(x))f(x) + f_0(x)^2 \right) \quad . \quad (1.47)$$

Étant donné que le terme $f_0(x)^2$ est constant, la solution f minimisant la fonctionnelle d'énergie 1.47 est aussi la solution du problème de minimisation de la fonctionnelle suivante :

$$\mathcal{J}_{ROF}(f, f_0, \lambda) = \int_{\Omega} \|\nabla f\|_1 + \lambda \int_{\Omega} (1 - f_0(x))f(x) dx \quad . \quad (1.48)$$

Ce modèle étant, de par sa formulation, un cas particulier du problème (1.41), Ta a proposé d'étendre la formulation de ce modèle aux graphes :

$$\min_{f:V \rightarrow \{0,1\}} \left\{ \mathcal{E}_{ROF}(f, f_0, \lambda) = \mathcal{R}_{w,1} + \sum_{u \in V} \lambda g(f_0)(u)f(u) \right\} \quad , \quad (1.49)$$

avec :

$$g(f_0) = (1 - 2f_0) \quad . \quad (1.50)$$

1.7.4.2 Modèle de Chan et Vese

Selon une logique similaire, Ta étend aux graphes la méthode de segmentation par contours actifs proposée dans [Chan et Vese, 2001].

L'objectif de l'approche proposée par Chan et Vese est de partitionner une image en deux régions composées de pixels dont les niveaux de gris respectifs sont homogènes. Cette méthode nécessite la définition d'un contour initial par l'utilisateur afin de permettre à l'algorithme d'obtenir une première évaluation des niveaux de gris moyens des deux régions. Le résultat de la segmentation est ensuite représenté sous la forme d'un ensemble de niveaux que l'algorithme déforme itérativement selon des critères d'homogénéité. Après chaque itération, les niveaux de gris moyens sont ré-évalués afin de correspondre au résultat qui vient d'être calculé, permettant ainsi une meilleure modélisation du problème. Étant donné

que cette méthode se base sur une caractéristique basée région, elle permet la détection d'objets ne présentant pas de contours nets.

Comme la méthode de ROF, l'approche proposée par Chan et Vese repose sur un problème de régularisation. Soit $f_0 : \Omega \subset \mathbb{R}^m \rightarrow [0, 1]$ la fonction initiale représentant l'image. Le modèle de Chan et Vese s'exprime dans le domaine continu à l'aide de la fonctionnelle d'énergie suivante :

$$\begin{aligned} \mathcal{J}_{CV}(f, f_0, \lambda_1, \lambda_2) = & \int_{\Omega} \|\nabla f\|_1 \\ & + \lambda_1 \int_{\Omega} f(x)(c_1 - f_0(x))^2 dx \\ & + \lambda_2 \int_{\Omega} (1 - f(x))(c_2 - f_0(x))^2 dx \quad , \end{aligned} \quad (1.51)$$

avec $f : \Omega \subset \mathbb{R}^m \rightarrow \{0, 1\}$ la solution recherchée indicatrice de la segmentation, et $c_1, c_2 \in \mathbb{R}$ deux variables représentant la valeur moyenne de la fonction f_0 respectivement à l'intérieur et à l'extérieur de la segmentation.

Les deux paramètres λ_1 et λ_2 permettent de pondérer l'influence du terme d'attache aux données vis-à-vis du terme de régularisation ainsi que l'importance relative de l'attache aux données pour chacune des deux régions. Cela permet la définition de contraintes concernant l'étendue des niveaux de gris autorisés dans chaque région. Étant donné que, dans l'essentiel de leurs tests, Chan et Vese ont fixés $\lambda_1 = \lambda_2$, Ta a choisi de ne conserver qu'un unique paramètre λ . La fonctionnelle d'énergie peut ainsi être reformulée de la façon suivante :

$$\begin{aligned} \mathcal{J}_{CV}(f, f_0, \lambda) = & \int_{\Omega} \|\nabla f\|_1 \\ & + \lambda \int_{\Omega} f(x)(c_1 - f_0(x))^2 + (1 - f(x))(c_2 - f_0(x))^2 dx \quad . \end{aligned} \quad (1.52)$$

Afin d'illustrer le fonctionnement du terme d'attache aux données, considérons que l'image à segmenter est composée de deux régions homogènes de niveaux de gris respectifs c_1 et c_2 . Ces deux régions seront respectivement identifiées par les valeurs 1 et 0 de la fonction indicatrice f . Considérons un pixel x appartenant à la première région pour lequel nous avons :

$$(f_0(x) - c_1)^2 \approx 0 \quad \text{et} \quad (f_0(x) - c_2)^2 \gg 0 \quad . \quad (1.53)$$

Si x est correctement identifié par l'algorithme de segmentation comme appartenant à la première région, $f(x) = 1$. Dans ces conditions le terme d'attache aux données contribue faiblement à la fonctionnelle d'énergie (1.52) :

$$f(x)(c_1 - f_0(x))^2 + (1 - f(x))(c_2 - f_0(x))^2 \approx 0 \quad . \quad (1.54)$$

En revanche s'il est identifié comme appartenant à la seconde région, $f(x) = 0$. Le terme d'attache aux données contribue donc fortement à la valeur de la fonctionnelle d'énergie (1.52) :

$$f(x)(c_1 - f_0(x))^2 + (1 - f(x))(c_2 - f_0(x))^2 \gg 0 \quad . \quad (1.55)$$

Un raisonnement similaire peut être appliqué aux pixels de la seconde région, illustrant ainsi le fait que tout pixel mal classé accroît l'énergie associée à la solution.

Afin de transformer cette fonctionnelle d'énergie en un problème de minimisation représenté à l'aide de la seconde classe de modèles variationnels, il est de nouveau nécessaire de procéder à une reformulation du terme d'attache aux données :

$$f \cdot (c_1 - f_0)^2 + (1 - f) \cdot (c_2 - f_0)^2 = \left((c_1 - f_0)^2 - (c_2 - f_0)^2 \right) f + (c_2 - f_0)^2 \quad . \quad (1.56)$$

Les variables c_1 et c_2 seront par la suite calculées avant chaque itération de l'algorithme et sont donc considérées comme constantes lors des calculs liés à l'annulation du gradient. Le dernier terme de l'équation (1.56) n'influant pas sur le calcul du gradient, il peut donc être ignoré. Il est ainsi possible de reformuler la fonctionnelle (1.52) par :

$$\mathcal{I}_{CV}(f, f_0, \lambda) = \int_{\Omega} \|\nabla f\|_1 + \lambda \int_{\Omega} f(x) \cdot g(f_0)(x) dx \quad , \quad (1.57)$$

avec :

$$g(f_0) = (c_1 - f_0)^2 - (c_2 - f_0)^2 \quad . \quad (1.58)$$

Cette fonctionnelle étant de nouveau un cas particulier du problème (1.41), elle peut être généralisée en un problème de minimisation défini sur les graphes :

$$\min_{f:V \rightarrow \{0,1\}} \left\{ \mathcal{E}_{CV}(f, f_0, \lambda) = \mathcal{R}_{w,1} + \sum_{u \in V} \lambda g(f_0)(u) f(u) \right\} \quad . \quad (1.59)$$

1.8 Conclusion

Comme nous avons pu le voir, il existe de nombreuses façons de représenter une image à l'aide d'un graphe. Cependant, si l'influence de certaines topologies sur la qualité des résultats a été constatée à quelques reprises, aucune théorie à ce sujet n'a été formulée.

Les trois premières familles de méthodes de segmentation d'images à base de graphes évoquées (arbres de couverture minimale, coupe minimales et marche aléatoire) représentent des sujets de recherche tout à fait dignes d'intérêt. En effet, elles dépendent exclusivement de la topologie des graphes et reposent sur d'importants résultats issue de la théorie des graphes dont le comportement, lorsqu'appliqués à des problèmes de vision par ordinateur, est encore mal maîtrisé. Ces méthodes sont cependant relativement lourdes. Malgré certaines optimisations et relaxations, les algorithmes de la théorie des graphes qu'elles emploient ne peuvent pas s'exécuter en temps linéaire (par rapport au nombre de nœuds du graphe). Ces méthodes sont donc, encore aujourd'hui, difficilement applicables à de réels problèmes de vision par ordinateur.

Dans cette thèse, nous nous intéresserons aux méthodes de segmentation par régularisation de graphes proposées par Ta [Ta, 2009]. Celles-ci peuvent s'exécuter en temps linéaire, mais s'inspirent surtout de théories beaucoup plus récentes, pour lesquelles de nombreuses options restent encore à explorer. Afin d'en améliorer la précision, nous proposons d'améliorer une de ces méthodes afin de lui permettre de tirer parti de caractéristiques de textures.

Chapitre 2

Segmentation d'images texturées par régularisation de graphes

Dans ce chapitre, nous présentons une approche permettant la prise en compte de caractéristiques de textures dans l'algorithme de segmentation par régularisation de graphe basé sur le modèle de Chan et Vese [Ta, 2009] présenté section 1.7.4.2. L'intérêt de cette contribution est de combiner deux techniques de traitement d'images qui manipulent les données relatives à l'image avec un plus haut niveau d'abstraction que les pixels qui la composent. Ces méthodes sont dites « non-locales ». La première est relative à l'utilisation d'une méthode basée sur les graphes, alors que la seconde repose sur l'utilisation de caractéristiques de textures.

Nous commencerons par établir un état de l'art des principales méthodes de caractérisation de textures, puis enchaînerons sur une présentation de notre approche. Nous illustrerons ensuite le fonctionnement de notre algorithme sur des images texturées artificielles, puis terminerons ce chapitre par une présentation des résultats obtenus sur des images échographiques 2D et 3D.

Les résultats de ces travaux ont été publiés dans [Fauchaux *et al.*, 2012b] et [Fauchaux *et al.*, 2012a].

2.1 État de l'art des méthodes de caractérisation de textures

La majorité des algorithmes de segmentation d'images travaille à partir de statistiques de premier ordre concernant la distribution des niveaux de gris : seule l'information relative à la probabilité d'obtenir un pixel d'un niveau de gris donné à un emplacement prédéfini est prise en compte.

Cependant, comme cela est décrit dans les travaux de Julesz sur le système psychovisuel humain [Julesz, 1962], et tel que l'illustre trivialement la figure 2.1, deux images de

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

textures « visuellement différentes¹ » peuvent présenter des statistiques de premier ordre identiques. Celles-ci ne sont donc pas suffisamment discriminantes, et limitent la capacité des algorithmes qui en dépendent à traiter des problèmes de vision complexes.

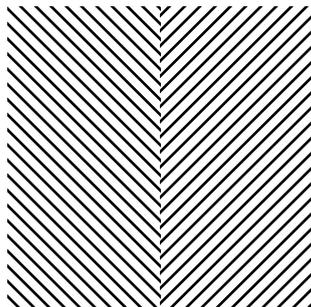


FIGURE 2.1 – Exemples de textures visuellement différentes mais dont les statistiques de premier ordre concernant les niveaux de gris sont identiques. Dans chacune des deux régions, le niveau de gris moyen vaut 192.

La notion de texture – bien qu'évidente pour notre cerveau – est relativement complexe à définir. Par ailleurs, comme l'illustre l'ensemble des définitions compilées dans [Tuceryan et Jain, 1990], celle-ci dépend souvent de l'objectif recherché par l'analyse d'une texture. En ce qui nous concerne, le point important est qu'au-delà de l'information relative à l'intensité des pixels, l'organisation spatiale de ces derniers joue un rôle extrêmement important. Ainsi, appliquée aux textures de la figure 2.1, une telle considération permet de mettre en évidence des continuités et discontinuités selon les angles de +45 et -45 degrés.

La notion de motifs permet d'offrir une description de plus haut niveau d'une texture, quantifiant ainsi son orientation, sa finesse ou encore sa périodicité. Ces concepts peuvent notamment être définis à l'aide de statistiques de second ordre, c'est à dire d'informations concernant la probabilité de trouver deux pixels de niveaux de gris donnés séparés par une distance et une orientation prédéfinies. Julesz souligne notamment que deux textures ayant des statistiques de second ordre identiques sont visuellement indiscernables.

Le rôle précis des différentes méthodes de caractérisation de textures est de fournir une description tonale et spatiale de la répartition des niveaux de gris dans une image. Dans la suite de cette section, nous allons présenter les principales méthodes de caractérisation de textures.

2.1.1 Caractéristiques d'Haralick

Haralick considère que la texture d'une image peut être décrite par le nombre et le type de ses primitives tonales ainsi que par leur organisation spatiale [Haralick *et al.*, 1973, Haralick, 1979]. Il propose donc de caractériser une texture à partir d'un ensemble de statistiques de second ordre qui permettent la mise en évidence des motifs dans la distribution et l'agencement des niveaux de gris qui la composent.

1. Julesz ne s'intéresse dans ses travaux qu'à la capacité du cerveau à discriminer sans efforts deux textures distinctes, c'est à dire sans procéder à un examen détaillé de celles-ci.

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

Il s'appuie pour cela sur la notion de matrice de co-occurrences des niveaux de gris². Le but d'une matrice de co-occurrence est de dénombrer les occurrences de paires d'évènements. Dans le cadre de l'analyse de textures, Haralick considère cette paire d'évènements comme étant une transition de niveaux de gris, c'est à dire l'apparition de deux pixels d'intensités données séparés par une relation de voisinage $\Delta = (\delta_x, \delta_y)$ ³. Une telle matrice recense alors les différentes transitions de niveaux de gris dans l'image. Elle est le plus souvent utilisée sous sa forme normalisée, et représente alors une distribution de probabilités.

Soit I une image composée de N niveaux de gris. Étant donnée une relation de voisinage Δ , la matrice de co-occurrence associée à cette image est définie pour chaque paire de niveaux de gris i et j par :

$$C_{\Delta}(i, j) = \text{Card} \{ (\mathbf{p}, \mathbf{q}) \in I \text{ tel que } \mathbf{q} = \mathbf{p} + \Delta \text{ et } I(\mathbf{p}) = i \text{ et } I(\mathbf{q}) = j \} \quad , \quad (2.1)$$

ou \mathbf{p} et \mathbf{q} représentent les coordonnées des pixels, et $I(\cdot)$ l'intensité du niveau de gris aux coordonnées spécifiées.

La taille d'une matrice de co-occurrence est quadratiquement proportionnelle au nombre de niveaux de gris dans l'image. Afin de permettre l'émergence de certaines tendances ou schémas dans la matrice, il est important de ne pas considérer un nombre trop important de niveaux de gris, au risque d'obtenir des valeurs statistiquement insignifiantes. Il est ainsi courant de réduire l'étendue des valeurs possibles en ré-échantillonnant l'image initiale, projetant ainsi l'ensemble des niveaux de gris dans un espace de taille réduite. Il n'existe cependant aucune règle indiquant le nombre de niveaux de gris nécessaires à une description optimale d'une texture, mais des valeurs situées entre 16 et 32 sont relativement courantes.

Dans ses travaux, Haralick ne tient pas compte de la direction de la relation de voisinage. Lors du calcul, il est donc nécessaire de considérer la relation de voisinage initiale ainsi que son opposée :

$$C_{\Delta}(i, j) = \text{Card} \{ (\mathbf{p}, \mathbf{q}) \in I \text{ tel que } \mathbf{q} = \mathbf{p} + \Delta \text{ et } (I(\mathbf{p}) = i \text{ et } I(\mathbf{q}) = j \text{ ou } I(\mathbf{p}) = j \text{ et } I(\mathbf{q}) = i) \} \quad . \quad (2.2)$$

La matrice de co-occurrence ainsi générée est symétrique : $C_{\Delta}(i, j) = C_{\Delta}(j, i)$. La figure 2.2 présente une image synthétique et la matrice de co-occurrences obtenue pour le décalage $\Delta = (1, 0)$.

À partir de la matrice de co-occurrences normalisée, Haralick propose un ensemble de quatorze statistiques permettant d'évaluer différentes propriétés de la texture analysée, telles que son niveau de contraste, sa régularité ou encore son degré d'entropie. Le lecteur intéressé trouvera en annexe A plus d'informations concernant le calcul de ces caractéristiques. Nous y faisons notamment le point sur les différents noms et formulations que la littérature leur associe et discutons de la validité de certaines d'entre elles. Nous en profitons aussi pour compléter les travaux d'Haralick et proposons une interprétation de chacune des caractéristiques ainsi qu'une évaluation de leur domaine de valeurs.

2. Aussi appelée matrice de dépendance spatiale des niveaux de gris, ou encore GLCM (de l'anglais *Grey-Level Co-occurrence Matrix*).

3. La notation Δ sera conservée car cette notion de relation de voisinage peut être étendue à des dimensions supérieures.

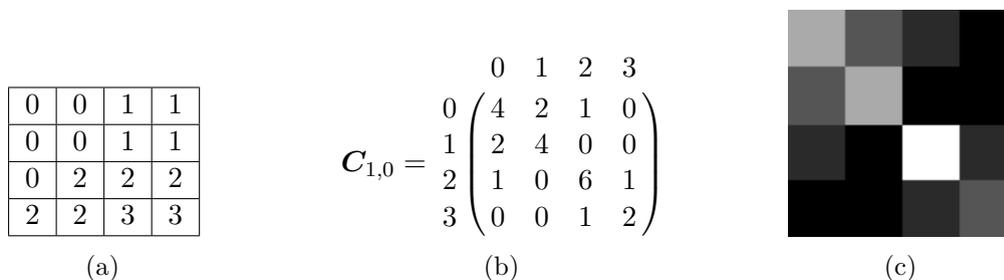


FIGURE 2.2 – Exemple de matrice de co-occurrences. (a) : une image de taille 4×4 composée de 4 niveaux de gris. (b) : la matrice de co-occurrences calculée pour le décalage $\Delta = (1, 0)$. (c) : la même matrice représentée sous forme de carte.

Étant donné que le décalage Δ définit la direction selon laquelle une texture est étudiée, une matrice de co-occurrences est un outil fortement anisotrope qui ne caractérisera pas de la même manière deux textures de motifs identiques mais d'orientations différentes. Afin d'assurer une invariance à la rotation, Haralick suggère de caractériser une texture selon quatre orientations, 0° , 45° , 90° et 135° , puis de considérer deux vecteurs composés respectivement de la moyenne et de l'écart-type de chacune des quatorze caractéristiques.

2.1.2 Modèles auto-régressifs

Un modèle auto-régressif (*AutoRegressive Model*, ou *AR model*) est un outil statistique qui vise à exprimer un processus aléatoire discret à valeurs réelles par le biais d'une combinaison linéaire de ses propres valeurs passées :

$$X_t = \sum_{i=1}^p \varphi_i X_{t-i} + \varepsilon \quad , \quad (2.3)$$

avec X le processus aléatoire et ε un bruit blanc d'écart-type σ_ε dont le but est de donner un caractère non-déterministe au modèle. On appelle « paramètres du modèle » les variables $\{\varphi_1, \dots, \varphi_p\}$ qui peuvent être considéré comme une représentation spectrale du signal, et donc une caractéristique de textures. Lorsque le modèle fait référence à p variables passées, il est dit d'ordre p .

À partir d'un échantillon $X = \{X_1, \dots, X_n\}$, la conception d'un modèle auto-régressif consiste à estimer les paramètres φ_i ainsi que l'écart-type σ_ε du bruit en optimisant un critère d'attache aux données compte tenu de la présence d'un élément de caractère aléatoire. Plusieurs méthodes sont disponibles, la plupart étant basées sur des outils statistiques : l'estimation du maximum de vraisemblance [Parzen, 1968], les moindres carrés [Akaike, 1969] ou encore les équations de Yule-Walker (application du raisonnement de la méthode des moments à partir d'une estimation des auto-covariances des éléments de l'échantillon) [Chan et Langford, 1982]. D'autres approches, notamment optimisées pour les données 2D, ont plus récemment vu le jour, telles que l'algorithme rapide 2D des moindres carrés récursif en treillis [Alata et Olivier, 2003].

Initialement appliqués à la modélisation et à la prédiction de signaux 1D tels que les séries temporelles [Akaike, 1969], les modèles auto-régressifs ont ensuite été étendus

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

aux signaux 2D pour le traitement d'images, tout d'abord en les considérant comme des signaux 1D, puis en intégrant la structure matricielle des images à la modélisation [Haralick, 1979, Maragos *et al.*, 1984]. Les modèles auto-régressifs sont utilisés dans des applications telles que la synthèse de textures [McCormick et Jayaramamurthy, 1974], la segmentation [Deguchi et Morishita, 1978, Bouman et Liu, 1991, Ergen, 2012], ou encore la compression d'image [Maragos *et al.*, 1984, Debure et Kubato, 1998].

Lorsqu'un tel modèle est appliqué au traitement d'images, excepté pour certains cas de synthèse de textures, la notion de « valeur passée » peut ne plus être pertinente. Le modèle peut ainsi considérer l'ensemble de pixels situés dans un voisinage donné – il est alors qualifié de modèle auto-régressif symétrique (*SAR*, pour *Symmetric AutoRegressive*). Dans cette configuration, il est alors possible de rendre le modèle invariant à la rotation : *CSAR* (*Circular Symmetric AutoRegressive model*) [Kashyap et Khotanzad, 1986], *RISAR* (*Rotation-Invariant Symmetric AutoRegressive model*) [Mao et Jain, 1992].

Enfin, des travaux ont permis d'appliquer cette approche dans un contexte de multi-résolution : *MR-SAR*, *MR-RISAR* [Mao et Jain, 1992].

2.1.3 Caractéristiques de textures dérivées de méthodes de traitement du signal

Ces méthodes sont basées sur un principe commun : mesurer la similarité du signal à étudier avec un second signal, connu, possédant des propriétés particulières (fréquence, forme...). La mesure ainsi obtenue permet de quantifier la présence de ces propriétés dans le signal étudié.

Mathématiquement, cela consiste à calculer la corrélation croisée entre ces deux signaux. Dans le cadre du traitement de signaux continus à une dimension, si $f(\cdot)$ désigne la fonction à étudier, et $g(\cdot)$ la fonction à laquelle f est comparée, la corrélation croisée ($f \star g$) entre f et g est obtenue par convolution de f avec le conjugué complexe \bar{g} de g :

$$(f \star g) = \int_{-\infty}^{\infty} f(\tau) \cdot \bar{g}(\tau) \, d\tau \quad (2.4)$$

Cette formule est transposable aux signaux discrets, définis sur un support compact, ou encore à plusieurs dimensions. En particulier, lorsque appliquée au traitement d'images 2D, la formule associée à ce calcul de corrélation croisée est :

$$(f \star g) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \cdot \bar{g}(x, y) , \quad (2.5)$$

une image de résolution $M \times N$ étant ici considérée.

2.1.3.1 Transformée de Fourier

La transformée de Fourier est une transformation mathématique qui vise à décrire les composantes d'un signal quelconque non plus dans un contexte temporel ou spatial – c'est à dire en exprimant la valeur du signal en chacune de ses coordonnées – mais fréquentiel.

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

Elle repose pour cela sur le théorème de Fourier, qui énonce que tout signal réel continu peut être exprimé sous la forme d'un polynôme trigonométrique (de taille potentiellement infinie), c'est à dire comme une somme de fonctions sinus et cosinus d'amplitudes et de fréquences déterminées.

Dans sa version 2D discrète, la transformée de Fourier s'appelle la DFT^4 et s'exprime à partir de l'équation 2.5 en considérant la fonction g comme une sinusoïde complexe :

$$\mathcal{F}(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \exp\left(-2i\pi\left(\frac{xu}{N} + \frac{yv}{M}\right)\right), \quad (2.6)$$

u et v étant les paramètres qui caractérisent la période de la sinusoïde suivant les deux axes du plan, et $\mathcal{F}(u, v)$ le coefficient de Fourier associé.

La transformée de Fourier peut être vue comme une décomposition⁵ en composantes fréquentielles, l'amplitude d'un coefficient de Fourier $\mathcal{F}(u, v)$ étant proportionnelle à la contribution de la composante de paramètres u et v dans l'image d'origine. Le spectre de puissance d'une image est ainsi défini comme étant la magnitude (aussi nommé valeur absolue ou encore module) de sa transformée de Fourier. Il est ainsi envisageable de caractériser l'aspect fréquentiel d'une texture, en particulier sa période et son orientation, par une analyse des coefficients de Fourier de forte amplitude (voir figure 2.3).

Plus particulièrement, la présence de couronnes ainsi que de secteurs angulaires présentant une forte énergie permet de quantifier la granularité d'une texture ainsi que son orientation (voir figure 2.4).

Cette approche a notamment été employée pour de la reconnaissance de terrains par classification de textures [Weszka, 1976].

2.1.3.2 Transformée en ondelettes

L'un des reproches souvent adressé à la transformée de Fourier est son incapacité à fournir des informations concernant la localisation des composantes fréquentielles dans le temps. Par analogie avec la musique, cela signifie qu'il est possible de connaître l'ensemble des fréquences jouées, mais il est en revanche impossible de savoir si une note précise a été jouée à un instant donné.

Afin de combler cette lacune, Gabor initia la théorie des ondelettes en proposant de limiter la portée spatiale de la sinusoïde présente dans la transformée de Fourier traditionnelle [Gabor, 1946]. Il employa pour cela une transformée de Fourier fenêtrée, dans laquelle le signal à analyser est convolué avec une sinusoïde modulée par une fonction Gaussienne. La sinusoïde a désormais la forme d'une petite onde, ce qui donnera naissance à la notion d'ondelette : un signal oscillant défini sur un support compact. Ce type de signal est aussi connu sous le nom d'« atome de Gabor ».

Tout comme la transformée de Fourier traditionnelle, celle de Gabor permet la décomposition ainsi que la reconstruction d'un signal selon ses composantes fréquentielles.

4. Pour *Discrete Fourier Transform*.

5. La transformée de Fourier est inversible : il est possible de reconstruire l'image à partir de ses composantes fréquentielles.

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

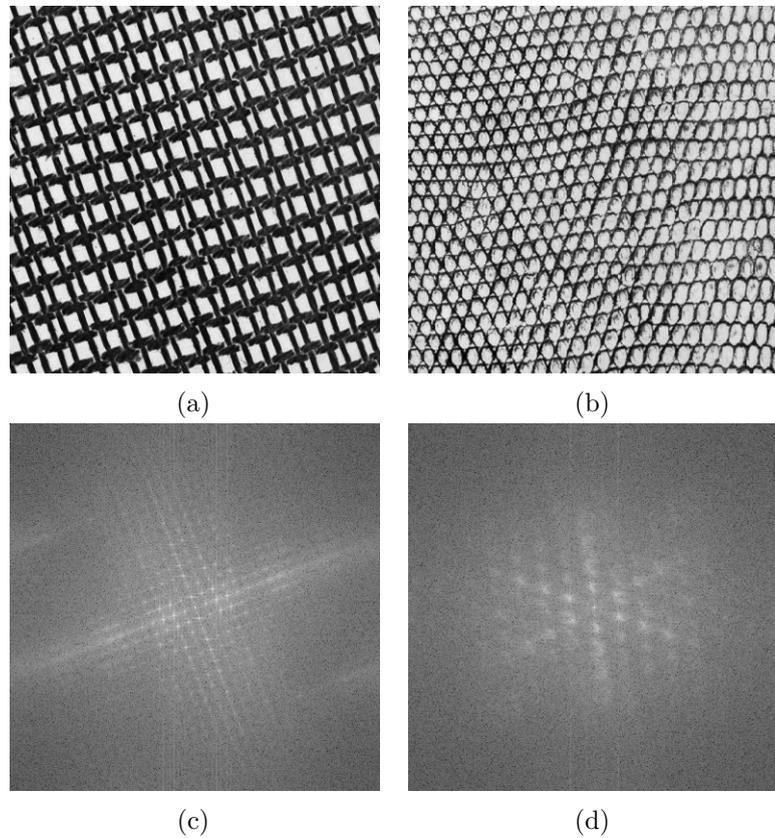


FIGURE 2.3 – (a), (b) : Exemples de textures issues de la base d'images Brodatz [Brodatz, 1966]. (c), (d) : Leurs spectres de puissance respectifs.

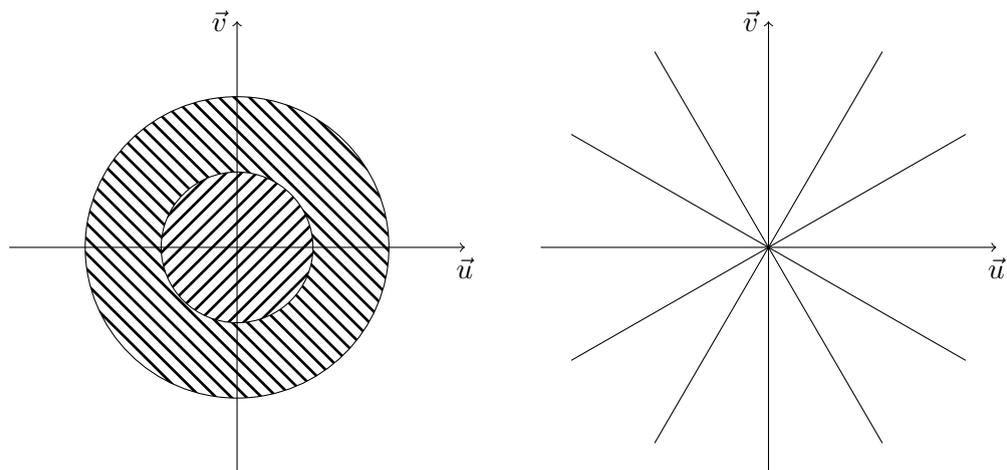


FIGURE 2.4 – Régions du spectre de puissance permettant l'identification de la fréquence (couronnes) et de l'orientation (secteurs angulaires) des composantes fréquentielles d'une texture.

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

La décomposition ainsi obtenue est cependant redondante, car l'ensemble formé par les translations en temps et en fréquence de l'ondelette mère ne constitue pas une base de l'espace à analyser. Cette sous-optimalité est sans aucun doute une des raisons du peu d'intérêt porté à cette transformée. Une version parcimonieuse de cette approche, les filtres de Gabor (voir section 2.1.3.3), aura cependant un plus grand succès.

La théorie des ondelettes a connu un essor important dans les années 80. Alors qu'il travaillait sur l'étude de signaux sismiques, Morlet proposa de remplacer les atomes de Gabor par d'autres fonctions oscillantes : les ondelettes [Grossmann et Morlet, 1984]. À partir d'une ondelette mère $\psi(x)$, et plus particulièrement de ses translations et dilatations $\psi_{s,t}(x) = \sqrt{s} \psi(sx + t)$, $(s, t) \in \mathbb{R}^+ \times \mathbb{R}$, il devient possible de décomposer tout signal continu réel de carré intégrable.

Dans [Meyer, 1985], il a été prouvé l'existence d'ondelettes mères $\psi(x)$ telles que l'ensemble des translations et dilatations $\psi_{j,k}(x) = \sqrt{2^j} \psi(2^j x + k)$, $(j, k) \in \mathbb{Z}^2$ d'une ondelette forme une base de l'espace des fonctions réelles de carré intégrable. Ce résultat est particulièrement important, puisqu'il permet d'assurer l'absence de redondance dans les éléments issus d'une décomposition en ondelettes.

Les méthodes de décomposition basées sur les ondelettes ont la particularité de fournir une analyse multi-échelle d'un signal. En effet, si l'on considère une ondelette $\psi()$ et la base $\{\psi_{j,k}\}_{(j,k) \in \mathbb{Z}^2}$ générée par ses translations et dilatations, les coefficients de la transformée associés à un niveau de dilatation j sont représentatifs du détail du signal étudié à l'échelle 2^j , tandis que les coefficients liés à des échelles supérieures fournissent des informations liées à des niveaux de détails plus fins.

Cependant, la décomposition complète d'un signal requiert de considérer l'ensemble des échelles possibles. À partir de ce constat, Mallat développe une méthode de décomposition multi-résolution de signal [Mallat, 1989]. L'idée est d'associer à une ondelette mère $\psi()$ une fonction d'échelle $\phi()$, elle aussi utilisée pour générer une famille de fonctions $\phi_{j,k}(x) = \sqrt{2^j} \phi(2^j x + k)$, $(j, k) \in \mathbb{Z}^2$, et dont le but est de fournir une série d'approximations d'un signal à des échelles successives⁶.

Ces deux fonctions sont mathématiquement liées de façon à ce qu'à une échelle 2^j et une translation k données, les fonctions $\phi_{j,k}()$ et $\psi_{j,k}()$ soient complémentaires. En d'autres termes, l'ondelette fait office de filtre passe-haut et permet l'extraction des composantes de haute fréquence, tandis que la fonction d'échelle joue le rôle de filtre passe-bas et permet d'approximer le signal, codant ainsi ce qui n'a pas pu l'être avec l'ondelette.

Ce processus de décomposition est en réalité implémenté à l'aide de deux filtres de convolution, G et H , lesquels sont respectivement générés à partir de l'ondelette et de la fonction d'échelle. Un signal initial S est ainsi convolué avec chacun de ces deux filtres, les réponses obtenues étant ensuite rééchantillonnées avec un facteur de 2. Les valeurs obtenues à partir des filtres G et H forment respectivement ce que l'on appelle les coefficients de détails et d'approximation. Ce processus peut ensuite être réappliqué récursivement en considérant les coefficients d'approximation en lieu et place du signal initial. Ces coefficients représentant une approximation du signal à une échelle inférieure, chaque itération permet

6. À ce titre, cette famille de fonctions ne constitue pas une base de l'espace des fonctions de carré intégrable car toute approximation à une échelle 2^j donnée contient une partie de l'information contenue à l'échelle 2^{j-1}

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

d'obtenir une nouvelle description du signal à une nouvelle échelle, et donc pour de nouvelles bandes de fréquence. Grâce à l'utilisation de la fonction d'approximation, ce processus peut être arrêté à tout moment, une décomposition en n niveaux étant au final constituée de n vecteurs de coefficients de détail et d'un unique vecteur de coefficients d'approximation. La figure 2.5 illustre le principe d'une décomposition en ondelettes 1D.

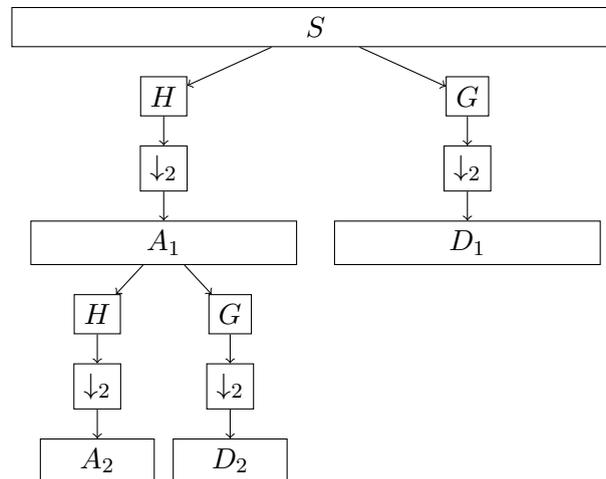


FIGURE 2.5 – Décomposition en ondelettes d'un signal 1D. S correspond au signal initial, A_x et D_x respectivement aux coefficients d'approximation et de détails associés à l'échelle x , G et H aux filtres passe-haut et passe-bas associés respectivement à l'ondelette et à la fonction d'échelle. \downarrow_2 dénote un rééchantillonnage d'un facteur 2.

Le passage à des signaux 2D discrets tels que les images est relativement simple à mettre en place. Comme illustré figure 2.6, chaque étape de l'algorithme de décomposition se passe désormais en deux temps. Chaque ligne du signal est tout d'abord décomposée, ce qui génère deux jeux de coefficients :

- D : détail horizontal,
- A : approximation horizontale.

Les colonnes de ces deux jeux de coefficients sont ensuite décomposées de la même façon. Nous obtenons ainsi quatre nouveaux jeux de coefficients :

- DD : détail,
- DA : détail horizontal et approximation verticale,
- AD : détail verticale et approximation horizontale,
- AA : approximation.

Le processus de décomposition peut ensuite être poursuivi à partir des coefficients d'approximation AA. La figure 2.6 illustre le principe d'une décomposition en ondelettes 2D.

Lors d'une transformée en ondelettes, la décomposition d'un signal est obtenue en traitant récursivement les coefficients d'approximation globale, ce qui permet une représentation de plus en plus fine des basses fréquences. Afin de permettre une meilleure représentation des hautes fréquences, la méthode des paquets d'ondelettes propose de décomposer aussi les coefficients de détails obtenus pour chaque échelle [Coifman *et al.*, 1992]. La décomposition passe ainsi d'une décomposition de type pyramidal en une décomposition

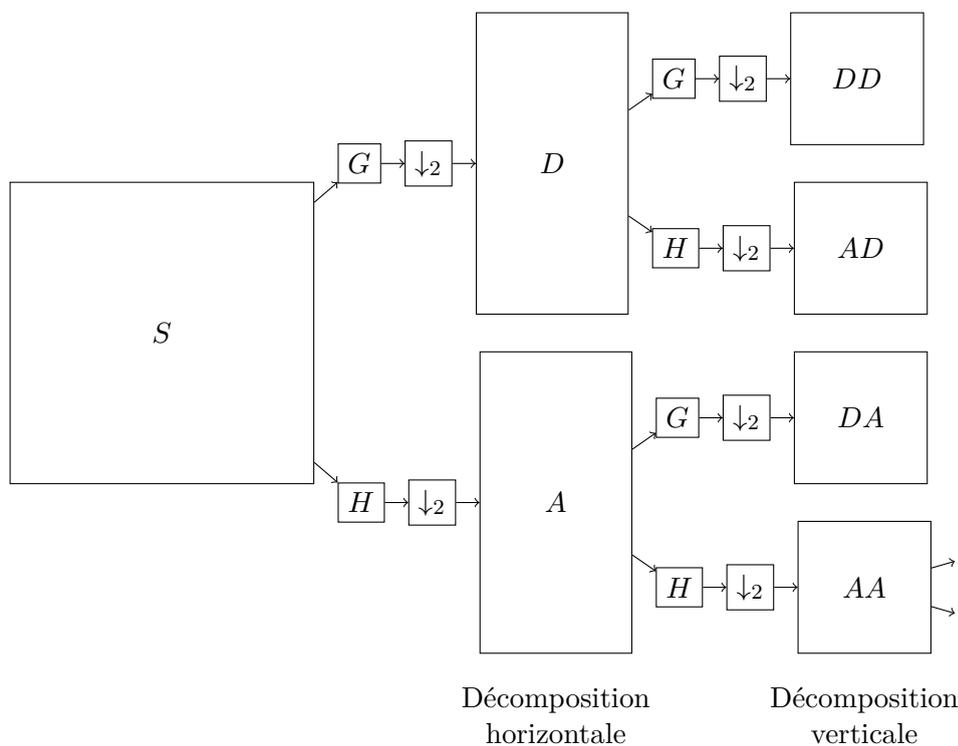


FIGURE 2.6 – Décomposition en ondelettes d'un signal 2D. Le signal est tout d'abord décomposé horizontalement, puis verticalement.

de type arborescente.

Coifman et Wickerhauser introduisent également un algorithme de sélection de la meilleure base [Coifman et Wickerhauser, 1992]. Il permet d'optimiser la représentation d'un signal à partir de ses coefficients d'ondelettes en mesurant la « quantité d'information » contenue par chaque décomposition. Sa mise en œuvre consiste à réaliser une décomposition par paquets complète du signal puis, en partant du niveau le plus bas de l'arborescence, à comparer l'énergie⁷ de chaque décomposition à la somme des énergies des décompositions filles. Ainsi, si l'une d'entre elles présente une énergie largement supérieure à celle de ses filles, cela signifie qu'il n'est pas nécessaire de la décomposer car aucune information complémentaire ne pourra être obtenue. Les décompositions filles qui ne satisfont pas cette condition sont donc supprimées de l'arbre de décomposition. Lorsque cet algorithme a été appliqué à la totalité de l'arbre, les décompositions associées à l'ensemble des feuilles forment une base qui représente de façon optimale le signal initial.

C'est précisément cette approche qui a été retenue par les premiers travaux de caractérisation de textures basés sur la décomposition en ondelettes. Dans [Laine et Fan, 1993], cette mesure d'énergie sert à caractériser les différentes bandes de fréquences que représente la décomposition. Une signature de la texture peut ainsi être créée, puis identifiée à l'aide d'un classificateur basé sur une mesure de distance ou encore un réseau de neurones. Selon un principe similaire, il a aussi été proposé de réduire la signature d'une texture aux énergies

7. Une mesure alternative basée sur l'entropie est aussi proposée.

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

des k décompositions les plus pertinentes [Chang et Kuo, 1993].

2.1.3.3 Bancs de filtres

À la différence de la transformée de Fourier qui réalise une décomposition exhaustive d'une texture en ses composantes fréquentielles, les méthodes à bancs de filtres visent à caractériser une texture par une mesure de la corrélation de celle-ci avec un ensemble prédéfini d'éléments de textures ayant des propriétés (forme, orientation, échelle...) particulières. De par l'utilisation de filtres localisés dans l'espace – alors que la transformée de Fourier ne s'intéresse qu'à la composante fréquentielle – cette approche permet de caractériser une texture en un point précis d'une image.

Les mesures de corrélations obtenues en un pixel donné par l'ensemble des filtres composant un banc forment ce que l'on appelle couramment la « réponse du banc ». Celle-ci peut être vue comme la signature de la texture présente en ce point vis-à-vis de ce banc et peut, associée à une mesure de distance, être utilisée pour classifier des textures.

Filtres de Gabor [Gabor, 1946]

Les filtres de Gabor sont les premiers à avoir été appliqués à la caractérisation de textures [Turner, 1986]. Ils correspondent à un cas particulier de la transformée de Fourier, le contrôle de la phase de la sinusoïde ayant été ajouté. Par ailleurs, afin de limiter la portée de la transformation, et ainsi d'assurer la localisation spatiale de la transformée, celle-ci est fenêtrée à l'aide d'une enveloppe gaussienne. Un filtre de Gabor s'exprime alors par :

$$g(x, y, u, v, \phi, \sigma) = \exp\left(2i\pi(ux + vy) + \phi\right) \cdot \exp\left(-\left(\frac{x^2 + y^2}{2\sigma^2}\right)\right), \quad (2.7)$$

avec u et v les paramètres définissant la fréquence et l'orientation de la sinusoïde, ϕ sa phase et σ l'écart-type associé à l'enveloppe gaussienne.

La réponse obtenue par convolution d'un filtre en chacun des pixels composant la texture permet de caractériser une texture. Un banc composé de 16, 24, 32 ou 48 filtres combinant plusieurs orientations, échelles et phases est généralement utilisé (voir figure 2.7).

Dans [Manjunath et Ma, 1996a], il est proposé de définir une signature d'une texture à l'aide de statistiques de premier ordre (moyenne, écart-type) pour chacun des filtres du banc. Des algorithmes de partitionnement, tels que les k -moyennes ou encore les arbres-kd peuvent aussi être utilisés afin de fournir un ensemble de « réponses types » d'une texture [Jain et Farrokhnia, 1991, Rubner et Tomasi, 1999].

Filtres de Schmid [Schmid, 2001]

L'auteur s'inspire des filtres de Gabor pour proposer un banc de treize filtres dit « génériques » qui résultent de la composition d'une sinusoïde exprimée sous forme radiale et d'une fenêtre gaussienne :

$$g(x, y, \sigma, \tau) = F_0(\sigma, \tau) + \cos\left(\frac{\pi\tau\sqrt{x^2 + y^2}}{\sigma}\right) \cdot \exp\left(\frac{x^2 + y^2}{2\sigma^2}\right), \quad (2.8)$$

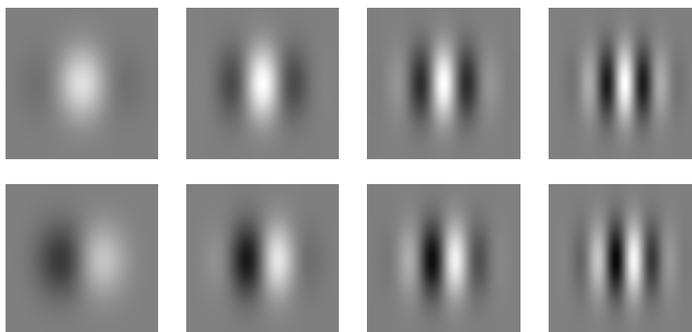
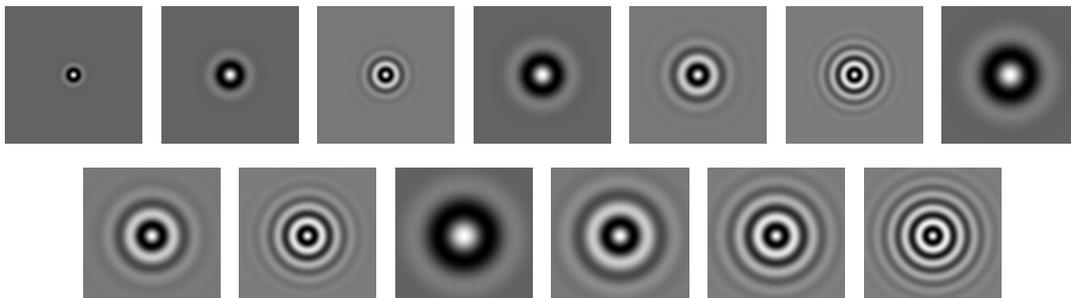


FIGURE 2.7 – Exemples de filtres de Gabor.

Exemples de filtres de Gabor pour différentes valeurs de phase, d'échelles et de fréquences. Le niveau de gris correspond à la valeur du filtre en chaque pixels.

τ représentant le nombre de cycle de la fonction harmonique, σ l'écart-type associé à la fenêtre gaussienne, et F_0 une compensation permettant de donner au filtre une moyenne nulle (les filtres sont par ailleurs généralement normalisés).

Le banc proposé est généré pour (σ, τ) prenant les valeurs de $(2, 1)$, $(4, 1)$, $(4, 2)$, $(6, 1)$, $(6, 2)$, $(6, 3)$, $(8, 1)$, $(8, 2)$, $(8, 3)$, $(10, 1)$, $(10, 2)$, $(10, 3)$, $(10, 4)$ (voir figure 2.8).

FIGURE 2.8 – Filtres de Schmid obtenus pour les valeurs (σ, τ) de $(2, 1)$, $(4, 1)$, $(4, 2)$, $(6, 1)$, $(6, 2)$, $(6, 3)$, $(8, 1)$, $(8, 2)$, $(8, 3)$, $(10, 1)$, $(10, 2)$, $(10, 3)$, $(10, 4)$.

L'utilisation de filtres présentant une symétrie radiale permet à l'auteur de générer un ensemble de caractéristiques de textures invariantes à la rotation. De la même façon que les approches appliquées aux filtres de Gabor, Schmid a recours à l'algorithme des k-moyennes pour générer un ensemble de signatures types. Il justifie alors l'intérêt d'un banc de taille réduite (comparé aux filtres de Gabor de taille 32 ou 48 régulièrement utilisés) par une réduction des risques liés à l'emploi de descripteurs de trop grande dimensionnalité.

Filtres de Leung-Malik (LM) [Leung et Malik, 2001]

Le banc LM est composé de 48 filtres générés à partir des dérivées premières et secondes de Gaussiennes selon 6 orientations et 3 échelles, 4 Gaussiennes et 8 Laplaciens de Gaussiennes (voir figure 2.9).

Ce banc de filtres est là encore couplé à l'algorithme des k-moyennes afin d'établir un dictionnaire de réponses types. L'originalité de cette approche réside dans l'application de

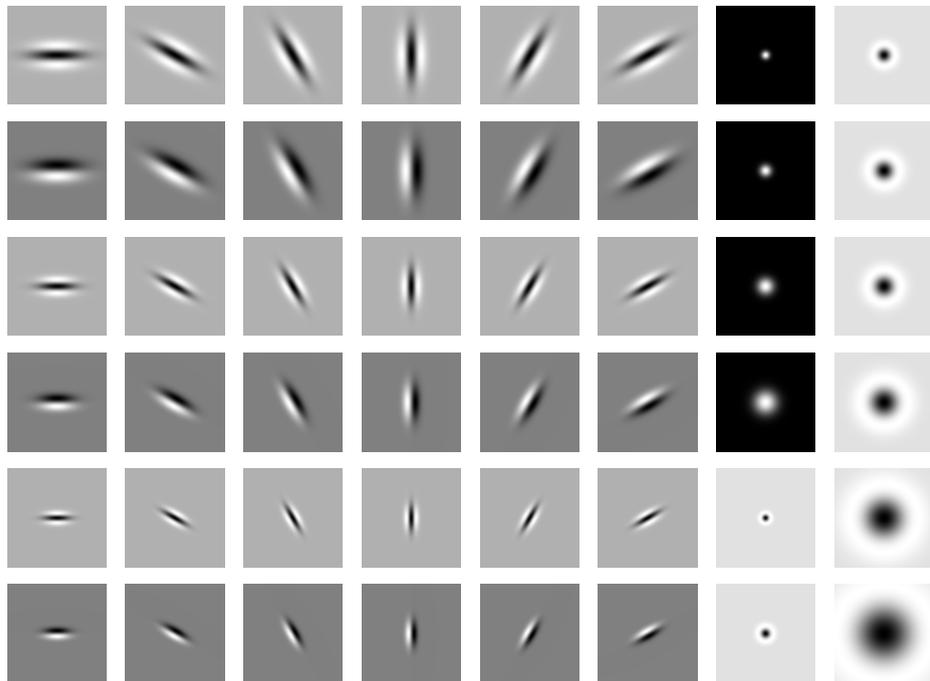


FIGURE 2.9 – Filtres de Leung-Malik.

ces filtres à la reconnaissance de textures 3D, composées des acquisitions photographiques de l'état de surface de matériaux réalisées sous différentes combinaisons d'angles d'incidence et de conditions d'éclairage.

Filtres de réponse maximum (MR) [Varma et Zisserman, 2002]

Les bancs MR⁸ sont de conception similaire au banc LM, puisque composés d'une Gaussienne, de son Laplacien, ainsi que des dérivées premières et secondes de Gaussiennes selon plusieurs échelles et orientations. Au-delà de la réduction du nombre de Gaussiennes et de Laplaciens, la différence vient du fait que seules les réponses des filtres directionnels présentant le plus grand intérêt seront par la suite considérées. En particulier, afin de permettre l'obtention d'une signature de texture invariante à la rotation, seule la réponse maximum d'un filtre parmi toutes ses orientations sera prise en compte.

Le banc MR4, illustré figure 2.10, ne considère qu'une unique échelle pour 6 orientations. 14 filtres sont ainsi générés, pour seulement 4 réponses conservées. Le banc MR8 considère quant à lui 3 échelles, portant le nombre de filtres à 38, et à 8 le nombre de réponses. Le banc MR4S, composé des mêmes filtres que le banc MR8, assure enfin l'invariance à la rotation ainsi qu'au changement d'échelle en ne conservant qu'une unique réponse parmi l'ensemble des orientations et échelles possibles d'une même famille de filtres.

Dans [Varma et Zisserman, 2002, Varma et Zisserman, 2005], ces filtres ont été comparés à ceux de Leung-Malik ainsi qu'à ceux de Schmid dans le cadre de la classification de textures 3D. Les tests réalisés montrent que les bancs MR8 et de Schmid obtiennent de

8. Pour *Maximum Response*.

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

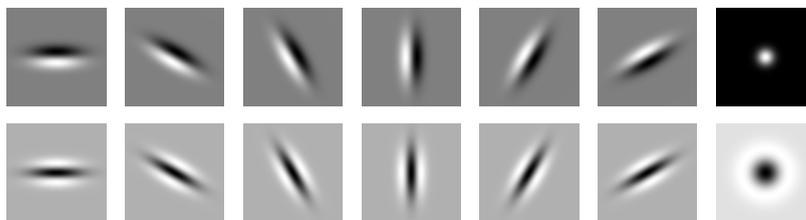


FIGURE 2.10 – Banc de filtres MR4. Pour chaque filtre directionnel, une unique réponse sera conservée parmi toutes les orientations possibles, permettant ainsi l'obtention d'un descripteur de texture invariant à la rotation.

meilleurs résultats que le banc LM, soulignant non seulement l'intérêt de travailler avec des descripteurs invariants à la rotation, mais aussi la possibilité de travailler avec des descripteurs de taille réduite. En revanche, les mauvaises performances du banc MR4 indiquent, quant à elles, la pertinence des approches multi-échelles.

2.1.4 Modèle binaire local

Plus connu sous son nom anglais, *local binary patterns* (LBP), cette méthode s'inspire des travaux réalisés sur les spectres de textures [Wang et He, 1990], dans lesquels un code est associé à chaque pixel en fonction de la différence de contraste de ce pixel avec chacun de ses voisins.

Wang et He définissent la texture d'un pixel à l'aide d'un code ternaire. Si l'on considère un pixel p , chacun de ses huit plus proches voisins se voit associé une valeur 0, 1 ou 2 indiquant si le niveau de gris de celui-ci est respectivement inférieur, égal ou supérieur au niveau de gris du pixel p . Selon un ordre de parcours du voisinage prédéfini, ces différentes valeurs permettent de construire un code ternaire qui identifie la texture localement associée au pixel p . L'ensemble de ces codes définit ce que l'on appelle un dictionnaire d'« unités de texture ».

Afin de réduire la taille de ce dictionnaire, il a été proposé dans [Ojala *et al.*, 1996] d'utiliser un code binaire en ne considérant que les valeurs 0 et 1 pour les états « inférieur ou égal » et « supérieur » (voir figure 2.11). Le dictionnaire passe ainsi de $3^8 = 6561$ éléments pour les spectres de textures à $2^8 = 256$ pour la méthode des motifs binaires locaux. À partir des codes obtenus pour l'ensemble des pixels d'une région représentative d'une texture, un histogramme des motifs présents est construit. Ces histogrammes peuvent alors être comparés par le biais de mesures de similarités, permettant ainsi la reconnaissance et la classification des textures associées.

Étant basée sur une mesure du contraste, cette méthode est invariante aux changements d'intensité. Afin de permettre une analyse multiéchelle, il est proposé dans [Ojala *et al.*, 2002] de recourir à un voisinage constitué de N points équirépartis sur un cercle de rayon R , le choix des valeurs associées à ces deux paramètres étant laissé à l'utilisateur. Une méthode permettant de rendre les motifs binaires locaux invariants à la rotation y est également proposée. Des extensions à la 3D ont par ailleurs été proposées [Fehr et Burkhardt, 2008, Paulhac *et al.*, 2008].

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

61	70	68
45	42	42
36	35	37

(a)

1	1	1
1		0
0	0	0

(b)

1	2	4
16		8
32	64	128

(c)

FIGURE 2.11 – Méthode de calcul d'un motif binaire local. (a) : un pixel et son voisinage, (b) : motif binaire local, (c) : pondération associée aux différents pixels du voisinage. Le motif de valeur $1 + 2 + 4 + 16 = 23$ est attribué au pixel.

Les différentes invariances que propose cette approche font des motifs binaires locaux une méthode robuste, capable de fournir des caractéristiques à la fois statistiques et structurelles sur une texture. Sa simplicité en fait par ailleurs un outil de choix pour les applications temps réel, comme par exemple la reconnaissance de visages [Zhao et Pietikäinen, 2006, Lei *et al.*, 2013].

2.1.5 Champs de Markov aléatoires

Un champs de Markov est un outil statistique qui permet de modéliser un phénomène stochastique par une analyse de ses dépendances spatiales. D'un point de vue mathématique, un tel phénomène correspond à un ensemble de variables aléatoires mises en relation par un système de voisinage. Sa modélisation à l'aide d'un champ de Markov aléatoire revient à exprimer la probabilité de réalisation de chacune des variables uniquement en fonction des valeurs courantes des variables aléatoires voisines.

La notion de texture étant intimement liée à la notion de dépendance spatiale, la modélisation d'une image par champs de Markov constitue une approche relativement populaire qui trouve des applications telles que la synthèse et la compression d'images [Cross et Jain, 1983, Chellappa *et al.*, 1985, Efros et Leung, 1999], le débruitage d'images [Malfait et Roose, 1997, Pierce et Ulaby, 2002] ainsi que la classification et la segmentation d'images [Manjunath et Chellappa, 1991, Wang et Liu, 1999, Hiremath *et al.*, 2012, Voisin, 2012].

Dans la terminologie des champs de Markov, une image composée de N pixels est représentée par un ensemble de N sites $S = \{S_1, \dots, S_N\}$. S est muni d'un système de voisinage V qui associe à chaque site $x \in S$ un ensemble de sites voisins $V_x \subset S$ et qui doit vérifier les propriétés suivantes :

$$x \notin V_x \quad , \quad (2.9)$$

$$\forall y \in S, x \in V_y \Leftrightarrow y \in V_x \quad . \quad (2.10)$$

En traitement d'images, le voisinage d'un site s correspond généralement à l'ensemble des pixels situés à une distance de s inférieure à un certain seuil. La taille du voisinage n'est cependant pas donnée par la valeur de ce seuil mais par un ordre de grandeur (voir figure 2.12). On parle alors de l'ordre du voisinage.

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

5	4	3	4	5
4	2	1	2	4
3	1	s	1	3
4	2	1	2	4
5	4	3	4	5

FIGURE 2.12 – Ordre (de grandeur) du voisinage pour un site s d'un champ de Markov. Le voisinage d'ordre n du site s est composé des sites dont la valeur associée est inférieure ou égale à n .

Des sous-ensembles de sites appelés cliques sont définis dans ce voisinage. Ils représentent les supports pour lesquels des motifs caractéristiques des textures seront par la suite recherchés. Ce sont les probabilités d'apparition de ces motifs qui seront ensuite utilisées afin d'exprimer la probabilité conditionnelle de réalisation de chacun des sites (voir figure 2.13).

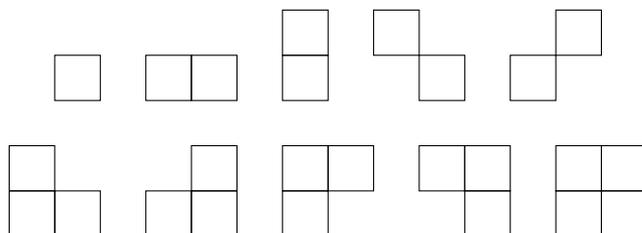


FIGURE 2.13 – Cliques de second ordre utilisées lors de la modélisation d'un phénomène par champs de Markov.

Chaque pixel de l'image étant vu comme une variable aléatoire, le fait qu'un pixel soit d'un niveau de gris donné peut être vu comme une réalisation de la variable associée. Chaque variable $X_s \in S$ prend donc ses valeurs dans un ensemble d'états Ω correspondant à l'ensemble des niveaux de gris. Dans le cas des images, nous avons généralement $|\Omega| = 256$. La probabilité de réalisation $x_s \in \Omega$ d'une variable aléatoire $X_s \in S$ se note alors $P(X_s = x_s)$.

La modélisation d'une texture par champs de Markov revient à exprimer la probabilité conditionnelle de réalisation de chaque site par rapport à son voisinage comme une loi de probabilité f dont la formulation aura préalablement été définie :

$$P(X_s = x_s | V_s) = f(X_s, x_s, V_s) \quad \forall X_s \in S, x_s \in \Omega \quad . \quad (2.11)$$

Cette modélisation s'appuie pour cela sur le théorème d'Hammersley-Clifford, qui établit que tout champs de Markov peut être associé à un champ de Gibbs [Besag, 1974]. À ce titre, la probabilité conditionnelle de réalisation d'un site s étant donné son voisinage V_s peut s'exprimer par la formule suivante :

$$P(X_s = x_s | V_s) = \frac{\exp(-U(X_s, x_s, V_s))}{\sum_{x \in \Omega} \exp(-U(X_s, x, V_s))} \quad , \quad (2.12)$$

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

avec $U(X_s, x, V_s)$ une fonction qui mesure l'énergie liée à la réalisation x d'un site X_s par rapport à son voisinage V_s . Cette énergie s'exprime généralement à l'aide d'une fonction δ_c qui mesure le potentiel de chaque clique c présente dans le voisinage V_s à caractériser la réalisation x de X_s :

$$U(X_s, x, V_s) = \sum_{c \in \text{Cliques}(V_s)} \delta_c(X_s, x) \quad . \quad (2.13)$$

Parmi les différentes méthodes de modélisation d'une texture par champs de Markov, nous pouvons citer le modèle de Derin-Elliott [Derin et Elliott, 1987] ainsi que le modèle auto-binomial [Besag, 1974]. Ces deux méthodes expriment de façon similaire le potentiel des cliques présentes dans le voisinage d'un site. Elles se basent sur la formulation suivante :

$$U(X_s, x, V_s) = \theta_1 \cdot w_1(X_s, x) + \theta_2 \cdot w_2(X_s, x) + \theta_3 \cdot w_3(X_s, x) + \theta_4 \cdot w_4(X_s, x) \quad , \quad (2.14)$$

le rôle de la fonction $w_r(X_s, x)$ étant de lier la présence de motifs (par le biais de cliques) à la probabilité de réalisation x du site X_s . Nous avons ainsi :

$$w_r(X_s, x) = \begin{cases} I(x, x_{s+r}) + I(x, x_{s-r}) & \text{pour le modèle de Durin-Elliott} \\ x \cdot (x_{s-r} + x_{s+r}) & \text{pour le modèle auto-binomial} \end{cases} \quad \forall r \in [1; 4] \quad , \quad (2.15)$$

avec :

$$I(a, b) = \begin{cases} -1 & \text{si } a = b \\ 1 & \text{sinon} \end{cases} \quad . \quad (2.16)$$

Les variables $\theta_1, \theta_2, \theta_3, \theta_4$ constituent les paramètres du modèle. Les probabilités $P(X_s = x_s | V_s)$ étant connues, la modélisation d'une texture par champs de Markov revient à estimer ces paramètres. Plusieurs méthodes permettent leur estimation, telles que les moindres-carrés [Zou *et al.*, 1995], le maximum de vraisemblance [Descombes *et al.*, 1997], le maximum *a posteriori* [Wu et Noé, 2010] ou encore le recuit simulé [Jeng et Woods, 1990]. Ces paramètres constituent une signature de la texture sur laquelle des algorithmes de classification ou de segmentation pourront se baser.

2.1.6 Bilan

Après cet état de l'art sur les méthodes de caractérisation de textures, il convient de se demander quel descripteur choisir. La littérature, relativement abondante à ce sujet, expose de nombreux avis contradictoires qui ne permettent pas d'identifier une méthode universelle.

D'après la littérature, les méthodes à base de champs de Markov semblent être d'assez mauvais candidats [Ohanian et Dubes, 1992, Clausi, 2001], tandis que les bonnes performances des caractéristiques d'Haralick sont régulièrement soulignées [Clausi, 2001, Howarth et Stefan, 2004]. Dans [Tang, 1998], les caractéristiques d'Haralick sont notamment reconnues comme plus performantes que les méthodes à base d'ondelettes en ce qui concerne la classification de textures. La conclusion inverse est cependant formulée dans [Bharati *et al.*, 2004]. Comparés à la transformée de Fourier, les filtres de Gabor se révèlent être plus

2.1. ÉTAT DE L'ART DES MÉTHODES DE CARACTÉRISATION DE TEXTURES

performants lorsqu'appliqués à une problématique d'indexation et de recherche d'images [Ursani *et al.*, 2007]. Néanmoins, la tendance est inversée en présence d'images bruitées. Les avis s'opposent concernant les performances des méthodes empruntées au traitement du signal [Clausi, 2001, Pichler *et al.*, 1996, Chen et Chen, 1999], mais s'accordent autour de l'importante complexité algorithmique de celles à base de bancs de filtres [Randen et Husoy, 1999, Manjunath et Ma, 1996b, Chen et Chen, 1999].

Dans [Stachowiak *et al.*, 2005], une étude visant à évaluer les performances de différents descripteurs de textures (caractéristiques d'Haralick, motifs binaires locaux, transformées en ondelettes et champs de Markov) sur différents types de textures est menée. Dans un contexte de classification, chaque descripteur se révèle être performant sur un type de texture, mais aucun ne l'est sur l'ensemble du panel de test. Les caractéristiques d'Haralick et la transformée en ondelettes offrent cependant une bonne complémentarité et leur combinaison permet d'assurer une indépendance vis à vis du contenu des images. Ces conclusions confirment celles déjà formulées dans le passé [Randen et Husoy, 1999, Sharma et Singh, 2001] : il n'existe pas de méthode de caractérisation de textures universelle et la combinaison de plusieurs descripteurs est nécessaire afin de permettre le traitement de tous types d'images.

La littérature s'accorde enfin sur un point, l'existence de deux types de textures : les microtextures, composées de motifs dont l'échelle est de l'ordre de quelques pixels, et les macrotextures, généralement composées de motifs plus larges [Davis *et al.*, 1979, Tuceryan et Jain, 1993, Unser, 1995, Wang *et al.*, 2013]. Les méthodes de caractérisation de textures qui reposent sur des supports de petite taille (motifs binaires locaux, champs de Markov) sont généralement reconnues comme peu efficaces sur les macrotextures car elles ne permettent pas d'appréhender correctement l'échelle de la texture. Pour la même raison, les méthodes empruntées au traitement du signal sont souvent peu adaptées aux microtextures.

Les méthodes que nous développons dans cette thèse sont indépendantes du descripteur de textures utilisé. L'objet de nos travaux n'étant pas l'évaluation des performances de ces descripteurs, nous avons choisi de nous limiter à seulement l'un d'entre eux. Notre choix s'est porté vers les caractéristiques d'Haralick pour plusieurs raisons. Tout d'abord, il s'agit d'un descripteur dont les bonnes performances sont régulièrement évoquées dans la littérature. C'est notamment l'un des rares descripteurs capable de traiter à la fois des micro-textures et des macro-textures. Il offre en effet la possibilité de régler précisément la taille de la région sur laquelle une matrice de co-occurrences est calculée ainsi que l'échelle à laquelle nous voulons qu'une texture soit analysée. Ensuite, il s'agit d'une méthode permettant d'obtenir directement un ensemble d'attributs numériques caractérisant une texture. Nous voulions en effet éviter d'avoir recours à des algorithmes de partitionnement de données, ce qui est requis par certaines méthodes qui imposent l'utilisation de dictionnaires de signatures. Enfin, l'un de nos objectifs étant de pouvoir traiter des images 3D, nous avons besoin d'une caractéristique de textures qui soit extensible à des dimensions supérieures, ce que les caractéristiques d'Haralick permettent nativement.

2.2 Contribution

L'algorithme de segmentation par régularisation de graphe sur lequel nous nous appuyons fait partie de la classe des méthodes d'optimisation globale pour la segmentation de données [Ta, 2009] présentée section 1.7.4.

Soit $G = (V, E, w)$ un graphe de similarité, et $f_0 \in \mathcal{H}(V)$ la fonction représentant les données à segmenter. Pour rappel, cette classe de modèles s'exprime sous la forme du problème de minimisation suivant :

$$\min_{f:V \rightarrow \{0,1\}} \left\{ \mathcal{E}_2(f, f_0, \lambda) = \mathcal{R}_{w,1} + \sum_{u \in V} \lambda g(f_0)(u) f(u) \right\} , \quad (2.17)$$

avec f la fonction indicatrice de la segmentation, $\mathcal{R}_{w,1}$ le terme de régularisation, $g(\cdot)$ le terme d'attache aux données et λ le coefficient exprimant le compromis entre la précision et la régularité de la solution.

Nous nous intéressons plus particulièrement à l'implémentation basée sur le critère de Chan et Vese [Chan et Vese, 2001], présenté section 1.7.4.2, dans laquelle le terme d'attache aux données s'exprime par :

$$g(f_0) = (c_1 - f_0)^2 - (c_2 - f_0)^2 \quad , \quad (2.18)$$

avec $c_1, c_2 \in \mathbb{R}$ deux paramètres représentant la valeur moyenne de la fonction f_0 respectivement à l'intérieur et à l'extérieur de la segmentation.

Cette méthode permet nativement de tirer parti d'informations non-locales caractérisant les pixels. À l'aide d'une mesure de similarité adaptée aux données manipulées, il est en effet possible de pondérer les relations relativement à ces données. Elles peuvent ainsi influencer indirectement sur le déroulement de l'algorithme. Ta applique notamment cette approche à la segmentation d'images texturées et bruitées, où la mesure de similarité est calculée dans l'espace Euclidien des patches d'images [Tschumperlé et Brun, 2009].

Les patches d'images ne sont pas des caractéristiques de textures à proprement parler, mais une construction permettant de comparer deux pixels en tenant compte des régions au milieu desquelles ils se trouvent. Soit $I : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}^m$ une image, potentiellement multivaluée. Le patch associé à un pixel de I est composé du pixel et de son voisinage, dont la taille est préalablement fixée. Il est ainsi possible de calculer la similarité des patches associés à deux pixels $p_1 = (x_1, y_1)$ et $p_2 = (x_2, y_2)$ en calculant la distance Euclidienne entre leurs patches respectifs :

$$\mu(p_1, p_2)^2 = \sum_{x=-v}^v \sum_{y=-v}^v G_a(x, y) \cdot \|I(x_1 + x, y_1 + y) - I(x_2 + x, y_2 + y)\|_2^2 \quad , \quad (2.19)$$

avec $v \in \mathbb{N}^+$ le rayon du voisinage, et G_a un noyau Gaussien de variance a , permettant de donner plus d'importance aux pixels centraux.

Les patches d'images ont été utilisés pour de la synthèse de textures [Efros et Leung, 1999] et plus récemment dans le cadre du filtrage et de la restauration d'images [Ta *et al.*, 2007a, Gilboa et Osher, 2007, Buades *et al.*, 2005, Bougleux *et al.*, 2008, Elmoataz *et al.*,

2008]. L'ensemble de ces travaux a mis en évidence l'importance des approches non-locales, qui permettent une meilleure prise en compte de la présence de motifs et de structures complexes dans les images.

Le critère d'attache aux données de Chan et Vese sur lequel repose la méthode de Ta ne profite cependant pas de ces informations non-locales, car il travaille uniquement à partir du niveau de gris de chacun des pixels. Plus particulièrement, son rôle est de minimiser l'écart-type de niveaux de gris de chacune des régions. Il s'agit donc de statistiques d'ordre 1, ce qui le rend incapable de discriminer deux textures visuellement différentes dont les niveaux de gris moyens seraient égaux (voir figure 2.1).

Cette limitation a cependant été étudiée, et il a été démontré qu'il est possible de remplacer l'intensité du niveau de gris de chaque pixel par une caractéristique non-locale évaluant l'agencement des niveaux de gris dans l'image, telle qu'une mesure de la courbure ou encore de l'orientation [Chan et Vese, 2001].

Cependant, la caractéristique de texture utilisée est unique. Il est donc nécessaire de disposer d'un certain *a priori* concernant l'image, afin de savoir quelle caractéristique employer. Afin de palier cela, nous avons opté pour une modification du critère d'attache aux données afin de profiter d'informations de plus haut niveau, disponibles sous la forme d'un vecteur de caractéristiques.

Nous nous sommes pour cela appuyés sur une extension du modèle de Chan et Vese qui permet le traitement d'images multivaluées [Chan *et al.*, 2000]. Le critère d'attache aux données qui y est défini tient compte de l'information présente dans l'ensemble des canaux de l'image, permettant ainsi la segmentation d'images couleurs ou multi-spectrales.

Soit $f_0 : \Omega \subset \mathbb{R}^m \rightarrow [0, 1]^n$ la fonction initiale représentant l'image. La fonctionnelle d'énergie sur laquelle se base le problème de minimisation s'exprime dans le domaine continu par :

$$\begin{aligned} \mathcal{J}_{CV}(f, f_0, \lambda_1, \lambda_2) = \int_{\Omega} \|\nabla f\|_1 + \sum_{i=1}^n \int_{\Omega} \left(\lambda_{1,i} \cdot f(x)(c_{1,i} - f_{0,i}(x))^2 \right. \\ \left. + \lambda_{2,i} \cdot (1 - f(x))(c_{2,i} - f_{0,i}(x))^2 \right) dx \quad , \end{aligned} \quad (2.20)$$

avec $c_{1,i}$ et $c_{2,i}$ les variables représentant les valeurs moyennes de chacun des n canaux de l'image f_0 respectivement à l'intérieur et à l'extérieur de la segmentation, et $\lambda_{1,i}, \lambda_{2,i} \in \mathbb{R}^+$ les paramètres permettant de moduler l'importance des différents termes de l'équation.

Cette formulation sous-entend que lors de l'utilisation d'un vecteur composé de k caractéristiques, le problème de minimisation possède $2k$ paramètres. De nombreuses caractéristiques de textures étant composées d'un nombre important de valeurs, un tel paramétrage du problème n'est pas envisageable, et une réduction du nombre de paramètres est nécessaire. À l'image de l'algorithme originel de Ta, nous avons tout d'abord opté pour la conservation d'un unique paramètre λ , et ainsi n'avoir à réaliser qu'un compromis entre régularité et précision de la solution. Néanmoins, les tests que nous avons réalisés nous ont permis de constater qu'un contrôle plus fin de l'algorithme était nécessaire, au risque de ne pas obtenir de résultats pertinents pour certaines images. D'après les tests réalisés dans [Chan *et al.*, 2000], deux options nous sont offertes.

La première consiste à utiliser un unique λ_i pour chaque canal i de l'image, avec

2.2. CONTRIBUTION

$\lambda_{1,i} = \lambda_{2,i} = \lambda_i$, ce qui permet de donner plus ou moins d'importance à certains canaux de l'image. Cependant, cela nécessite de nouveau de posséder un *a priori* concernant l'importance des différents canaux, information complexe à obtenir (voire inexistante) lors de l'utilisation de caractéristiques de textures. Par ailleurs, cette approche ne simplifie pas suffisamment le paramétrage du problème, puisque qu'il reste nécessaire de fixer autant de paramètres que de canaux.

La seconde option consiste à n'utiliser que deux paramètres, λ_1 et λ_2 , avec $\lambda_{1,1} = \dots = \lambda_{1,n} = \lambda_1$ et $\lambda_{2,1} = \dots = \lambda_{2,n} = \lambda_2$. À l'image de ce qui est fait dans l'algorithme de Chan et Vese scalaire, cela permet de donner plus de poids au terme d'attache aux données associé à l'une des deux régions. Nos tests nous ont permis de vérifier que la présence de ces deux paramètres était suffisante pour corriger les problèmes de pertinence des résultats liés à l'utilisation d'un unique paramètre. La fonctionnelle de régularisation sur laquelle nous nous basons peut donc être reformulée :

$$\begin{aligned} \mathcal{J}_{CV}(f, f_0, \lambda_1, \lambda_2) = \int_{\Omega} \|\nabla f\|_1 + \sum_{i=1}^n \int_{\Omega} \left(\lambda_1 \cdot f(x) \cdot (c_{1,i} - f_{0,i}(x))^2 \right. \\ \left. + \lambda_2 \cdot (1 - f(x)) \cdot (c_{2,i} - f_{0,i}(x))^2 \right) dx \quad , \end{aligned} \quad (2.21)$$

Dans cette formulation vectorielle du problème, le critère de Chan et Vese offre toujours une évaluation de l'homogénéité intra-région, mais cette mesure est désormais globalement calculée sur l'ensemble des canaux de l'image.

La procédure permettant de transformer le problème de minimisation représenté par la fonctionnelle d'énergie (2.21) en un problème d'optimisation globale s'exprimant par la seconde classe de modèles variationnels est identique à celle proposée par Ta (exposée section 1.7.4.2). En réécrivant le terme d'attache aux données présent dans l'équation (2.21) :

$$\begin{aligned} \sum_{i=1}^n \left(\lambda_1 \cdot f \cdot (c_{1,i} - f_{0,i})^2 + \lambda_2 \cdot (1 - f)(c_{2,i} - f_{0,i})^2 \right) \\ = f \cdot \sum_{i=1}^n \left(\lambda_1 \cdot (c_{1,i} - f_{0,i})^2 - \lambda_2 \cdot (c_{2,i} - f_{0,i})^2 \right) + \lambda_2 \sum_{i=1}^n (c_{2,i} - f_{0,i})^2 \quad , \end{aligned} \quad (2.22)$$

nous pouvons mettre en évidence le dernier terme qui, étant donné que les variables $c_{1,i}$ et $c_{2,i}$ sont calculées avant chaque itération de l'algorithme, est constant et n'influe donc pas sur le processus de minimisation de la fonctionnelle (2.21). Dans ces conditions, nous pouvons formuler un problème de minimisation équivalent à partir de la fonctionnelle suivante :

$$\begin{aligned} \mathcal{J}_{CV}(f, f_0, \lambda_1, \lambda_2) = \int_{\Omega} \|\nabla f\|_1 + \sum_{i=1}^n \int_{\Omega} f(x) \cdot \left(\lambda_1 \cdot (c_{1,i} - f_{0,i}(x))^2 \right. \\ \left. - \lambda_2 \cdot (c_{2,i} - f_{0,i}(x))^2 \right) dx \quad . \end{aligned} \quad (2.23)$$

Cette dernière constitue un cas particulier défini dans le domaine continu de la seconde classe de modèles variationnels (1.41) ; il est donc naturel de considérer la version discrète

2.3. RÉSULTATS

définie sur les graphes de ce modèle de Chan et Vese vectoriel :

$$\min_{f:V \rightarrow \{0,1\}} \left\{ \mathcal{E}_{CV}(f, f_0, \lambda_1, \lambda_2) = \mathcal{R}_{w,1} + \sum_{u \in V} g(f_0, \lambda_1, \lambda_2)(u) f(u) \right\}, \quad (2.24)$$

avec :

$$g(f_0, \lambda_1, \lambda_2) = \sum_{i=1}^n \left(\lambda_1 \cdot (c_{1,i} - f_{0,i})^2 - \lambda_2 \cdot (c_{2,i} - f_{0,i})^2 \right). \quad (2.25)$$

Algorithme 1 : Algorithme d'approximation du modèle de Chan et Vese pour images multi-valuées.

Entrées : un graphe de similarité $G = (V, E, w)$, une fonction $f_0 \in \mathcal{H}^m(V)$ représentant les caractéristiques des pixels, $m = V \rightarrow \{0, 1\}$ l'initialisation de la segmentation, N un nombre d'itérations à réaliser, $\lambda_1, \lambda_2 \in R^+$ les paramètres du problème de minimisation.

Sorties : une fonction $f : V \rightarrow \{0, 1\}$ indicatrice de la segmentation.

pour tous les $u \in V$ **faire** $f^0(u) = m(u)$;

pour $n = 0$ **a** $N - 1$ **faire**

pour $i = 1$ **a** m **faire**

$c_{1,i} =$ moyenne de $f_{0,i}$ lorsque $f^n \geq 0.5$

$c_{2,i} =$ moyenne de $f_{0,i}$ lorsque $f^n < 0.5$

pour tous les $(u, v) \in E$ **faire**

$\beta_{uv} = \sqrt{w_{uv}} |f^n(u) - f^n(v)|^{-1}$

pour tous les $u \in V$ **faire**

$$\gamma = \frac{\sum_{v \sim u} \beta_{uv} \cdot f^n(v) - \sum_{i=1}^m \left(\lambda_1 \cdot (c_{1,i} - f_{0,i}(u))^2 - \lambda_2 \cdot (c_{2,i} - f_{0,i}(u))^2 \right)}{\sum_{v \sim u} \beta_{uv}}$$

$f^{n+1}(u) = \max(\min(\gamma, 1), 0)$

pour tous les $u \in V$ **faire**

si $f^N(u) > 0.5$ **alors** $f(u) = 1$ **sinon** $f(u) = 0$;

L'algorithme 1 résume la méthode de régularisation basée sur le modèle de Chan et Vese vectoriel.

Si l'on considère une image scalaire comme une image multivaluée composée d'un unique canal, notre méthode est compatible avec celle de Ta.

2.3 Résultats

2.3.1 Segmentation d'images multispectrales

L'une des applications de l'algorithme de Chan et Vese vectoriel est l'imagerie multispectrale [Chan *et al.*, 2000]. Une image multispectrale correspond à une image dans laquelle

2.3. RÉSULTATS

une scène est acquise à l'aide de différents capteurs, chacun étant conçu pour quantifier une propriété particulière. L'image est alors composée de plusieurs canaux.

La technique d'imagerie multispectrale la plus commune est la photographie couleur, pour laquelle chaque canal de l'image caractérise une gamme de longueurs d'ondes à laquelle la vision humaine est sensible. De nombreux appareils de télémétrie tels que les satellites de télédétection ou encore les télescopes spatiaux sont capables d'acquérir des images à des longueurs d'ondes auxquelles l'œil humain n'est pas sensible, comme l'infrarouge ou l'ultraviolet. Certaines technologies d'imagerie médicale peuvent aussi caractériser simultanément plusieurs propriétés physiques. La microscopie à fluorescence permet par exemple d'exciter certains composés chimiques afin de leur faire émettre des photons et ainsi révéler leur présence.

L'intérêt du critère vectoriel d'attache aux données de Chan et Vese est de pouvoir combiner l'information présente dans chacun des canaux et ainsi de permettre le traitement de l'image malgré la présence de bruit ou d'occlusions dans certains d'entre eux.

Afin d'illustrer la capacité de notre méthode à traiter ce type d'image ainsi que le rôle des différents paramètres, nous allons dans un premier temps traiter l'image artificielle présentée figure 2.14.

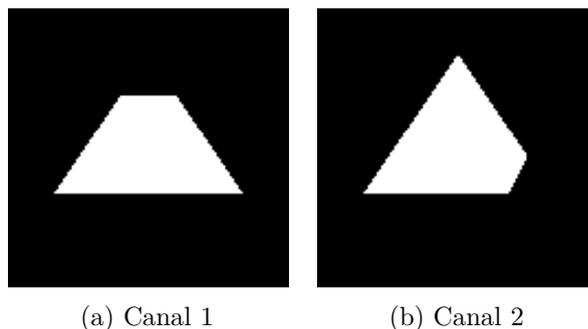


FIGURE 2.14 – Exemple d'image multispectrale. Les occlusions présentes dans chacun des canaux ne permettent pas de distinguer l'objet dans sa totalité.

Le critère d'attache aux données que nous employons (voir équation (2.25)) peut être reformulé ainsi :

$$\begin{aligned}
 g(f_0, \lambda_1, \lambda_2) &= \lambda_1 \cdot \sum_{i=1}^2 (c_{1,i} - f_{0,i})^2 - \lambda_2 \cdot \sum_{i=1}^2 (c_{2,i} - f_{0,i})^2 \quad , \\
 &= \lambda_1 \cdot d(c_1, f_0)^2 - \lambda_2 \cdot d(c_2, f_0)^2 \quad , \quad (2.26)
 \end{aligned}$$

avec f_0 le vecteur de caractéristiques associé à un nœud, c_1 et c_2 les vecteurs composés des valeurs moyennes des différents canaux de l'image pour chacune des deux régions identifiées par la segmentation et $d()$ l'opérateur distance euclidienne. Il s'agit donc d'une mise en opposition des distances entre les caractéristiques d'un nœud et les caractéristiques moyennes des deux régions. Pour un nœud particulier, lorsque ce critère prend une valeur négative (respectivement positive), cela signifie que du point de vue du critère d'attache aux données, ce nœud appartient à la région intérieure (respectivement extérieure).

2.3. RÉSULTATS

Les paramètres λ_1 et λ_2 permettent de pondérer cette mise en opposition en autorisant une des deux régions à présenter un écart à la moyenne plus important. Ainsi, si $\lambda_1 > \lambda_2$ (respectivement $\lambda_1 < \lambda_2$), l'homogénéité, relativement à la distance euclidienne, de la région intérieure (respectivement extérieure) sera privilégiée.

Dans l'algorithme 1, le terme d'attache aux données contrebalance le terme de régularisation. Ce dernier n'étant associé à aucun paramètre permettant de le pondérer, ce sont les valeurs de λ_1 et λ_2 qui jouent ce rôle : plus λ_1 et λ_2 seront faibles, plus la régularité de la solution sera privilégiée.

Afin de traiter l'image 2.14, nous avons construit un graphe grille en 4-voisinage. Aucune mesure de similarité n'a été utilisée : la valeur constante 1 a été affectée à l'ensemble des arêtes du graphe pour permettre au processus de diffusion de s'exécuter. Dans chacun de nos tests, l'algorithme sera initialisé dans l'état présenté figure 2.15a.

Afin d'identifier le triangle complet, il convient de régler les paramètres λ_1 et λ_2 de façon adéquate. Pour cette image, cette tâche se révèle être relativement simple. En effet, nous savons que dans la segmentation que nous souhaitons obtenir, la région extérieure ne devra être composée que de pixels noirs, tandis que – du fait de la présence d'occlusions – la région représentant le triangle devra contenir à la fois des pixels de faible et de forte intensité. L'homogénéité de la région extérieure doit donc être privilégiée : λ_2 devra avoir une valeur plus importante que λ_1 .

Concernant la régularité de la solution, la forme que nous cherchons à identifier présentant des angles francs, il est là encore évident que l'attache aux données devra être privilégiée. λ_1 et λ_2 ne devront donc pas être fixés à des valeurs trop faibles.

La segmentation du triangle est accomplie avec succès en fixant $\lambda_1 = 1$ et $\lambda_2 = 2$ (voir figure 2.15b). À l'inverse, en privilégiant l'attache aux données de la région intérieure ($\lambda_1 = 2$, $\lambda_2 = 1$), seule la région du triangle commune aux deux canaux sera identifiée (voir figure 2.15c). Si en revanche nous choisissons de ne pas privilégier une région par rapport à une autre, le terme de régularisation se charge d'établir un compromis entre la régularité et la précision de la solution (voir figure 2.15d). L'obtention d'un résultat stable nécessite alors un plus grand nombre d'itérations car la régularisation s'appuie sur un principe de diffusion. Enfin, si nous accordons moins d'importance au terme d'attache aux données, les contours du triangle perdent en précision et gagnent en régularité (voir figures 2.15e et 2.15f).

2.3.2 Segmentation d'images texturées artificielles

Dans ce second test, nous avons souhaité mettre en évidence la capacité de notre méthode à segmenter des images à partir de caractéristiques de textures. Le principe est similaire au traitement des images multispectrales, à la différence que chaque canal représente désormais une caractéristique de texture. Afin d'illustrer cela, nous nous intéressons à l'image de test présentée figure 2.16a.

Cette image résulte de l'addition d'un bruit blanc Gaussien d'écart-type $\sigma = 48$ à l'image 2.16b. Cette dernière est composée de deux motifs qui ne diffèrent que par leur orientation : hachures orientées à -45° pour l'arrière-plan et à $+45^\circ$ pour l'objet central. Ce bruit étant appliqué de façon homogène à toute l'image, la distribution des niveaux de gris

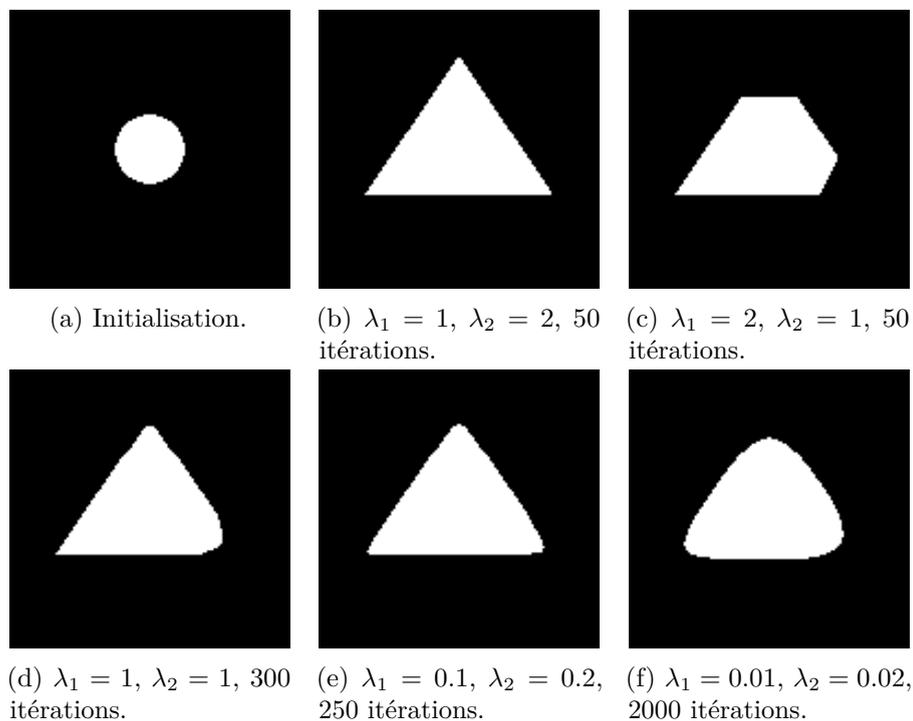


FIGURE 2.15 – (a) : initialisation de l’algorithme. (b, c, d, e, f) : segmentations obtenues après stabilisation du résultat.

dans chacune des deux régions reste homogène.

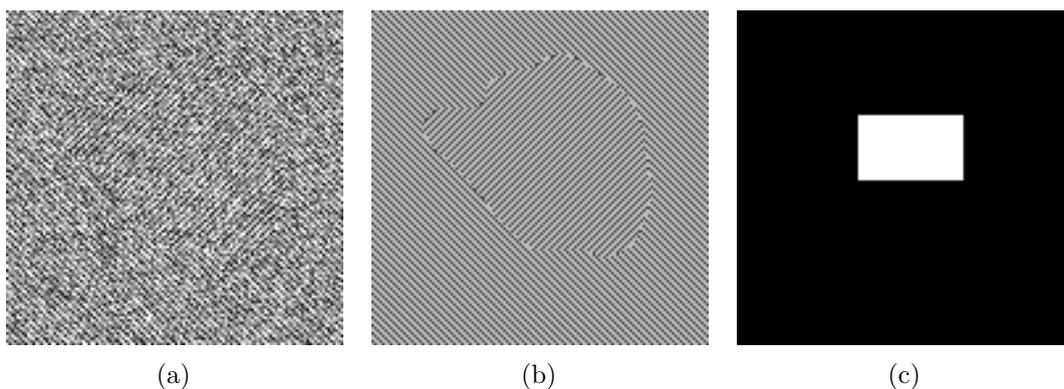


FIGURE 2.16 – Exemple d’image texturée artificielle. L’image (a) résulte de l’addition d’un bruit blanc Gaussien d’écart-type $\sigma = 48$ à l’image (b). Les statistiques de premier ordre concernant la distribution des niveaux de gris de l’objet central et de l’arrière-plan sont identiques. L’image (c) correspond à la segmentation initiale qui servira de point de départ de l’algorithme dans chacun des tests.

En choisissant de travailler avec des textures dont les statistiques de premier ordre sont identiques, nous excluons tout algorithme qui ne se baserait que sur une analyse brute

de la distribution des niveaux de gris. Ceux basés sur le critère d'attache aux données de Chan et Vese (dans sa version scalaire) sont les premiers visés car incapables de différencier les deux régions à partir de leurs niveaux de gris moyens respectifs. Il est donc nécessaire d'avoir recours à des caractéristiques de textures.

Les caractéristiques d'Haralick [Haralick *et al.*, 1973] ont été calculées en chaque pixel de cette image dans une région de 11 pixels de diamètre et selon le décalage (1, 1), l'image ayant été préalablement réduite à 16 niveaux de gris. Ce décalage a été choisi car il permet de cibler une propriété évidente de la texture : son orientation. Le décalage correspond en effet à l'orientation du motif de la région intérieure. Comme illustré figure 2.17, la matrice de co-occurrences associée à un pixel de cette région est majoritairement composée de co-occurrences de niveaux de gris similaires, ce qui se traduit par la présence de fortes valeurs à proximité de sa première diagonale. La matrice associée à un pixel de la région extérieure présente en revanche un nombre important de co-occurrences de niveaux de gris d'intensité différentes, ce qui contribue à la présence de valeurs importantes à proximité de la seconde diagonale. Ces deux structures de matrices sont reflétées par plusieurs caractéristiques d'Haralick, nous permettant de différencier les deux régions de l'image.

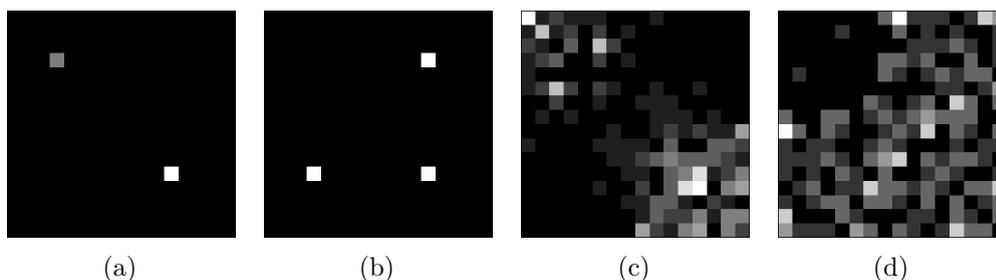


FIGURE 2.17 – Structures typiques des matrices de co-occurrences des deux régions de l'image texturée artificielle. (a) : région intérieure de l'image non bruitée. (b) : région extérieure de l'image non bruitée. (c) : région intérieure de l'image bruitée. (d) : région extérieure de l'image bruitée. La région intérieure est essentiellement composée de co-occurrences d'intensités similaires, tandis que la région extérieure présente un nombre important de co-occurrences d'intensités différentes.

Une représentation des caractéristiques d'Haralick normalisées correspondant à cette image est présentée figure 2.18.

Influence des paramètres λ_1 et λ_2 sur la qualité du résultat

Dans un premier temps, nous avons souhaité mettre en évidence le rôle important des deux paramètres λ_1 et λ_2 . Nous avons donc segmenté l'image 2.16a pour différentes valeurs de ces paramètres tout en suivant l'évolution du résultat au cours de l'exécution du processus de régularisation. Les valeurs de λ_1 et λ_2 considérées ont été prises dans l'ensemble $\{0.5, 0.75, \dots, 3\}$, lequel échantillonne un intervalle raisonnable de valeurs de compromis. Le graphe utilisé pour représenter l'image correspond à un 4-voisinage, et la constante 1 fait office de mesure de similarité.

Les tableaux 2.1 et 2.2 résument les résultats obtenus après convergence de l'algorithme

2.3. RÉSULTATS

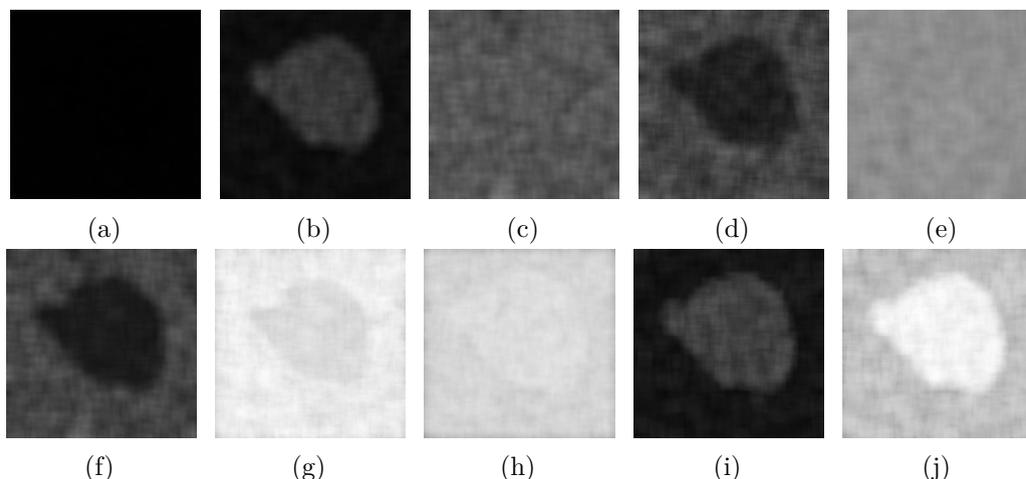


FIGURE 2.18 – Représentation normalisée des caractéristiques d’Haralick obtenues pour chaque pixel de l’image 2.16a. Paramètres de calcul : nombre de niveaux de gris = 16, diamètre de la fenêtre = 11 pixels, décalage = (1, 1).

pour l’ensemble des configurations. L’axe horizontal représente la valeur attribuée à λ_1 et l’axe vertical celle attribuée à λ_2 . Chaque tableau présente une mesure d’erreur qui évalue la qualité de la segmentation calculée par l’algorithme vis à vis de la vérité terrain.

La mesure d’erreur utilisée dans le premier tableau est la distance générique d’anomalie, qui est définie pour les problèmes de segmentation binaire comme le nombre de pixels mal classés [Cardoso et Corte-Real, 2005]. Elle est ici exprimée sous forme normalisée, et représente alors le pourcentage de pixels mal classés dans l’image.

Le second tableau est quant à lui dédié à une mesure d’erreur basée sur la distance de Hausdorff. Celle-ci évalue la distance maximum entre le contour de l’objet détecté par l’algorithme et celui défini par la vérité terrain. Cette mesure d’erreur peut être vue comme une évaluation de la plus forte erreur commise par l’algorithme de segmentation, c’est à dire la distance à laquelle se trouve le pixel mal classé le plus éloigné du contour. Cette mesure d’erreur est relativement sensible, car il suffit d’un unique pixel mal classé pour attribuer une erreur importante au résultat.

Ces deux tableaux nous montrent que pour cette image, les bons résultats sont obtenus lorsque le rapport λ_1/λ_2 est légèrement supérieur à 1. Cela signifie que le critère d’attache aux données autorise une plus grande variabilité en ce qui concerne les textures de la région extérieure, privilégiant ainsi l’homogénéité de la région intérieure.

En revanche, nous pouvons voir que, selon la distance de Hausdorff, les meilleures solutions sont obtenues lorsque le rapport λ_1/λ_2 est légèrement plus élevé que celui des meilleurs solutions selon la distance générique d’anomalie.

Ce comportement est à attribuer au processus de régularisation, dont l’une des conséquences est de lisser le contour des régions. Or, lorsqu’un contour présente une forte courbure, sa régularisation entraîne une augmentation de l’écart avec le contour défini par les données (voir figure 2.19). L’effet est d’autant plus prononcé que la courbure est importante. Cet écart correspond précisément à ce que mesure la distance de Hausdorff.

2.3. RÉSULTATS

$\lambda_2 \downarrow / \lambda_1 \rightarrow$	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00
0,50	2,37	4,14	5,22	6,07	6,81	7,39	7,98	8,44	8,86	9,36	9,74
0,75	1,84	1,52	2,34	3,09	3,69	4,16	4,57	4,89	5,29	5,59	5,85
1,00	3,35	1,54	1,23	1,49	2,11	2,47	2,94	3,20	3,43	3,83	3,97
1,25	4,71	2,40	1,46	1,22	1,29	1,45	1,82	2,16	2,36	2,61	2,86
1,50	5,87	3,25	1,97	1,41	1,16	1,09	1,25	1,40	1,62	1,86	2,12
1,75	6,79	4,00	2,66	1,74	1,36	1,15	1,07	1,11	1,20	1,34	1,53
2,00	7,88	4,69	3,14	2,26	1,62	1,35	1,12	1,06	1,06	1,12	1,17
2,25	9,05	5,22	3,70	2,69	2,00	1,57	1,36	1,13	1,04	1,01	1,06
2,50	10,37	5,82	4,15	3,05	2,41	1,83	1,53	1,34	1,14	1,02	0,99
2,75	12,62	6,23	4,52	3,47	2,73	2,22	1,73	1,53	1,34	1,14	1,06
3,00	18,98	6,75	4,91	3,80	3,03	2,48	2,07	1,68	1,46	1,34	1,15

TABLE 2.1 – Évaluation de l’influence des paramètres λ_1 et λ_2 . Les valeurs exposées correspondent à l’erreur mesurée selon la distance générique d’anomalie normalisée pour l’ensemble des combinaisons testées. Ces résultats sont calculés à partir des segmentations obtenues après convergence de l’algorithme.

$\lambda_2 \downarrow / \lambda_1 \rightarrow$	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00
0,50	6,3	8,2	9,5	10,8	11,7	12,4	13,0	13,6	14,6	14,9	15,5
0,75	5,4	4,2	5,4	6,1	6,7	7,3	7,8	8,2	8,6	9,2	9,5
1,00	6,3	5,0	4,1	4,0	4,5	5,0	5,4	5,7	5,8	6,4	6,4
1,25	7,3	5,4	5,0	4,1	3,6	3,6	4,0	4,2	4,2	4,5	5,0
1,50	7,6	6,1	5,4	5,0	4,1	3,6	3,6	3,2	3,6	3,6	4,0
1,75	8,1	6,3	5,4	5,0	4,5	4,1	3,6	3,6	3,6	3,0	3,0
2,00	8,9	6,3	6,1	5,4	5,0	4,5	4,1	3,6	3,6	3,6	3,2
2,25	9,0	7,3	6,1	5,4	5,0	5,0	4,1	4,1	3,6	3,6	3,6
2,50	10,8	7,3	6,3	5,4	5,4	5,0	4,5	4,1	4,1	3,6	3,6
2,75	14,8	7,6	6,3	6,1	5,4	5,0	5,0	4,5	4,1	4,1	3,6
3,00	25,0	7,6	6,3	6,1	5,4	5,4	5,0	5,0	4,1	4,1	4,1

TABLE 2.2 – Évaluation de l’influence des paramètres λ_1 et λ_2 . Les valeurs exposées correspondent à l’erreur mesurée selon la distance de Hausdorff pour l’ensemble des combinaisons testées. Ces résultats sont calculés à partir des segmentations obtenues après convergence de l’algorithme.

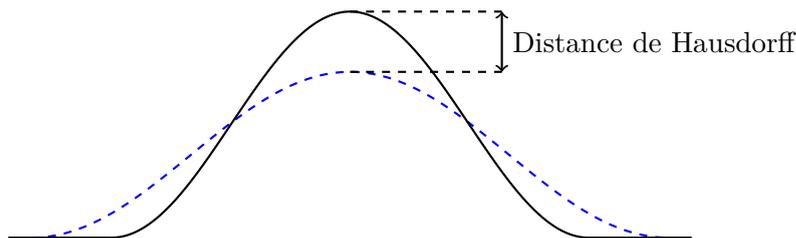


FIGURE 2.19 – Régularisation d’un contour de forte courbure. La courbe noire représente le contour initial (tel que défini par les données) et la courbe bleue en pointillés sa version régularisée.

2.3. RÉSULTATS

Nous pouvons illustrer cela sur la région supérieure gauche de l'image 2.16, dont la vérité terrain est représentée figure 2.20a. Les figures 2.20b et 2.20c représentent les meilleures solutions obtenues pour $\lambda_1 = 3$ (dernières colonnes des tableaux de résultats) avec respectivement la distance générique d'anomalie ($\lambda_2 = 2.5$) et la distance de Hausdorff ($\lambda_2 = 1.75$). Les figures 2.20d et 2.20e représentent les différences entre chacune des solutions et la vérité terrain. Les pixels noirs correspondent à ceux qui ont été ajoutés par la régularisation, et les blancs à ceux qui ont été retirés.

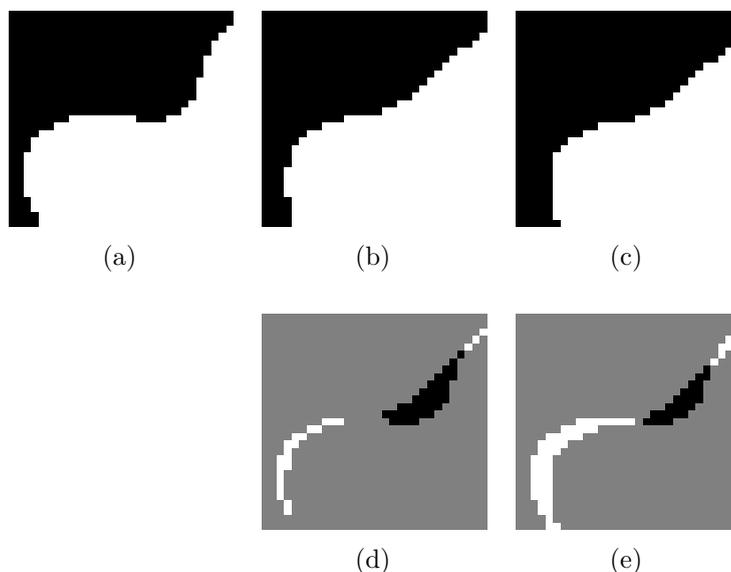


FIGURE 2.20 – Illustration de l'influence de la régularisation sur les contours de forte courbure. Les images présentées ici se focalisent sur la région supérieure gauche de l'image 2.16. La figure (a) représente la vérité terrain et les figures (b) et (c) les segmentations obtenues par notre méthode pour $\lambda_1 = 3$ et respectivement $\lambda_2 = 2.5$ et $\lambda_2 = 1.75$. Les images (d) et (e) correspondent à la différence entre la solution calculée et la vérité terrain : les pixels noirs représentent ceux qui ont été supprimés, et les blancs ceux qui ont été ajoutés.

Lorsque $\lambda_2 = 2.5$, le rapport λ_1/λ_2 est proche de 1, ce qui se traduit par une régularisation relativement importante. Cette solution étant la meilleure du point de vue de la distance générique d'anomalie, les pixels sont globalement bien classés, exceptés dans la région concave (les pixels en noir dans la figure 2.20d) qui subit une forte régularisation.

Lorsque $\lambda_2 = 1.75$, nous autorisons l'algorithme à mal classer un plus grand nombre de pixels de l'arrière-plan (les pixels blancs dans l'image 2.20e) et à moins régulariser la solution. Comparé à la solution précédente, cela se traduit par une augmentation globale du nombre de pixels mal classés, sauf dans la région concave où des pixels avaient été ajoutés.

Dans ces deux configurations, les pixels les plus éloignés du vrai contour se trouvent précisément dans cette région concave. Les paramètres λ_1 et λ_2 jouent donc un rôle important en ce qui concerne la qualité du résultat. Cette notion de qualité dépend cependant du critère d'évaluation, et un même jeu de paramètres ne permettra pas nécessairement de satisfaire plusieurs critères.

Influence des paramètres λ_1 et λ_2 sur la vitesse de convergence

Nous pouvons aussi noter que ces paramètres influent sur la vitesse de convergence de l'algorithme. Les valeurs contenues dans le tableau 2.3 nous indiquent que la méthode converge d'autant plus rapidement que les paramètres sont fixés à des valeurs fortes.

$\lambda_2 \downarrow / \lambda_1 \rightarrow$	1	1,25	1,5	1,75	2	2,25	2,5	2,75	3
1	1200	1200	1200	1200	1200	1200	1200	1200	1200
1,25	1200	1000	1000	1000	1000	1000	1000	1000	1000
1,5	1200	1000	900	900	900	900	900	900	900
1,75	1200	1000	900	900	900	900	900	900	900
2	1200	1000	900	900	700	700	700	700	700
2,25	1200	1000	900	900	700	700	700	700	700
2,5	1200	1000	900	900	700	700	600	600	600
2,75	1200	1000	900	900	700	700	600	600	600
3	1200	1000	900	900	700	700	600	600	600

TABLE 2.3 – Influence des paramètres λ_1 et λ_2 sur la vitesse de convergence. Les valeurs indiquées représentent le nombre d'itérations nécessaires à l'obtention d'une solution stable.

Le terme d'attache aux données réalise une classification qui se base sur des critères d'homogénéité. Ceux-ci sont calculés de façon globale par rapport à l'image. En une itération de l'algorithme, il est donc possible d'attribuer une classe à chacun des nœuds du graphe. La régularisation est en revanche réalisée par un processus de diffusion dont le principe est d'attribuer à chaque nœud une valeur indiquant le potentiel d'appartenance à chacune des deux classes. Ce potentiel est calculé en un nœud comme étant la moyenne du potentiel de chacun des nœuds voisins. La régularisation se déroule ainsi de proche en proche : il est donc nécessaire d'effectuer un certain nombre d'itérations afin de permettre au potentiel d'un nœud d'influer sur celui d'un autre nœud qui ne serait pas un voisin direct.

Afin d'illustrer la raison pour laquelle cette méthode nécessite un plus grand nombre d'itérations lorsque l'un des paramètres est de faible valeur, considérons $G = (V, E, w)$ un graphe de similarité, $f_0 \in \mathcal{H}(V)$ une fonction caractérisant les nœuds et u et v deux nœuds dont les caractéristiques sont typiques respectivement des régions 1 et 2 ($f_0(u) = c_1$ et $f_0(v) = c_2$). Le critère d'attache aux données :

$$g(f_0, \lambda_1, \lambda_2) = \lambda_1(c_1 - f_0)^2 - \lambda_2(c_2 - f_0)^2 \quad , \quad (2.27)$$

vaut alors pour chacun de ces nœuds :

$$g(f_0(u), \lambda_1, \lambda_2) = -\lambda_2(c_2 - c_1)^2 \quad , \quad (2.28)$$

$$g(f_0(v), \lambda_1, \lambda_2) = \lambda_1(c_1 - c_2)^2 \quad . \quad (2.29)$$

Ces formules nous indiquent que si $\lambda_1 < \lambda_2$, alors le critère d'attache aux données aura plus de poids pour le nœud u que pour le nœud v . Les nœuds qui, du point de vue des données, appartiennent à la première classe subissent ainsi plus fortement les effets de la régularisation que ceux qui appartiennent à la seconde. Un plus grand nombre d'itérations est alors nécessaire afin de stabiliser le potentiel de ces nœuds.

2.3. RÉSULTATS

Influence de la pertinence des caractéristiques

Nous pouvons voir dans la figure 2.18 que parmi les dix caractéristiques d'Haralick dont nous disposons, seulement la moitié d'entre elles contient une information utile à la segmentation de l'image. Or, nous savons que le critère de Chan et Vese considère équitablement chacune des caractéristiques de textures. Il est donc envisageable que des caractéristiques non pertinentes ou encore bruitées perturbent l'identification des textures et rendent la méthode moins précise.

Nous avons donc souhaité évaluer la capacité de notre méthode à traiter des données non pertinentes. Nous avons pour cela réduit à cinq le nombre de caractéristiques en ne conservant que celles qui, visuellement, sont les plus pertinentes. Elles correspondent à celles labellisées respectivement (b), (d), (f), (i) et (j) dans la figure 2.18. Le même test que précédemment a ensuite été réalisé à partir de ces données.

Le tableau 2.4 présente les résultats obtenus, évalués selon la distance de générique d'anomalie.

$\lambda_2 \downarrow / \lambda_1 \rightarrow$	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00
0,50	1,26	2,03	2,70	3,30	3,84	4,16	4,55	4,86	5,08	5,33	5,62
0,75	1,97	1,19	1,25	1,53	1,93	2,31	2,61	2,85	3,11	3,36	3,52
1,00	3,06	1,65	1,14	1,04	1,15	1,37	1,57	1,80	2,08	2,25	2,46
1,25	4,01	2,38	1,56	1,15	1,03	1,01	1,13	1,20	1,40	1,53	1,70
1,50	4,72	2,94	2,06	1,50	1,18	1,04	0,99	1,02	1,10	1,14	1,29
1,75	5,29	3,52	2,54	1,83	1,44	1,21	1,07	1,00	0,96	0,99	1,07
2,00	5,87	3,97	2,93	2,25	1,73	1,45	1,26	1,10	1,04	0,99	0,99
2,25	6,30	4,38	3,35	2,62	2,09	1,68	1,44	1,26	1,15	1,04	0,99
2,50	6,81	4,73	3,58	2,92	2,42	1,98	1,68	1,44	1,26	1,15	1,10
2,75	7,18	5,04	3,89	3,20	2,67	2,21	1,84	1,66	1,43	1,28	1,17
3,00	7,50	5,27	4,17	3,45	2,94	2,51	2,10	1,81	1,61	1,45	1,29

TABLE 2.4 – Évaluation de la capacité à gérer des caractéristiques de faible pertinence. Les valeurs exposées correspondent à l'erreur mesurée selon la distance générique d'anomalie pour l'ensemble des combinaisons testées. Ces résultats sont calculés à partir des segmentations obtenues après convergence de l'algorithme.

Ces résultats nous montrent une augmentation sensible de la qualité de segmentation pour l'ensemble des jeux de paramètres. À titre indicatif, 30 résultats sont désormais sous le seuil de 1.20, contre seulement 21 lorsque l'ensemble des caractéristiques était utilisé.

Comme l'indique les résultats présentés dans le tableau 2.5, la vitesse de convergence est, elle aussi, améliorée : le nombre d'itérations est globalement réduit de 40 %.

Une étape de sélection des caractéristiques les plus pertinentes est donc nécessaire afin d'optimiser la modélisation du problème et permettre une meilleure identification des textures. Nous reviendrons dans la conclusion de ce chapitre sur une méthode permettant de la rendre automatique.

2.3. RÉSULTATS

$\lambda_2 \downarrow / \lambda_1 \rightarrow$	1	1,25	1,5	1,75	2	2,25	2,5	2,75	3
1	800	700	700	700	700	700	700	700	700
1,25	800	700	700	700	700	700	700	700	700
1,5	700	600	600	500	500	500	500	500	500
1,75	700	600	600	500	500	500	500	500	500
2	700	600	600	500	500	500	500	500	500
2,25	700	600	600	500	500	400	400	400	400
2,5	700	600	600	500	500	400	400	400	400
2,75	700	600	600	500	500	400	400	400	400
3	700	600	600	500	500	400	400	400	400

TABLE 2.5 – Évaluation de la capacité à gérer des caractéristiques de faible pertinence. Les valeurs indiquées représentent le nombre d’itérations nécessaires à l’obtention d’une solution stable.

2.3.3 Segmentation d’images échographiques 2D et 3D

Le dernier test concerne le traitement d’images échographiques. L’échographie est une technique d’imagerie médicale réputée pour produire des images particulièrement bruitées.

Ce bruit est en réalité ce que l’on appelle un motif d’interférence. Le fonctionnement d’un échographe est basé sur l’émission d’un signal sonore et l’analyse de ses échos. L’amplitude de ces échos est représentatrice de l’échogénicité des milieux que le signal traverse, c’est à dire leur capacité à renvoyer un écho. Chaque tissu ou organe possède un niveau d’échogénicité qui lui est propre, ce qui permet de les distinguer. En fonction des différents éléments que rencontre l’onde sur son passage, plusieurs échos peuvent se superposer et interférer, produisant ainsi des pics de fortes ou de faibles amplitudes, propres à la composition du milieu. Ils ne représentent donc pas un bruit mais forment des motifs dont l’analyse permet une meilleure caractérisation des différents milieux. Ces images sont donc de parfaites candidates à une analyse basée texture.

Nous avons validé notre méthode sur un jeu de six images échographiques. Ces acquisitions ont été réalisées sur l’épiderme et permettent l’analyse de névus, plus communément appelés grains de beauté. Nous disposons pour chaque image d’une vérité terrain, définie par un expert, qui nous permet d’évaluer la précision des segmentations produites.

Chaque image est représentée par un graphe de similarité. Sa structure correspond à un 4-voisinage, et la constante 1 a été utilisée en tant que mesure de similarité. Les caractéristiques d’Haralick ont été calculées en réduisant le nombre de niveaux de gris à 16, pour une fenêtre de diamètre 11 et pour le décalage (1, 0). Seules les trois caractéristiques d’Haralick les plus pertinentes ont été conservées.

Nous avons comparé notre méthode à l’algorithme de segmentation par régularisation de graphe originel [Ta, 2009] ainsi qu’à l’approche par contours actifs de Chan et Vese, dans sa version scalaire [Chan et Vese, 2001] ainsi que vectorielle [Chan *et al.*, 2000]. Pour cette dernière, nous avons employé les mêmes caractéristiques d’Haralick que pour notre méthode.

Les paramètres optimaux des différentes méthodes ont été déterminés selon un principe d’optimisation aléatoire [Matyas, 1965] visant à obtenir la distance générique d’anomalie la plus faible.

2.3. RÉSULTATS

Le tableau 2.6 résume les résultats obtenus pour chacun de ces tests. Ceux-ci montrent une amélioration notable de la précision de segmentation, en particulier par rapport à la méthode de Ta dont nous nous inspirons et qui se révèle parfois incapable de produire un résultat exploitable.

Image	Notre méthode	Méthode de C&V scalaire	Méthode de C&V vectorielle (basée textures)	Méthode de Ta
echo1	1,82	2,41	3,45	8,31
echo2	2,02	1,59	1,25	N.A.
echo3	1,86	7,16	3,26	2,45
echo4	2,47	3,77	3,34	4,90
echo5	2,62	3,39	3,62	10,69
echo6	2,16	3,07	3,14	3,40
Moyenne	2.2	6.0	3.6	3.0
Écart type	0.3	3.1	1.8	0.8

TABLE 2.6 – Comparaison de la précision de segmentation des différentes méthodes sur un ensemble d’images échographiques 2D : distance générique d’anomalie.

Nous constatons aussi que notre méthode présente de meilleurs résultats que les méthodes de segmentation par contours actifs, y compris celle basée textures. Nous pouvons donc affirmer que la combinaison d’une méthode de segmentation par régularisation de graphes et d’un critère d’attache aux données basé textures permet une meilleure identifications des contours des objets à segmenter.

Les figures 2.21 et 2.22 illustrent les résultats obtenus pour deux des images de tests pour chacune des méthodes évaluées.

Comme évoqué dans le chapitre 1, l’un des avantages des approches par graphes est de permettre à un même algorithme de traiter différents types de données, notamment de dimensions différentes. Afin d’illustrer cela, nous avons appliqué notre méthode à la segmentation d’une image échographique 3D.

L’implémentation de cette méthode reste identique, seul le graphe doit être adapté aux données manipulées. Nous avons donc étendu à la 3D la notion de grille que nous utilisions dans les précédents tests, permettant de créer des 6-voisinages. Le calcul des caractéristiques d’Haralick a lui aussi été adapté afin de supporter des fenêtre volumiques ainsi que des décalages exprimés dans les trois dimensions de l’espace.

L’image *echo6* dont il est question dans le tableau 2.6 est une coupe 2D extraite de cette image 3D. Il n’a cependant pas été possible de valider numériquement la qualité de l’intégralité du résultat, la création de vérités terrains pour des données volumiques étant un processus relativement laborieux. Seule une confirmation visuelle de la précision du résultat global a donc pû être obtenue.

La figure 2.23 illustre l’image sur laquelle notre méthode a été appliquée ainsi que des représentations volumiques du maillage ainsi calculé.

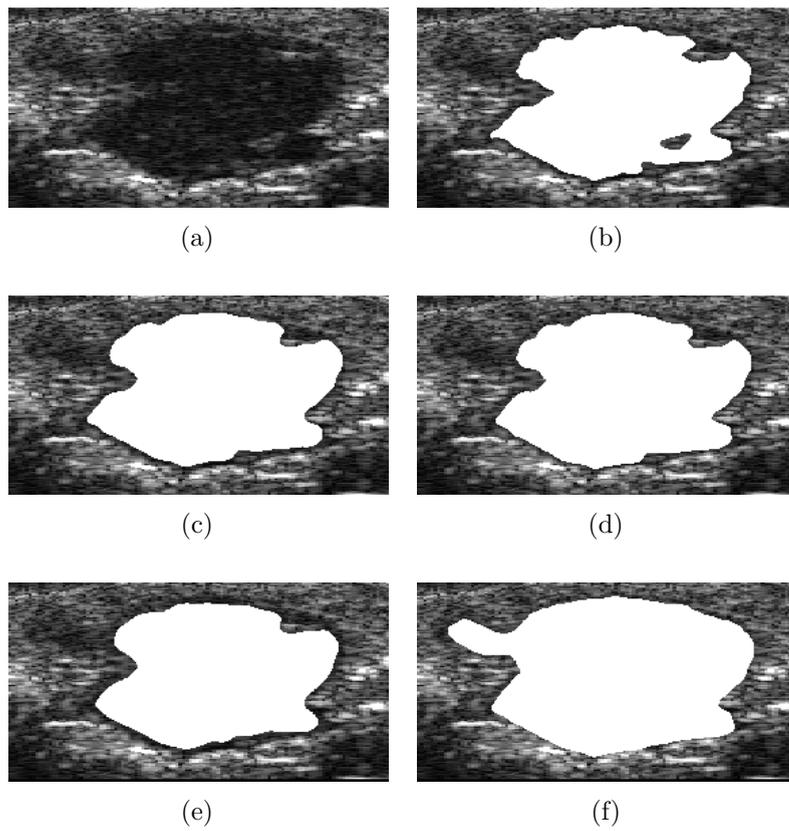


FIGURE 2.21 – Image *echo1*. (a) : image originale. (b) : vérité terrain. Résultats obtenus avec : (c) notre méthode, (d) la méthode de Chan & Vese scalaire, (e) la méthode de Chan & Vese vectorielle, (f) la méthode de Ta.

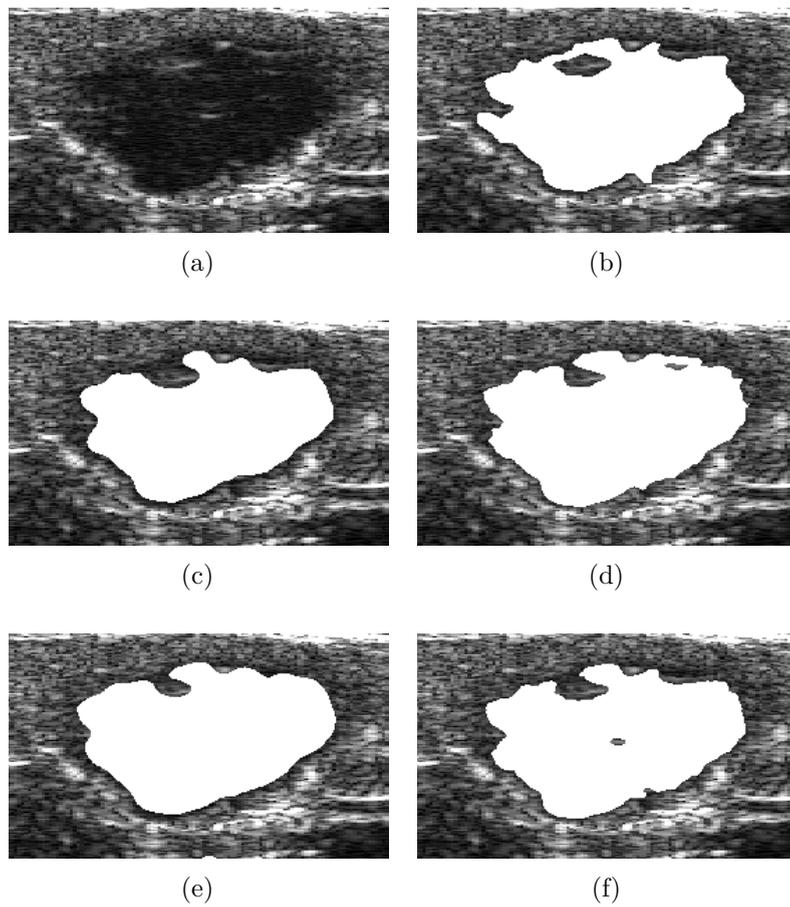
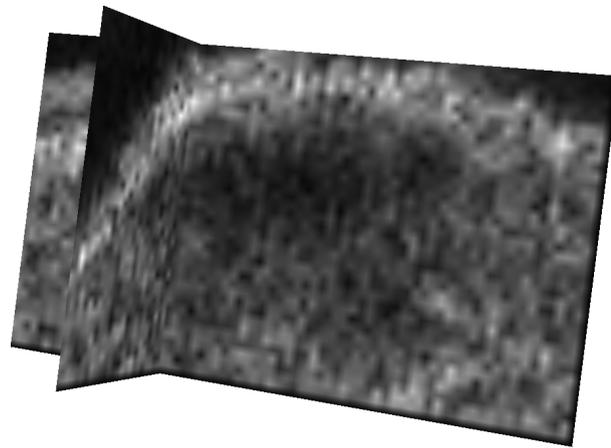
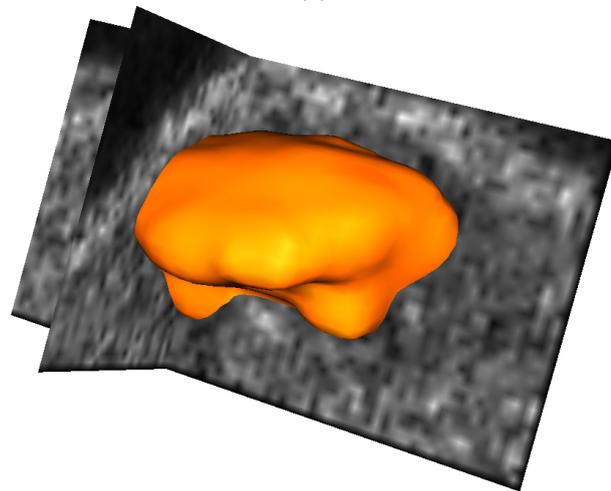


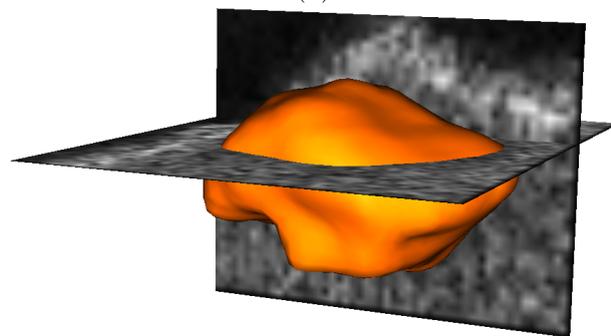
FIGURE 2.22 – Image *echo5*. (a) : image originale. (b) : vérité terrain. Résultats obtenus avec : (c) notre méthode, (d) la méthode de Chan & Vese scalaire, (e) la méthode de Chan & Vese vectorielle, (f) la méthode de Ta.



(a)



(b)



(c)

FIGURE 2.23 – Illustration des résultats obtenus sur une image échographique 3D. (a) : image initiale. (b) et (c) : représentation volumique du maillage calculé.

2.4 Conclusion

La méthode que nous avons présentée dans ce chapitre apporte à un récent algorithme de segmentation d'images par régularisation de graphes la capacité de tirer parti de caractéristiques de textures. Les différents tests réalisés ont permis d'illustrer le fonctionnement de l'algorithme ainsi que l'intérêt des approches non locales. Nous avons tout de même entrevu certaines limites de notre méthode.

La première concerne la détermination des paramètres λ_1 et λ_2 , dont les valeurs peuvent influencer grandement sur la qualité des résultats. Mis à part dans les cas extrêmement simples, il est relativement complexe de prédire quelle région doit voir son homogénéité privilégiée. Il est donc inenvisageable de les fixer automatiquement.

La seconde limite découle d'une simplification que nous avons choisi d'opérer sur le critère d'attache aux données. Afin de conserver un nombre raisonnable de paramètres, nous ne permettons pas de pondérer individuellement chaque caractéristique de textures. Nous avons montré que notre méthode est capable de fonctionner malgré un certain nombre de caractéristiques non pertinentes, mais cela se répercute sur les performances de l'algorithme, tant en précision qu'en vitesse d'exécution.

Comme nous avons pu le voir dans l'état de l'art des méthodes de caractérisation de textures réalisé en début de ce chapitre, aucune méthode existante ne permet d'obtenir une caractérisation « universelle » qui serait pertinente sur tous types de textures. Afin de remédier à cela, l'une des solutions est de combiner plusieurs descripteurs. Nous ne pouvons cependant pas avoir d'assurance concernant la pertinence de chacun d'entre eux, et un trop grand nombre de descripteurs non pertinents pourrait avoir un effet désastreux sur la capacité de notre méthode à résoudre certains problèmes de segmentation. Une sélection manuelle des caractéristiques les plus pertinentes reste donc nécessaire, ce qui réduit une éventuelle automatisation de la méthode.

Ces deux problèmes découlent de l'incapacité du terme d'attache aux données à considérer les caractéristiques de textures en fonction de leur pertinence. Afin de remédier à cela, nous proposons dans le prochain chapitre d'intégrer un classificateur supervisé au processus de segmentation. Celui-ci aura pour rôle de fournir une modélisation plus aboutie des textures à identifier, permettant ainsi l'expression d'un terme d'attache aux données plus précis.

2.4. CONCLUSION

Chapitre 3

Segmentation d'images texturées par régularisation de classificateurs

Les tests réalisés dans le chapitre précédent montrent qu'un réglage fin des paramètres de notre méthode de segmentation par régularisation de graphes est nécessaire afin d'optimiser sa précision. Procéder à ces réglages requiert cependant de posséder un certain *a priori* concernant les caractéristiques de textures utilisées. Malheureusement, celui-ci n'est pas nécessairement disponible ou exploitable et, sous réserve de l'obtention d'un tel *a priori*, trouver les paramètres optimaux reste – comme le montrent nos tests – une opération complexe pour laquelle il n'existe actuellement aucune méthode non-supervisée.

Nous avons par ailleurs vu qu'une sélection des caractéristiques les plus pertinentes permet d'optimiser la méthode, aussi bien en ce qui concerne la qualité des résultats que la vitesse de convergence. Pour des raisons pratiques – exposées section 2.2 – nous avons réduit à deux le nombre de paramètres requis par notre méthode¹. Il devient donc impossible d'inhiber « algorithmiquement » certaines caractéristiques de textures en leur attribuant un poids nul ; seule une modification des données d'entrée permet d'y parvenir.

Les travaux que nous présentons dans ce chapitre visent à corriger ces lacunes tout en conservant les deux approches non-locales sur lesquelles se base notre précédente méthode : les caractéristiques de textures et la régularisation de graphes. Afin d'y parvenir, nous devons permettre au seul élément en relation avec les caractéristiques de textures, le terme d'attache aux données, de prendre en compte la pertinence de chacune des caractéristiques.

Une solution consisterait à revenir au critère d'attache aux données de Chan et Vese pour images multispectrales [Chan *et al.*, 2000]. Or, alors que le nombre de canaux dans une image multispectrale est généralement faible, le nombre de caractéristiques de textures que nous sommes amenés à considérer peut être relativement important. Il est donc inenvisageable d'aboutir manuellement à un paramétrage optimal de ce critère.

Un problème de segmentation d'images peut être vu comme un problème de classifi-

1. L'algorithme de Chan et Vese pour images multivaluées originel requiert deux paramètres pour chaque caractéristique de textures.

cation de données. L'objectif est en effet d'attribuer une classe à chacun des pixels d'une image d'après un vecteur de caractéristiques. Dans la méthode que nous avons présentée précédemment, le terme d'attache aux données est l'élément qui a pour rôle d'indiquer si un vecteur de caractéristiques de textures est typique de l'objet à identifier ou bien de l'arrière-plan. Son rôle est donc similaire à celui d'un classificateur binaire. Nous pouvons d'ailleurs noter que Ta évoque un lien entre le modèle de Chan et Vese (dans sa version scalaire) et l'algorithme des k-moyennes [Ta, 2009].

Afin de permettre une meilleure modélisation du problème, nous proposons de nous baser sur un classificateur supervisé. L'idée est alors d'intégrer la réponse de ce classificateur au terme d'attache aux données.

Nous commencerons ce chapitre par un état de l'art sur la classification de données, en s'intéressant plus particulièrement aux méthodes supervisées. Nous détaillerons ensuite notre approche, puis terminerons par une présentation des résultats obtenus sur différents types d'images : textures artificielles et images médicales 2D et 3D.

Les résultats de ces travaux ont été publiés dans [Faucheux *et al.*, 2013].

3.1 Classification supervisée de données

La classification de données est une branche de l'apprentissage automatique. Ce domaine vise à produire des « machines » qui, à partir d'un jeu de données caractéristique d'un système, sont capables d'inférer un ensemble de règles visant à modéliser le système dans son ensemble. De manière générale, cette modélisation permet d'associer un label à chaque individu de l'espace considéré.

Ces techniques s'opposent donc aux systèmes experts, des outils d'aide à la décision conçus pour appliquer un ensemble de règles à de nouveaux jeux de données. À la différence des méthodes d'apprentissage automatique, les systèmes experts requièrent que les règles aient au préalable été fixées par un expert humain, selon un formalisme spécifique à l'outil. L'intérêt des systèmes experts a cependant décliné avec la généralisation des outils informatiques : ces derniers permettant d'obtenir facilement de grands volumes de données, il est devenu impossible pour un expert humain de fournir une modélisation pertinente d'un système complexe.

Les méthodes d'apprentissage automatique se divisent en plusieurs catégories qui identifient le mode d'apprentissage. Les trois principales sont : l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage semi-supervisé. Nous allons voir que lorsque les données sont représentées à l'aide de graphes, ou encore que la modélisation du système est soumise à certaines contraintes, il est alors possible d'établir un pont entre certaines de ces catégories.

En apprentissage supervisé, un ensemble d'apprentissage constitué de n individus et de leurs labels respectifs est requis. Cet ensemble, choisi pour être le plus représentatif possible des différentes classes que l'on souhaite identifier, est utilisé pour entraîner l'algorithme et générer la modélisation du problème. En comptant sur la capacité du modèle à généraliser ce qu'il a assimilé durant la phase d'apprentissage, celui-ci peut ensuite être appliqué à tout nouvel individu d'un ensemble de test afin de lui associer son propre label.

3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

Un algorithme d'apprentissage est qualifié de non-supervisé lorsque son objectif est de détecter des structures cachées dans un ensemble de test. Puisque ces structures sont *a priori* inconnues, l'algorithme se base sur des notions de proximité et d'homogénéité afin d'identifier des groupes composés des données les plus similaires.

L'apprentissage semi-supervisé consiste à utiliser à la fois des données labellisées et non-labellisées pour entraîner l'algorithme. Afin de compléter l'information contenue dans l'ensemble d'apprentissage, des notions de régularité ou de proximité sont utilisées. La labellisation automatique de certaines données – qui viendront s'ajouter à l'ensemble d'apprentissage initial – permet en particulier de rendre un algorithme supervisé plus robuste. Cette technique minimise en effet l'éventualité d'un sur-apprentissage lorsque les données labellisées ne sont disponibles qu'en faible quantité [Zhu, 2005, Chapelle et Schölkopf, 2006].

Certaines méthodes d'apprentissage supervisé peuvent être vues comme des problèmes d'interpolation : des marqueurs initiaux sont fournis pour un ensemble d'individus qui doivent être utilisés pour calculer le label d'individus « intermédiaires ». Le problème est alors résolu par diffusion des labels présents dans l'ensemble d'apprentissage aux individus non-labellisés. Cette diffusion se déroule de proche en proche, en suivant les règles de voisinage imposées par la structure des données.

L'une des hypothèses sur lesquelles s'appuie l'apprentissage semi-supervisé est que, dans un espace de grande dimension, les données reposent en réalité dans une variété de faible dimension. Cela s'illustre aisément à l'aide d'un graphe de similarité : les nœuds similaires sont généralement situés dans des sous-graphes présentant une forte connexité (voir figure 3.1). La topologie du graphe constitue ainsi une représentation implicite de ces variétés, permettant à un processus de diffusion de se dérouler selon un principe géodésique (en suivant la structure des données) et non plus selon un principe Euclidien. L'algorithme est donc capable de tirer parti des relations qui ont été établies avec des données potentiellement non labellisées pour mener à bien sa tâche. Pour cette raison, les méthodes d'apprentissage automatique qui se basent sur des graphes sont souvent qualifiées de semi-supervisées.

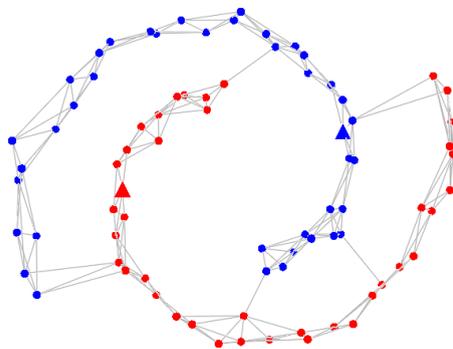


FIGURE 3.1 – Illustration de la notion d'apprentissage semi-supervisé dans un graphe. Ce graphe de similarité est un graphe des 5-plus proches voisins. Les nœuds en forme de triangle composent l'ensemble d'apprentissage. Les relations de voisinage mettent en évidence la structure des données, ce dont un algorithme d'apprentissage pourra tirer parti afin d'identifier les deux spirales.

Une autre hypothèse de l'apprentissage semi-supervisé concerne la régularité des labels : si deux données sont proches dans l'espace d'entrée, alors leurs labels doivent être similaires. Étant donné que les algorithmes de régularisation imposent ce type de contraintes, ils peuvent eux aussi être qualifiés de semi-supervisés.

L'approche que nous avons présentée dans le chapitre précédent repose à la fois sur l'utilisation de graphes pour représenter les données et sur une méthode de régularisation. Elle entre donc pleinement dans la catégorie des algorithmes semi-supervisés. Nous pouvons cependant voir un lien avec une quatrième catégorie d'algorithmes, moins répandue : l'apprentissage partiellement supervisé [Ambroise et Govaert, 2000]. Ces algorithmes ont la particularité de pouvoir travailler à partir d'informations incomplètes, par exemple « l'individu x n'appartient pas à la classe A ». Notre méthode de segmentation par régularisation de graphes entre dans cette catégorie car l'ensemble d'apprentissage que nous définissons ne concerne qu'une seule classe. Le critère d'attache aux données de Chan et Vese ne dispose donc initialement que d'une modélisation partielle du problème, mais celle-ci est affinée au fil des itérations.

Dans l'approche que nous proposons dans ce chapitre, nous nous appuyons sur un classificateur afin de modéliser les textures à identifier.

Les méthodes non supervisées ont été exclues puisqu'elles se basent sur des critères d'homogénéité ou de densité pour détecter les différentes classes. Or, les données que nous souhaitons manipuler en premier lieu sont des images, dans lesquelles les données sont réparties dans l'espace avec une densité constante. De plus, nous employons des caractéristiques de textures qui, du fait de leur non-localité, induisent un flou dans l'espace de représentation des données. Il est ainsi relativement complexe d'exprimer des critères d'homogénéité.

En traitement d'images, les données sont généralement disponibles en quantité suffisante pour permettre la construction d'un ensemble d'apprentissage pertinent. Nous n'avons donc pas jugé nécessaire de recourir à des approches semi-supervisées.

Nous avons ainsi opté pour l'utilisation d'un classificateur supervisé. Dans la suite de cette section, nous présentons les trois classificateurs supervisés les plus populaires : la méthode des k plus proches voisins, les réseaux de neurones artificiels et les machines à vecteurs supports. Nous pouvons d'ores et déjà noter qu'aucun de ces classificateur ne tire parti de la structure de graphe sur laquelle repose le processus de régularisation. Ces classificateurs ne peuvent donc pas être vus comme des approches semi-supervisées.

3.1.1 Méthode des k plus proches voisins

La méthode des k plus proches voisins² est l'un des algorithmes de classification de données les plus simples [Fix et Hodges, 1951]. Cette méthode ne repose pas sur un algorithme d'apprentissage à proprement parler, mais sur une heuristique qui réalise une classification par analogie : la classe attribuée à un individu inconnu correspond à la classe des individus connus les plus similaires. Plus précisément, il s'agit d'appliquer le principe du vote majoritaire à partir des k individus de l'ensemble d'apprentissage les plus proches.

Il n'y a donc pas de phase d'apprentissage, tous les calculs sont effectués lors la phase

2. Souvent abrégée k -PPV, ou k -NN, de l'anglais *k-nearest neighbors*.

3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

de classification. Pour chaque individu à classer, il est nécessaire de calculer la distance entre cet individu et ceux présents dans l'ensemble d'apprentissage. Plusieurs mesures de distances sont couramment employées : distance Euclidienne ou de Manhattan pour les données numériques [Weijer et Schmid, 2006], distance de Hamming pour les données symboliques [Smith *et al.*, 1994] ou encore distance de Mahalanobis lorsque l'on est en présence de données bruitées [Yates *et al.*, 2003].

Étant dépourvu de phase d'apprentissage, ce classificateur est apprécié pour sa capacité à assimiler instantanément de nouvelles connaissances. Cependant, derrière l'apparente simplicité de cette méthode, des précautions doivent être prises quant à la constitution de l'ensemble d'apprentissage.

Tout d'abord, il est important de disposer de données dont les attributs sont normalisés. Les mesures de distance traditionnelles ne permettent pas de tenir compte de l'amplitude des valeurs selon chacune des dimensions. Appliqué à notre problématique, cela signifie qu'une caractéristique de textures présentant de faibles variations pourrait être masquée par une caractéristique de plus forte amplitude.

De plus, les individus présents dans cet ensemble doivent être identiquement distribués dans leur espace. En effet, étant donné que seuls les k plus proches voisins sont considérés, il y a un risque d'introduire un biais dans la classification si une classe est statistiquement plus représentée qu'une autre.

Enfin, lorsque l'espace des individus est composé d'un nombre important de dimensions, les mesures de distance perdent de leur intérêt [Beyer et Goldstein, 1999]. Cette méthode est ainsi sujette à ce qui est couramment appelé la « malédiction de la dimension ». Afin de remédier à cela, il convient d'utiliser des techniques de sélection de caractéristiques, telles que l'analyse en composantes principales. Cependant, cela revient à la mise en place d'une phase d'apprentissage.

3.1.2 Réseaux de neurones artificiels

Un neurone biologique est un élément de base du système nerveux. Il s'agit d'une cellule qui peut être excitée par une impulsion nerveuse et qui, en fonction de son excitation, est capable d'émettre de nouvelles impulsions. Les neurones sont présents en grande quantité dans un cerveau, et sont impliqués dans l'essentiel des fonctions cognitives d'un être vivant. Le cerveau fonctionne en effet selon une approche connexionniste : des processus complexes émergent de réseaux d'unités fonctionnelles simples interconnectées (les neurones).

Les réseaux de neurones artificiels constituent une classe d'outils de modélisation inspirés du fonctionnement du cerveau. Introduits dans [Rosenblatt, 1958], le principe est d'interconnecter de nombreuses unités de calcul simples, appelées neurones formels, afin de permettre la résolution de problèmes complexes.

Présenté dans [Rumelhart *et al.*, 1986], l'un des modèles de réseaux de neurones les plus populaires est le perceptron multicouche³, connu pour être un approximateur universel de fonctions [Hornik *et al.*, 1989, Hartman *et al.*, 1990]. Il s'agit d'un réseau à propagation avant composé de plusieurs couches de neurones de type perceptron. Chaque neurone représente une unité de calcul capable de résoudre un problème de séparation linéaire,

3. Aussi appelé MLP, de l'anglais *Multi Layer Perceptron*.

3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

et reçoit en entrée l'ensemble des valeurs calculées par les neurones de la couche qui le précède (sauf la couche d'entrée qui reçoit les données du problème). Cette interconnexion de neurones permet la résolution de problèmes non-linéaires. La figure 3.2 ainsi que les formules (3.1) et (3.2) décrivent le fonctionnement d'un perceptron d'indice j :

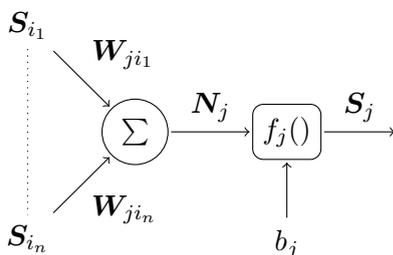


FIGURE 3.2 – Schéma d'un neurone formel j de type perceptron.

$$N_j = \sum_{i \in I} S_i \cdot W_{ji} \quad (3.1)$$

$$S_j = f_j(N_j - b_j) \quad (3.2)$$

avec $I = \{S_1 \dots S_n\}$ l'ensemble des entrées de ce neurone, $\{W_{j1} \dots W_{jn}\}$ les poids associés aux entrées, b_j un biais et $f_j()$ une fonction de transfert, généralement une fonction de type sigmoïde ou log-sigmoïde. S_j représente la sortie de ce neurone.

Un MLP est ainsi constitué de plusieurs couches. La première est appelée couche d'entrée et reçoit les données du problème, tandis que la dernière est appelée couche de sortie et contient la réponse du réseau. Les couches intermédiaires sont appelées couches cachées. Chaque couche d'un MLP est fortement connectée à la couche précédente : les entrées d'un neurone correspondent aux sorties de ceux de la couche précédente⁴. Le nombre de neurones sur la couche d'entrée est égal à la dimension du problème, tandis que le nombre de neurones de sortie peut varier en fonction de sa modélisation. Le nombre de couches et de neurones cachés est à fixer en fonction de la complexité du problème. La figure 3.3 schématise la structure d'un MLP composé de quatre entrées, une couche cachée de cinq neurones et d'une unique sortie.

Un réseau de neurones artificiels nécessite un apprentissage afin de résoudre un problème particulier. Cette phase revient à fixer les poids et les biais de chacun des neurones. L'algorithme de rétropropagation du gradient [Rumelhart *et al.*, 1986], la première méthode permettant d'entraîner un perceptron multicouche, est encore largement utilisée aujourd'hui. Elle consiste à présenter chaque individu de l'ensemble d'apprentissage au réseau de neurones. Les valeurs obtenues en sortie sont ensuite comparées à celles attendues et les poids et biais de chaque neurone sont mis à jour selon leur influence sur l'erreur. Ce processus est exécuté itérativement jusqu'à ce que la modélisation soit suffisamment vraisemblable.

Il est ainsi courant de diviser l'ensemble des données disponibles pour l'apprentissage en deux sous-ensembles : l'ensemble d'apprentissage et l'ensemble de validation. Le premier

4. Sauf les neurones de la couche d'entrée qui reçoivent les données du problème.

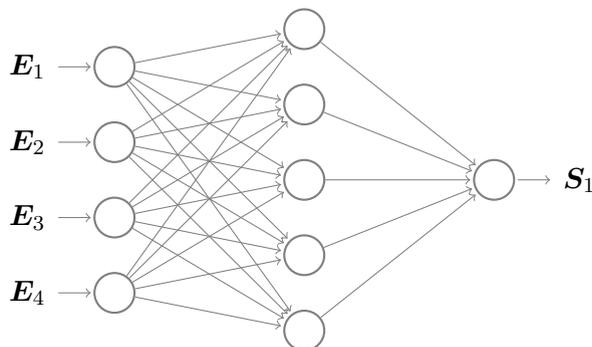


FIGURE 3.3 – Structure d’un perceptron multicouche composé de quatre entrées, une couche cachée de cinq neurones et d’une unique sortie.

reste utilisé pour entraîner le réseau alors que le second permet de mesurer l’aptitude du réseau à généraliser et ainsi de prévenir le sur-apprentissage. Un troisième sous-ensemble, l’ensemble de test, peut aussi être utilisé afin de choisir le nombre de couches et de neurones cachés que doit comporter le réseau pour modéliser au mieux et le plus efficacement le problème.

3.1.3 Machines à vecteurs supports

Les machines à vecteurs supports⁵ constituent l’une des plus récentes approches de la classification supervisée de données. Elles sont notamment reconnues pour leur bonnes performances vis à vis du traitement de problèmes non linéairement séparables [Suykens, 2001]. Ce résultat s’oppose cependant à l’une des théories sur lesquelles cette approche repose : la notion de marge maximale.

Étant donné un jeu de données composé de deux classes d’éléments et un algorithme chargé de calculer la frontière séparant les données, la marge est définie comme la distance minimale qui existe entre la frontière de séparation et les données les plus proches (voir figure 3.4). La maximisation de cette marge constitue une heuristique dont le but est de réduire l’erreur de généralisation de l’algorithme d’apprentissage en charge du calcul de la frontière : plus la marge est importante, moins l’algorithme a de chances de mal classer un individu inconnu.

Dans une machine à vecteurs supports, cette notion de marge maximale est appliquée au calcul d’un hyperplan séparateur. Soit \mathbb{R}^n l’espace de définition des données. Un hyperplan de cet espace correspond à l’ensemble des $\mathbf{x} \in \mathbb{R}^n$ tel que :

$$\mathbf{w}^T \cdot \mathbf{x} + w_0 = 0 \quad , \quad (3.3)$$

avec $\mathbf{w} = (w_1 \dots w_n)$ et w_0 ses paramètres.

Soit $X = \{(\mathbf{x}_1, c_1) \dots (\mathbf{x}_p, c_p)\}$ un ensemble d’apprentissage avec $c_k \in \{-1, +1\}$ la classe d’appartenance de l’individu \mathbf{x}_k . Le calcul de l’hyperplan de marge minimale est

5. Aussi appelées SVM, de l’anglais *Support Vector Machines*. Par analogie avec l’acronyme anglais, le terme Séparateurs à Vaste Marge est parfois employé.

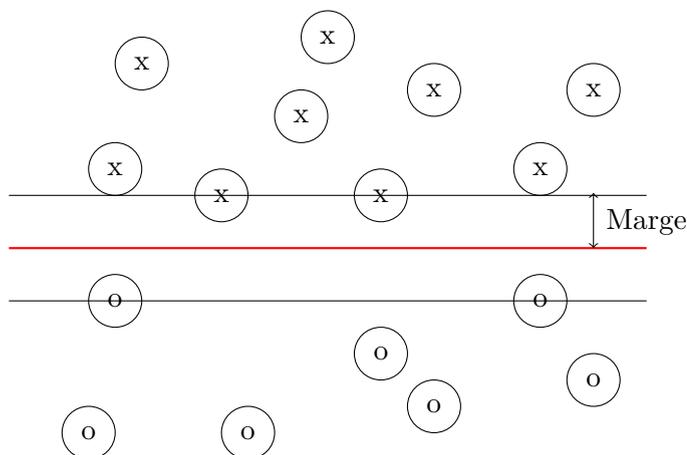


FIGURE 3.4 – Principe de marge maximale. Toute frontière de séparation présentera une marge inférieure à celle représentée par la ligne rouge.

obtenu par résolution du problème suivant :

$$\arg \max_{\mathbf{w}, w_0} \left\{ \frac{1}{\|\mathbf{w}\|} \min_k \left(c_k (\mathbf{w}^T \cdot \mathbf{x}_k + w_0) \right) \right\} . \quad (3.4)$$

Il s'agit d'un problème d'optimisation quadratique que de nombreuses méthodes permettent de résoudre. Nous pouvons notamment citer celles basées sur les algorithmes génétiques [Pai et Hong, 2005, Zhang *et al.*, 2009] ou encore les méthodes de recuit simulé [Lin *et al.*, 2008, Zhang *et al.*, 2009].

Après calcul de l'hyperplan séparateur, la classification d'un nouvel individu est obtenue en mesurant sa distance à l'hyperplan, par projection orthogonale. Le signe de cette distance est alors indicatif de la classification.

Dans cette configuration, l'hyperplan n'est calculé qu'à partir des données les plus proches de la frontière. Il suffit donc d'une donnée bruitée ou erronée à proximité de la frontière pour décaler l'hyperplan séparateur, réduisant ainsi la capacité de généralisation du classificateur. Afin de rendre les machines à vecteurs supports plus robustes, une configuration relaxée du problème appelée « marges souples » fut proposée [Cortes et Vapnik, 1995]. Celle-ci autorise certains éléments de l'ensemble d'apprentissage à se trouver à une distance de l'hyperplan inférieure à la marge, ou même à être mal classés.

La frontière étant un hyperplan, l'approche par marge maximale ne peut être appliquée avec succès qu'à des problèmes linéairement séparables. L'originalité des machines à vecteurs supports est due à Vapnik qui proposa de remédier à cette limitation en projetant les données d'entrée dans un espace de plus grande dimension [Vapnik, 1995]. Comme l'illustre la figure 3.5, cette opération peut en effet permettre de rendre linéairement séparable un problème qui ne l'est pas dans son espace d'origine. Cet opérateur de projection s'appelle une fonction noyau.

À l'inverse de nombreux algorithmes de classification qui tentent de calculer une frontière non linéaire, ce sont ici les données qui sont modifiées afin de permettre la résolution d'un problème « plus simple ». Cette innovation n'est cependant pas sans défaut puisqu'en

3.1. CLASSIFICATION SUPERVISÉE DE DONNÉES

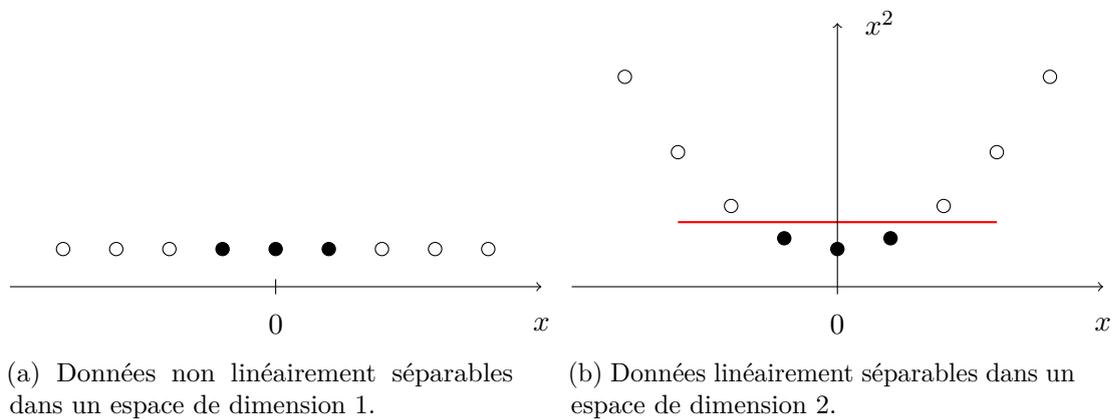


FIGURE 3.5 – Exemple de problème qui n'est linéairement séparable que lorsque les données sont projetées dans un espace de plus grande dimension.

l'absence d'*a priori* concernant les données, il est nécessaire de recourir à des fonctions noyau qui maximisent la probabilité de rendre le problème linéairement séparable. L'espace dans lequel sont projetées les données peut ainsi être de très grande dimension, rendant le problème insolvable. La solution consiste à appliquer « l'astuce du noyau » (*kernel trick*) [Aizerman *et al.*, 1964]. Pour des fonctions noyau respectant certaines propriétés, cette astuce rend inutile l'étape de projection, réduisant ainsi considérablement la complexité des opérations.

3.1.4 Bilan des méthodes de classification supervisée

La méthode des k plus proches voisins est souvent choisie pour sa simplicité de mise en œuvre. Nous ne l'avons cependant pas retenue pour plusieurs raisons, essentiellement liées à l'absence de phase d'apprentissage capable de synthétiser les données.

Le but recherché par l'ajout d'un classificateur à notre méthode de segmentation est de permettre une pondération automatique des différentes caractéristiques de textures. L'algorithme des k plus proches voisins étant uniquement basé sur une comparaison de mesures de distance, il est, au même titre que le critère d'attache aux données de Chan et Vese, incapable d'identifier la pertinence des différentes caractéristiques de textures et donc de les pondérer. Nous sommes par ailleurs amenés à travailler avec un nombre important de caractéristiques de textures, ce qui est un contexte défavorable aux k -PPV puisque ce classificateur est sujet à la malédiction de la dimension. Nous pouvons aussi noter que la classification d'un individu est un processus relativement long, puisque cela nécessite de calculer sa distance à chacun des éléments de l'ensemble d'apprentissage.

La littérature cite régulièrement les mérites des machines à vecteurs supports, qui sont notamment reconnues comme sensiblement plus performantes que les réseaux de neurones, tant en termes de précision que de rapidité d'exécution [Dal Moro *et al.*, 2006, Kumar *et al.*, 2011]. L'objectif de nos travaux n'étant pas l'évaluation des performances de classificateurs, nous nous sommes tournés vers celui qui nous est le plus familier : les réseaux de neurones artificiels. Ce choix nous permet notamment de limiter les risques liés à une mauvaise

configuration des classificateurs, et ainsi de nous focaliser sur leur intégration dans le processus de régularisation. La méthode que nous proposons reste compatible avec l'ensemble des classificateurs supervisés connu ce jour. Des travaux visant à permettre l'utilisation de classificateurs différents sont actuellement menés.

3.2 Contribution

Dans la méthode de segmentation d'images par régularisation de graphes présentée au chapitre précédent, le terme d'attache aux données de Chan et Vese fait office de classificateur binaire. En employant un classificateur supervisé tel que ceux présentés précédemment, nous sommes en mesure de gérer un nombre n arbitraire de classes. La méthode de segmentation que nous présentons ici a donc été conçue dans le but de permettre une segmentation multiclassée. Nous distinguerons cependant le cas où le nombre de classes est égal à deux du cas où il est supérieur à deux.

Toutes les images que nous allons traiter à l'aide de cette méthode doivent être converties sous forme de graphes de similarité selon l'une des techniques décrites section 1.3.

Nous considérons qu'il est possible d'obtenir – pour chaque objet que nous souhaitons identifier dans une image – un ensemble de nœuds dont les attributs de textures sont représentatifs de la texture des objets. Ces p nœuds constituent un ensemble d'apprentissage $X = \{(\mathbf{x}_1, c_1) \dots (\mathbf{x}_p, c_p)\}$, avec $\mathbf{x}_k \in \mathbb{R}^m$ un vecteur de caractéristiques de textures et $c_k \in [c_1 \dots c_n]$ un symbole qui identifie la classe associée.

3.2.1 Segmentation biclasse

Nous nous plaçons ici dans le cadre où, à partir de caractéristiques de textures, nous souhaitons localiser dans une image un unique objet. Celui-ci se distingue donc de l'arrière-plan, d'où la présence de deux classes que nous nommerons c_1 (objet) et c_2 (arrière-plan).

À partir de l'ensemble d'apprentissage, nous sommes en mesure d'entraîner un classificateur. Celui-ci se formalise par une règle de décision $f(\mathbf{x})$ qui associe à tout vecteur de caractéristiques de textures $\mathbf{x} \in \mathbb{R}^m$ la probabilité d'appartenance à la première classe :

$$f(\mathbf{x}) = p(c_1|\mathbf{x}) \quad , \quad (3.5)$$

Les classes c_1 et c_2 étant complémentaires, la probabilité d'appartenance de \mathbf{x} à la seconde classe est donnée par :

$$p(c_2|\mathbf{x}) = 1 - p(c_1|\mathbf{x}) \quad . \quad (3.6)$$

Une fois l'apprentissage réalisé, le classificateur obtenu permet de traiter toute image présentant des textures similaires. La classification de chacun des nœuds du graphe de similarité associé à une image nous permet d'obtenir une segmentation probabiliste de celle-ci.

Du fait de la présence de bruit dans l'image, et donc de caractéristiques de textures sensiblement différentes, la segmentation retournée par le classificateur peut être assez

3.2. CONTRIBUTION

grossière. Cela est d'autant plus vrai que nous employons des caractéristiques de textures car la non-localité des attributs génère un flou dans les régions à proximité des frontières des objets. De faibles variations de la texture suffisent alors à induire le classificateur en erreur. Afin de corriger les imperfections ainsi générées, une solution consiste à régulariser cette segmentation. La sortie du classificateur étant une fonction définie sur l'ensemble des nœuds du graphe ($f \in \mathcal{H}(V)$), nous pouvons la considérer comme une nouvelle donnée sur laquelle nous proposons d'appliquer un algorithme de segmentation par régularisation de graphe. La figure 3.6 résume le fonctionnement de cet algorithme.

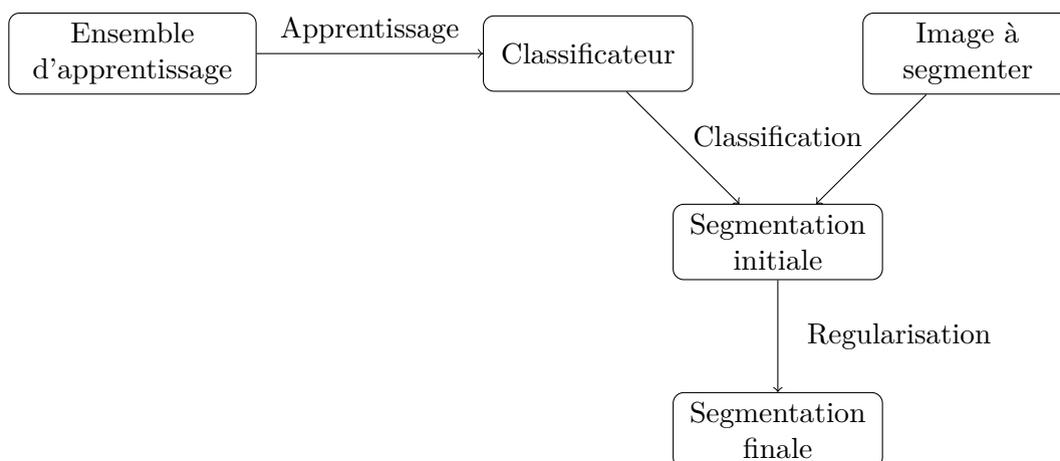


FIGURE 3.6 – Processus de segmentation bivalente par régularisation de classificateurs.

3.2.2 Segmentation multiclasse

Les méthodes de régularisation de graphes que nous avons vu précédemment (voir section 1.7.4) ne permettent de régulariser qu'une unique fonction. Dans un contexte de segmentation, le domaine de valeur de cette fonction est considéré comme binaire. Il existe donc une relation d'ordre qui permet de distinguer les deux régions associées aux valeurs 0 et 1 de la variable. Il est cependant impossible d'établir une telle relation d'ordre pour un nombre de classes supérieur à deux : ces algorithmes ne peuvent donc être utilisés que pour des problèmes de segmentation binaire.

Ta a suggéré de traiter les problèmes de régularisation de fonctions vectorielles à l'aide de plusieurs processus de régularisation distincts, chacun d'entre eux s'intéressant à une composante particulière de la fonction [Ta, 2009]. Afin de traiter un problème de segmentation multiclasse, nous allons appliquer une stratégie similaire.

Le classificateur associé désormais à tout vecteur de caractéristiques de textures $\mathbf{x} \in \mathbb{R}^m$ un vecteur composé des probabilités d'appartenance⁶ à chacune des n classes $c \in \{c_1 \dots c_n\}$

6. Le terme probabilité est ici utilisé par abus de langage, la somme des valeurs obtenues pour chacune des classes n'étant pas nécessairement égale à 1. La majorité des classificateurs fonctionne en effet selon le principe de vote majoritaire : $p(c_k|\mathbf{x})$ doit être vue comme une mesure de la validité de l'hypothèse « la texture \mathbf{x} est caractéristique de la classe c_k ». Ces mesures peuvent être comparées entre elles, mais ne correspondent pas à une mesure de probabilité.

3.2. CONTRIBUTION

connues du système :

$$f(x) = (p(c_1|\mathbf{x}), \dots, p(c_n|\mathbf{x})) \quad . \quad (3.7)$$

Après classification de l'ensemble des nœuds du graphe, les résultats obtenus pour chaque classe sont ensuite considérés indépendamment afin d'être régularisés. Nous faisons cependant l'impasse sur l'étape de seuillage pour permettre à une règle de décision de combiner les résultats des régularisation et ainsi attribuer une classe à chacun des nœuds. Dans nos tests, cette décision sera prise d'après un vote majoritaire. La figure 3.7 résume le fonctionnement de cet algorithme.

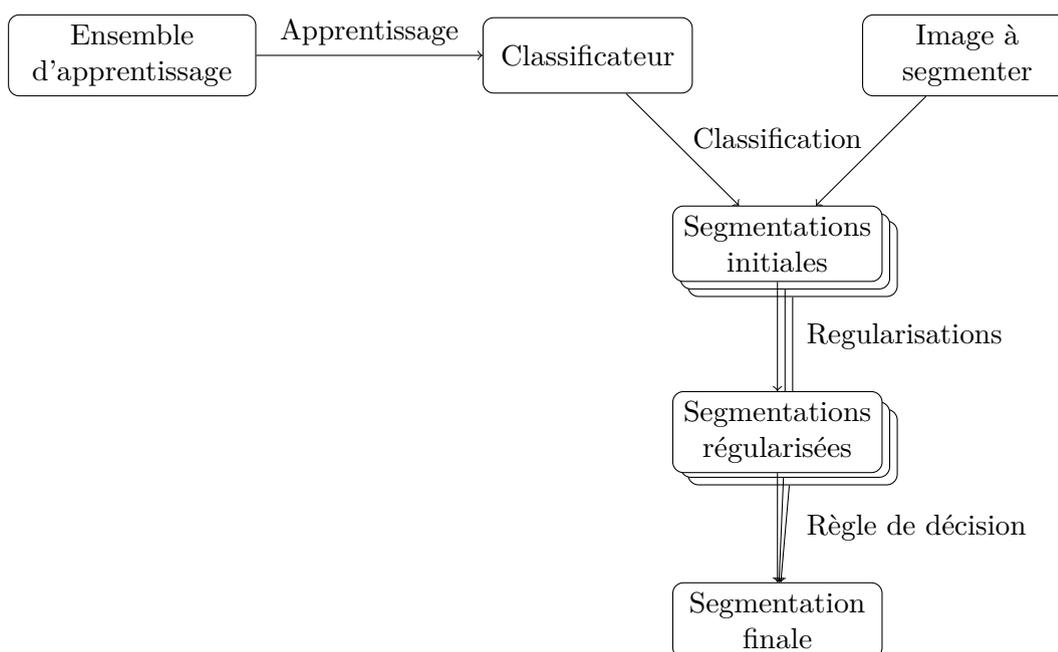


FIGURE 3.7 – Processus de segmentation multiclasse par régularisation de classificateurs.

Il est à noter que cet algorithme est tout à fait applicable à un problème de segmentation biclasse. Cependant, comme nous avons appliqué une stratégie « un contre tous », les deux segmentations retournées par le classificateur sont complémentaires⁷. Les traiter toutes les deux reviendrait alors à traiter deux fois la même information.

3.2.3 Architecture et entraînement des réseaux de neurones

Même si l'objectif de nos travaux n'est pas l'évaluation des performances des classificateurs, nous avons cherché à obtenir, pour chacun de nos tests, les réseaux de neurones permettant une résolution optimale des problèmes de classification.

Afin d'éviter tout phénomène de spécialisation, nous avons opté pour un apprentissage avec validation : deux tiers des éléments de la base d'apprentissage $X = \{(\mathbf{x}_1, c_1) \dots (\mathbf{x}_p, c_p)\}$ sont aléatoirement sélectionnés afin d'être appris et de fixer les poids du réseau tandis que le tiers restant forme l'ensemble de validation et permet d'évaluer la capacité à généraliser du

⁷. En théorie. Dans la pratique, le classificateur présentera une certaine variabilité.

3.2. CONTRIBUTION

réseau. Après chaque cycle d'apprentissage du premier ensemble, la configuration du réseau est enregistrée et sa capacité à généraliser est évaluée sur l'ensemble de validation. Après un nombre prédéfini d'itérations (fixé à mille dans nos tests), la configuration présentant la plus faible taux d'erreurs de classification sur l'ensemble de validation est sélectionnée.

Nous avons également cherché à identifier l'architecture de réseaux de neurones la plus adaptée à nos problèmes de classification. Dans un contexte de classification binaire d'individus décrits par au plus dix attributs numériques, et à partir du processus d'apprentissage décrit précédemment, nous avons évalué les performances de réseaux de neurones composés de une et deux couches cachées, elles-mêmes composées de deux à cinq neurones. L'architecture présentant les meilleurs résultats ainsi que la plus faible complexité (en terme de nombre de poids à fixer) est alors retenue. Pour l'ensemble de nos tests, l'architecture composée d'une unique couche cachée de trois neurones permet l'obtention des meilleurs résultats de classification.

3.2.4 Choix de l'algorithme de régularisation

L'un des objectifs de cette approche étant de se libérer du problème de fixation des poids de l'approche précédente, notre première intention fut de revenir à l'algorithme proposé par Ta et basé sur le modèle de Chan et Vese (voir section 1.7.4.2). Les données que nous manipulons sont désormais scalaires et la pondération des deux classes est rendue inutile du fait de la présence d'un classificateur capable de gérer la question de l'homogénéité des textures. La présence d'un unique paramètre permettant d'établir un compromis entre régularité et précision de la solution reste nécessaire, mais sa détermination est relativement intuitive : il suffit de diminuer sa valeur pour augmenter la régularité de la segmentation. Des tests nous ont permis de valider le bon fonctionnement de cette approche.

Nous pouvons cependant simplifier le processus de régularisation. L'intérêt majeur du terme d'attache aux données de Chan et Vese est sa capacité à identifier la frontière de séparation entre deux classes d'individus selon un critère d'homogénéité. Or, c'est précisément la tâche accomplie par le classificateur. Celui-ci retourne pour chaque individu une valeur réelle comprise dans l'intervalle $[0, 1]$, 0.5 étant le seuil permettant de distinguer une classe d'une autre. Une simple transformation affine $g : [0, 1] \rightarrow [-1, 1]$ permet alors de transformer la borne 0 en une valeur positive et la borne 1 en une valeur négative, tel que requis par la formulation du processus de régularisation (voir équation (1.43)) :

$$g(x) = 1 - 2x \quad , \quad (3.8)$$

avec x la sortie du classificateur.

Nous retrouvons ainsi la formulation du terme d'attache aux données utilisé par le modèle de Rudin, Osher et Fatemi (voir section 1.7.4.1). Les tests réalisés nous ont permis d'obtenir des résultats similaires à ceux du modèle de Chan et Vese, mais avec une amélioration sensible du temps de calcul. Il n'est en effet plus nécessaire de recalculer, avant chaque itération, les caractéristiques moyennes de chacune des régions. Les résultats présentés dans la section 3.3 sont obtenus avec ce modèle.

L'algorithme 2 résume la méthode de régularisation basée sur le modèle de Rudin, Osher et Fatemi.

3.2. CONTRIBUTION

Algorithme 2 : Algorithme d'approximation du modèle de Rudin, Osher et Fatemi.

Entrées : un graphe de similarité $G = (V, E, w)$, une fonction $f_0 : V \rightarrow [0, 1]$ à régulariser, N un nombre d'itérations à réaliser, $\lambda \in R^+$ le paramètre établissant le compromis entre la régularité et la précision de la solution.

Sorties : une fonction $f : V \rightarrow [0, 1]$, version régularisée de f_0 .

pour tous les $u \in V$ **faire**

└ $f^0(u) = f_0(u)$

pour tous les $(u, v) \in E$ **faire**

└ $\beta_{uv} = \sqrt{w_{uv}} |f^n(u) - f^n(v)|^{-1}$

pour $n = 0$ **a** $N - 1$ **faire**

└ **pour tous les** $u \in V$ **faire**

$$\gamma = \frac{\sum_{v \sim u} \beta_{uv} \cdot f^n(v) - \lambda(1 - 2f^0(u))}{\sum_{v \sim u} \beta_{uv}}$$

$$f^{n+1}(u) = \max(\min(\gamma, 1), 0)$$

3.2.5 Application à la segmentation automatique d'images

Un outil de segmentation d'images est souvent utilisé afin de traiter un grand nombre d'images de même nature. Nous pouvons prendre l'exemple d'un praticien qui aurait à analyser des lésions dont les images proviendraient d'un même imageur médical, ou encore celui d'une chaîne de production dans laquelle la conformité de chaque pièce doit être validée.

Ces images étant de même nature, il est envisageable de pouvoir compter sur un classificateur « expert » capable de traiter un problème particulier, et ainsi automatiser l'utilisation de notre méthode. Un tel classificateur est alors conçu par un expert à partir d'un ensemble de vérités terrain qu'il aura manuellement définies. Une fois le classificateur entraîné, celui-ci peut être sauvegardé et réutilisé. Aucune intervention humaine n'est alors nécessaire.

Le classificateur peut par ailleurs être conçu interactivement, sans nécessiter l'obtention de vérités terrain. Afin de définir l'ensemble d'apprentissage, l'utilisateur peut procéder à une labellisation partielle d'une ou plusieurs images. En entraînant le classificateur sur cet ensemble d'apprentissage, puis en exécutant le processus de segmentation sur ces mêmes images, il est possible de valider le fonctionnement du classificateur. Au cas où les résultats ne seraient pas satisfaisants, il est possible de consolider l'ensemble d'apprentissage en le corrigeant ou bien en y ajoutant de nouveaux pixels. Ce processus peut alors être répété jusqu'à obtention d'un résultat satisfaisant.

3.3 Résultats

3.3.1 Segmentation d'images texturées artificielles

Nous avons dans un premier temps souhaité valider le fonctionnement de notre méthode sur une problématique de segmentation binaire. Nous avons pour cela généré un ensemble de seize images texturées, chacune étant composée d'un objet de forme aléatoire. Un exemple d'image est illustré figure 3.8.

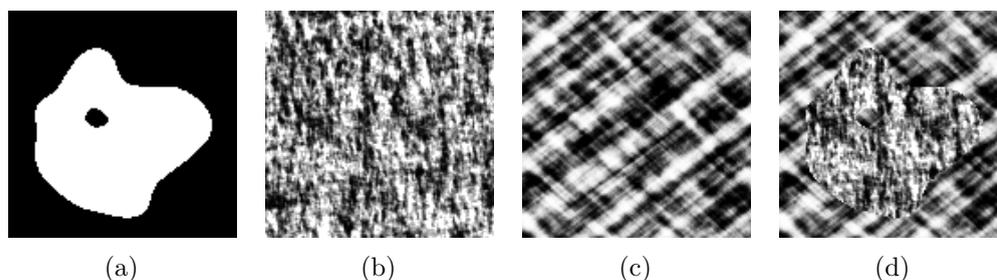


FIGURE 3.8 – Exemple d'image présentant un objet texturé de forme aléatoire. (a) : masque de composition. (b) et (c) : textures. (d) : image résultante.

Les textures employées proviennent de la base d'images Brodatz [Brodatz, 1966]. Nous n'avons cependant considéré que celles présentant des microtextures, les caractéristiques d'Haralick étant relativement peu adaptées aux macrotextures. Nous avons par ailleurs égalisé l'histogramme de chacune des images afin de rendre leurs distributions de niveaux de gris homogènes, rendant ainsi impossible de les discriminer à l'aide de statistiques de premier ordre.

La texture en chacun des pixels des images a été décrite à l'aide de coefficients d'Haralick, calculés dans une région de 11 pixels de diamètre et pour le décalage $(1, 0)$, les images ayant avant cela été réduites à 16 niveaux de gris.

Pour ce test, nous nous sommes placés dans le contexte où un classificateur serait fourni par un expert puis utilisé pour segmenter de multiples images de textures similaires. Nous avons donc entraîné seize classificateurs à reconnaître les textures utilisées dans chacune des images, puis sauvegardé leurs configurations.

Afin d'introduire une certaine variabilité dans les textures, les images ont ensuite été bruitées à l'aide d'un bruit blanc Gaussien de niveau croissant⁸. Cette altération a pour but de forcer le classificateur à généraliser ce qu'il a appris à des caractéristiques de textures encore inconnues. Selon le niveau de bruit, cette généralisation générera des erreurs de classification. Nous serons ainsi en mesure d'évaluer la capacité du processus de régularisation à les corriger.

La texture en chacun des pixels de ces images bruitées a été caractérisée selon les mêmes modalités. Chaque image a ensuite été segmentée pour différentes valeurs du paramètre λ , avec exécution du processus de régularisation jusqu'à l'obtention de solutions stables. Pour chaque image et chaque niveau de bruit, la segmentation présentant les meilleurs résultats

8. Le niveau d'un bruit Gaussien correspond à son écart-type.

3.3. RÉSULTATS

(en termes de distance générique d'anomalie) a été conservée.

Afin d'évaluer le bénéfice lié à la combinaison d'un processus de régularisation et d'un classificateur, nous avons comparé les résultats de notre méthode à ceux obtenus sans régularisation. Dans ce cas, la sortie du classificateur a été binarisée en considérant un seuil de 0.5.

Nous avons aussi comparé les performances de cette méthode à celle du chapitre précédent. Nous avons manuellement défini une initialisation pour chacune des seize images, sous la forme d'une ellipse située dans l'objet central. Afin de se placer dans les mêmes conditions d'utilisation, nous avons simplifié la première méthode en imposant $\lambda_1 = \lambda_2$. Nos deux approches ne comportent ainsi qu'un unique paramètre, dont la valeur optimale (en termes de distance générique d'anomalie) a été déterminée pour chaque niveau de bruit par recherche exhaustive sur un échantillon de valeurs.

Le tableau 3.1 présente les résultats obtenus pour ces trois méthodes sur l'ensemble des images pour le niveau de bruit 8.

Image	Classificateur seul	Régularisation de graphes	Régularisation de classificateurs
1	17.8	11.9	6.3
2	15.1	7.1	11.9
3	15.0	9.5	7
4	8.0	5.5	4
5	4.1	5.3	2.2
6	3.9	6.8	3.7
7	7.3	8.2	4.8
8	11.2	5.4	4.5
9	6.3	4.8	3.9
10	1.3	4.3	0.9
11	17.3	14.5	17.2
12	10.2	5.2	4.2
13	1.8	3.7	1.6
14	2.7	4.2	2.3
15	11.9	6.8	3.2
16	1.2	3.5	1.5
Moyenne	8.5	6.7	4.9
Écart type	5.6	3	4

TABLE 3.1 – Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur la série d'images texturées artificielles. Les valeurs indiquées correspondent à la distance générique d'anomalie.

Étant donné que les classificateurs ont été entraînés à reconnaître les textures non bruitées, il leur est impossible de produire de bonnes classifications à partir d'images fortement altérées. Lorsque le niveau du bruit devient trop important, celui-ci n'est plus

3.3. RÉSULTATS

une source de variabilité mais représente une texture à part entière. Les résultats que nous évoquons par la suite ne concernent donc que les niveaux de bruit 2, 4, 6, 8, 10 et 12.

Les résultats présentés dans les colonnes 2 et 4 du tableau 3.1 illustrent le bénéfice lié à l'application d'un processus de régularisation sur la sortie d'un classificateur. Sur l'ensemble des niveaux de bruit considérés, cette méthode permet de réduire jusqu'à 13.8 % le nombre de pixels mal classés, avec une amélioration moyenne de 3.7 %.

Cette nouvelle méthode se révèle aussi plus précise que celle du chapitre précédent. L'amélioration est certes moins importante, puisque le nombre moyen de pixels mal classés est réduit de seulement 1.9 % (sur l'ensemble des niveaux de bruit). Il n'est cependant plus nécessaire de sélectionner manuellement les caractéristiques de textures les plus pertinentes. Nous pouvons par ailleurs noter une réduction sensible du nombre d'itérations nécessaires à l'obtention d'une solution stable : environ 1000 avec notre nouvelle approche contre plus de 2000 avec la précédente.

Les figures 3.9, 3.10, 3.11 illustrent les résultats obtenus par les différentes méthodes sur trois des images de test. Nous pouvons constater que l'application de la méthode de régularisation sur la sortie du classificateur permet de corriger un certain nombre d'erreurs de classification localisées. Comme nous pouvons le voir dans les figures 3.9 et 3.10, la régularisation peut cependant empêcher l'identification de régions de petite taille, notamment les trous présents dans certains objets.

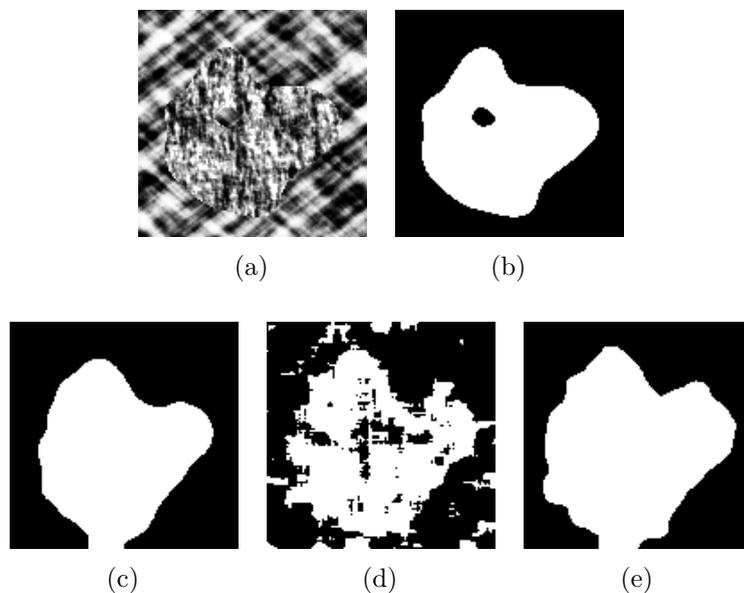


FIGURE 3.9 – Segmentations obtenues par les différentes méthodes sur l'image texturée artificielle 1 pour le niveau de bruit 8. (a) : image bruitée. (b) : vérité terrain. (c) : régularisation de classificateurs. (d) : classificateur seul. (e) : régularisation de graphes.

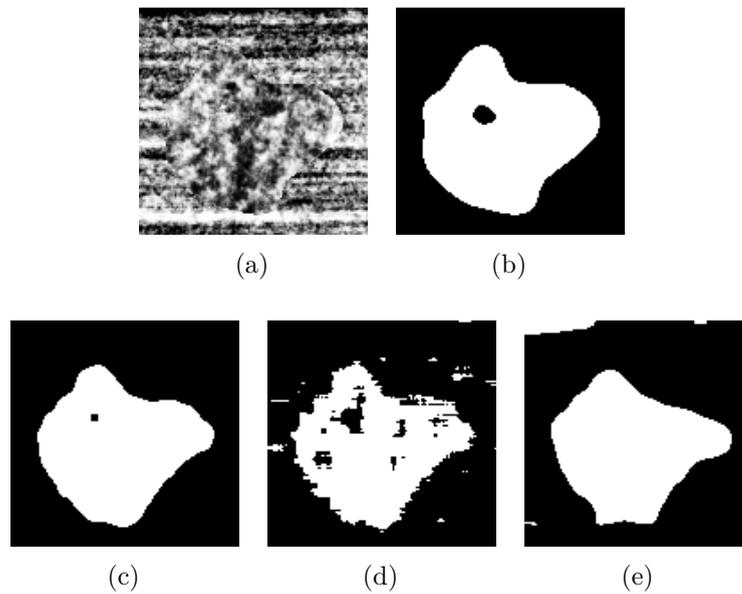


FIGURE 3.10 – Segmentations obtenues par les différentes méthodes sur l'image texturée artificielle 7 pour le niveau de bruit 8. (a) : image bruitée. (b) : vérité terrain. (c) : régularisation de classificateurs. (d) : classificateur seul. (e) : régularisation de graphes.

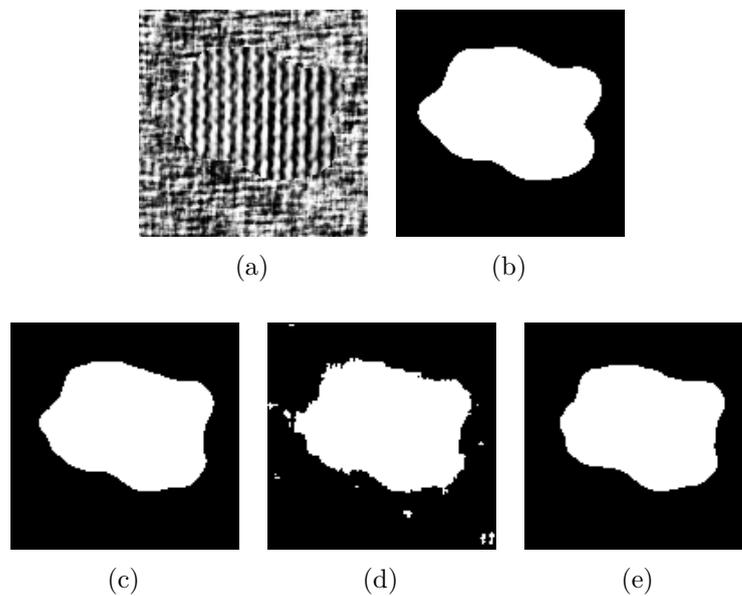


FIGURE 3.11 – Segmentations obtenues par les différentes méthodes sur l'image texturée artificielle 14 pour le niveau de bruit 8. (a) : image bruitée. (b) : vérité terrain. (c) : régularisation de classificateurs. (d) : classificateur seul. (e) : régularisation de graphes.

3.3.2 Segmentation d'images médicales 2D

Comme évoqué dans la section 3.2.5, l'une des applications de cette méthode est la segmentation d'images médicales à l'aide d'un classificateur expert. L'objectif de ce test est de valider ce mode d'utilisation.

Dans la série d'images échographiques utilisée lors des tests de la méthode précédente (voir section 2.3.3), trois d'entre elles présentent des textures similaires. Nous nous sommes donc servis de la vérité terrain de l'une d'entre elles afin d'entraîner un classificateur. Une fois celui-ci entraîné à distinguer la lésion des tissus sains selon leurs textures respectives, il a été utilisé pour segmenter les autres images.

Afin d'évaluer la précision de cette méthode, nous avons mesuré les performances de deux autres approches sur les mêmes images de test : la méthode du chapitre précédent et une méthode de segmentation par contours actifs supervisés [Olivier *et al.*, 2008].

L'approche par contours actifs utilise un classificateur binaire afin de guider l'évolution du contour. Ses performances ont été notamment validées dans un contexte de segmentation d'images médicales et pour une configuration similaire à la notre : un perceptron multicouche en tant que classificateur et des textures décrites à l'aide de caractéristiques d'Haralick. Nous avons ainsi pu appliquer une logique de test similaire : le classificateur a été entraîné sur la première image et utilisé sur les deux autres afin de guider l'évolution du contour.

En ce qui concerne la méthode de segmentation par régularisation de graphes, nous avons, comme pour le test précédent, réduit à un le nombre de paramètres.

Pour nos deux méthodes, différentes valeurs du paramètre d'attache aux données ont été testées, et le meilleur résultat conservé. Le tableau 3.2 présente les résultats obtenus par ces trois méthodes sur la série d'images échographiques. La figure 3.12 représente l'image utilisée pour entraîner le classificateur tandis que les figures 3.13 et 3.14 illustrent les résultats obtenus sur les deux autres images par les différentes méthodes.

Image	Régularisation de graphes	Contour actif	Régularisation de classificateurs
2	2.20	2.52	1.84
3	3.19	4.12	2.0

TABLE 3.2 – Comparaison des différentes méthodes de segmentation sur la série d'images échographiques. Les valeurs indiquées représentent la distance générique d'anomalie normalisée.

Les résultats obtenus confirment que ce mode d'utilisation est parfaitement applicable : un classificateur expert permet de segmenter de multiples images de nature similaire. En fixant la valeur du paramètre de régularisation, il est donc envisageable de rendre la phase de segmentation totalement automatique.

Nous pouvons par ailleurs constater que cette nouvelle méthode est là aussi plus précise que la précédente, ce qui permet de confirmer la supériorité du classificateur face au critère d'attache aux données de Chan et Vese. Le bénéfice est donc double, puisque nous obtenons à la fois une simplification du paramétrage de la méthode ainsi qu'une pondération automatique des différentes caractéristiques de textures.

3.3. RÉSULTATS

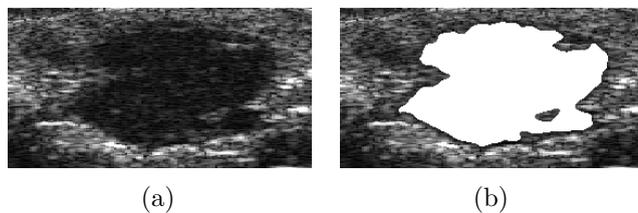


FIGURE 3.12 – Image échographique (a) et vérité terrain (b) utilisées pour entraîner le classificateur.

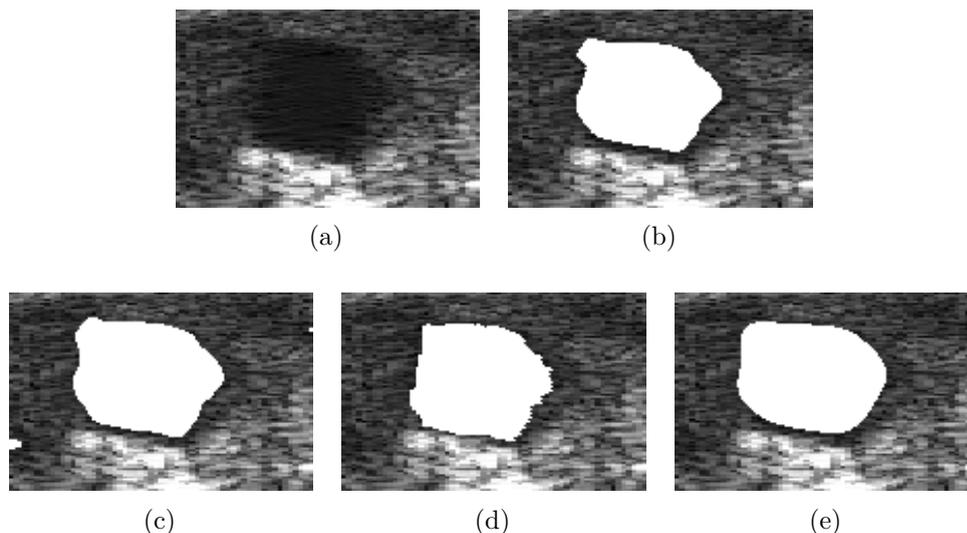


FIGURE 3.13 – Résultats obtenus sur l'image échographique 2D n° 2 par les différentes méthodes. (a) : image originale. (b) : vérité terrain. (c) : régularisation de graphes. (d) : contours actifs. (e) : régularisation de classificateurs.

Afin d'illustrer la souplesse de notre approche, nous l'avons appliquée à de la segmentation d'images acquises par microscopie confocale. Les images obtenues par cette technique sont très différentes de celles obtenues en échographie, puisqu'elles sont relativement nettes. Par ailleurs, la microscopie confocale produit naturellement des images 3D, sous forme de séries de coupes acquises à différentes profondeurs.

L'image considérée pour ce test correspond à une vue du système olfactif de l'abeille, les éléments circulaires présents dans l'image 3.15a étant les glomérules⁹.

Pour ce test, nous avons voulu segmenter plusieurs coupes successives d'une même image. À partir de la vérité terrain de la première coupe, nous avons donc entraîné un classificateur à distinguer les glomérules des autres tissus, puis segmenté les coupes suivantes. Le tableau 3.3 résume les résultats obtenus sur une série de coupes. La segmentation obtenue pour la coupe numéro 4 est illustrée figure 3.15c.

Comme l'illustre l'image 3.15c, notre approche peine à distinguer la frontière séparant

9. Structures chargées de transmettre aux neurones les signaux émis par les cellules chémoréceptrices (capable de détecter des substances chimiques).

3.3. RÉSULTATS

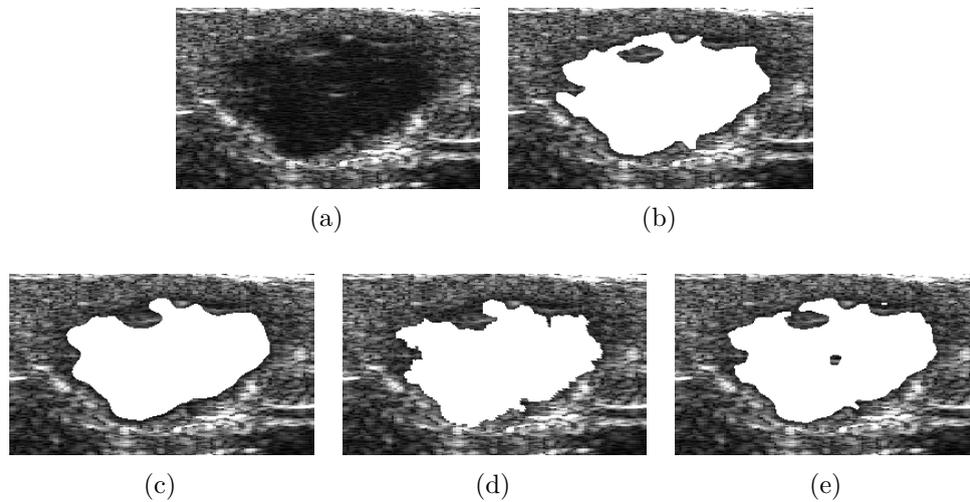


FIGURE 3.14 – Résultats obtenus sur l'image échographique 2D n° 3 par les différentes méthodes. (a) : image originale. (b) : vérité terrain. (c) : régularisation de graphes. (d) : contours actifs. (e) : régularisation de classificateurs.

Image	Distance générique d'anomalie normalisée	Distance de Hausdorff moyenne
2	7.1	0.86
3	8.1	0.90
4	9.1	0.92
5	8.9	0.66

TABLE 3.3 – Évaluation de la précision de la méthode de segmentation par régularisation de classificateurs sur des images de microscopie confocale.

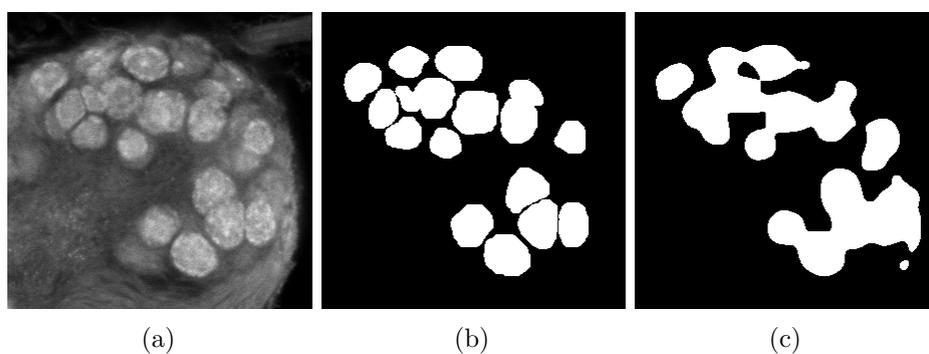


FIGURE 3.15 – Résultat obtenu par la méthode de segmentation par régularisation de classificateurs sur l'image de microscopie confocale. (a) : image originale. (b) : vérité terrain. (c) : segmentation.

des glomérules très proches. Ce comportement est essentiellement dû à l'utilisation de caractéristiques de textures qui sont par définition non-locales et font disparaître ce genre

3.3. RÉSULTATS

de détails. La longueur totale des frontières étant importante dans ces images, le nombre de pixels mal classés augmente rapidement (voir la seconde colonne du tableau 3.3). Afin d'en tenir compte, nous pouvons évaluer la précision de notre méthode à l'aide de la distance de Hausdorff moyenne, qui mesure la distance moyenne entre un pixel mal classé et le contour le plus proche. Comme l'indique la seconde colonne du tableau 3.3), les pixels mal classés se trouvent à moins d'un pixel de distance des contours de la vérité terrain. Selon ce critère, notre méthode reste donc précise.

3.3.3 Segmentation d'images IRM 3D

Dans ce dernier test, nous avons souhaité évaluer les performances de notre méthode sur un problème de segmentation qui soit à la fois multiclasse et 3D.

Concevoir une image artificielle texturée respectant ces critères est relativement complexe. Alors que plusieurs techniques d'imagerie médicale sont capables d'acquérir de telles images, peu de jeux de données sont disponibles. L'un des rares à être fourni accompagné des vérités terrain est mis à disposition par le Centre d'Analyse Morphométrique de l'Hôpital Général du Massachusset¹⁰. Ce jeu de données est composé de vingt images de cerveaux sains, acquises en imagerie par résonance magnétique (IRM). Chaque image est composée d'environ 55 coupes mesurant 256 pixels de côté. Quatre régions sont labellisées dans chaque image : substance blanche, substance grise, liquide céphalo-rachidien et autres (crâne, arrière-plan, globes oculaires...). La figure 3.16 représente une coupe extraite d'une de ces images ainsi que la vérité terrain associée.

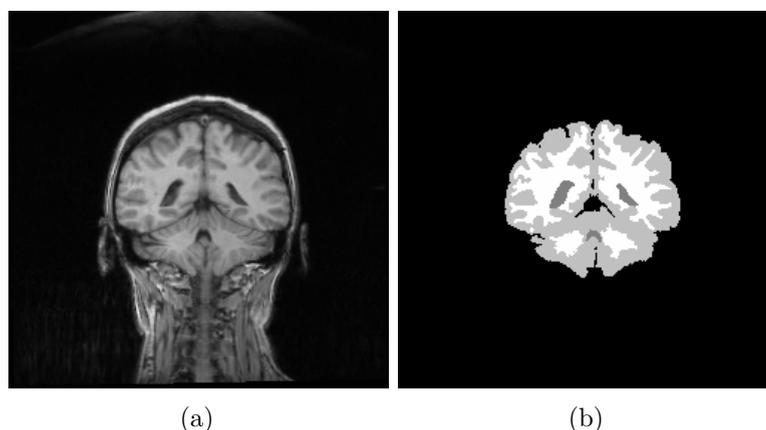


FIGURE 3.16 – Exemple d'image de cerveau acquise en imagerie par résonance magnétique. (a) : image acquise. (b) : vérité terrain. Dans la vérité terrain, les pixels noirs correspondent à la classe « autre », les gris foncé au liquide céphalo-rachidien, les gris clair à la substance grise, et les blancs à la substance blanche.

Afin de traiter une telle image 3D, nous nous sommes placés dans un contexte de segmentation interactive : le praticien doit labelliser une partie des données afin d'entraîner le classificateur, lequel pourra ensuite être utilisé pour segmenter l'image 3D entière. Dans notre cas, nous avons entraîné le classificateur à partir de la vérité terrain d'une des coupes.

10. <http://www.cma.mgh.harvard.edu/ibsr/>

3.3. RÉSULTATS

Le graphe construit pour ces images correspond à un 6-voisinage, dans lequel la valeur 1 fait office de mesure de similarité. Concernant les caractéristiques d'Haralick, nous avons fixé à 16 le nombre de niveaux de gris, et considéré le décalage (1, 0). Les régions que nous souhaitons identifier étant parfois de très petite taille, nous avons dû réduire la taille de la fenêtre de calcul. Les tests réalisés ont montré qu'un diamètre de 9 pixels était tout de même nécessaire afin de correctement caractériser les textures.

Nous avons par ailleurs fixé le nombre d'itérations de régularisation à 50, l'objectif étant ici de corriger les erreurs de classification évidentes et non d'agir sur la régularité des contours. Il est en effet nécessaire d'éviter une régularisation trop importante, au risque de faire disparaître les éléments de petite taille que nous souhaitons identifier.

La figure 3.17 présente le résultat obtenu sur la coupe dont la vérité terrain est illustrée figure 3.16. Les figures 3.18, 3.19 et 3.20 présentent, sous forme de maillage 3D, le résultat obtenu pour l'image complète.



FIGURE 3.17 – Segmentation obtenue sur une des coupes de l'image IRM 3D.

Nous constatons que la segmentation respecte la forme globale des régions. Les éléments de petite taille sont en revanche absents du résultat. Cet effet est essentiellement dû aux caractéristiques de textures qui, du fait de leur non localité, ne permettent pas ce niveau de détail. La région correspondant au liquide céphalo-rachidien, qui est de petite taille, n'est d'ailleurs pas détectée.

Afin d'évaluer la précision de notre méthode, nous avons comparé la segmentation obtenue à la vérité terrain. Nous avons de nouveau choisi de mesurer la distance de Hausdorff moyenne car celle-ci permet de mieux apprécier la précision globale de la segmentation malgré le manque de détails. Le tableau 3.4 présente les résultats obtenus avec et sans régularisation pour chacune des classes identifiées.

Pour les deux régions du cerveau qui ont été identifiées, la distance générique d'anomalie montre un taux de pixels mal classés non négligeable. Ce mauvais score est malheureusement amplifié du fait de la petite taille de ces régions. La distance de Hausdorff moyenne nous indique cependant que ces pixels mal classés restent très proches des contours de la vérité terrain, confirmant ainsi la cohérence de la segmentation calculée. La distance de Hausdorff moyenne est par ailleurs plus importante en l'absence de régularisation, ce qui justifie de nouveau la validité de notre approche.

3.3. RÉSULTATS

Région	Avec régularisation (DGA)	Avec régularisation (\bar{H})	Sans régularisation (\bar{H})
Autre	2	0.03	0.03
Substance grise	20	1.54	1.90
Substance blanche	50	2.34	2.78

TABLE 3.4 – Évaluation de la précision de la méthode sur une image d’IRM 3D. DGA : distance générique d’anomalie. \bar{H} : distance de Hausdorff moyenne.

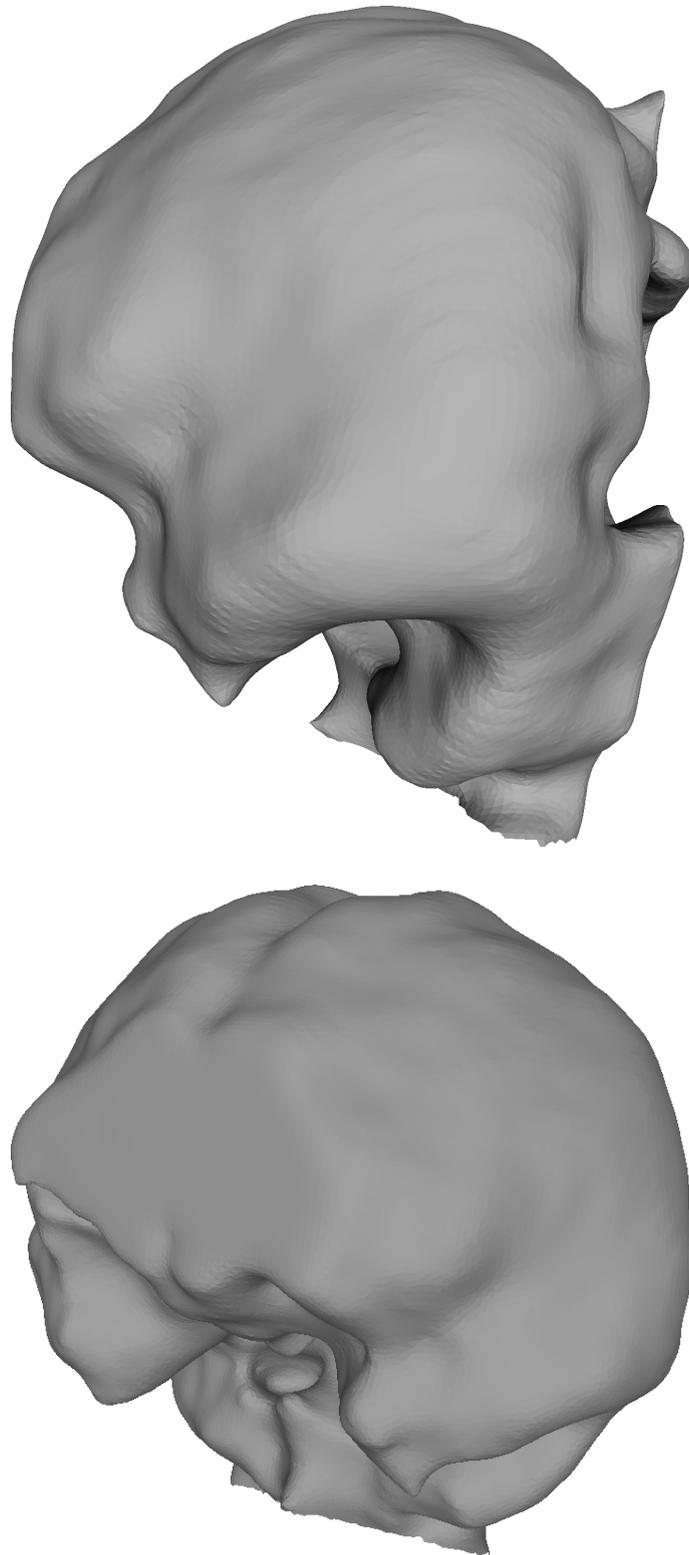


FIGURE 3.18 – Représentation sous forme de maillage de la région correspondant à la substance grise.

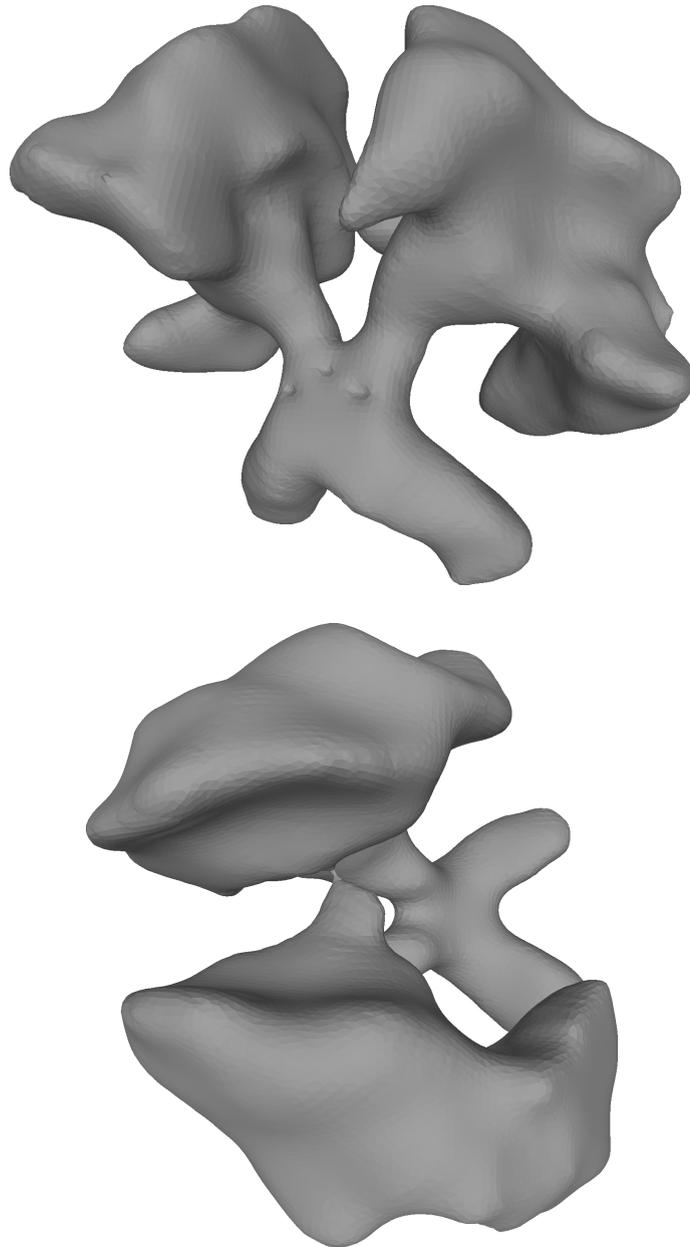


FIGURE 3.19 – Représentation sous forme de maillage de la région correspondant à la substance blanche.

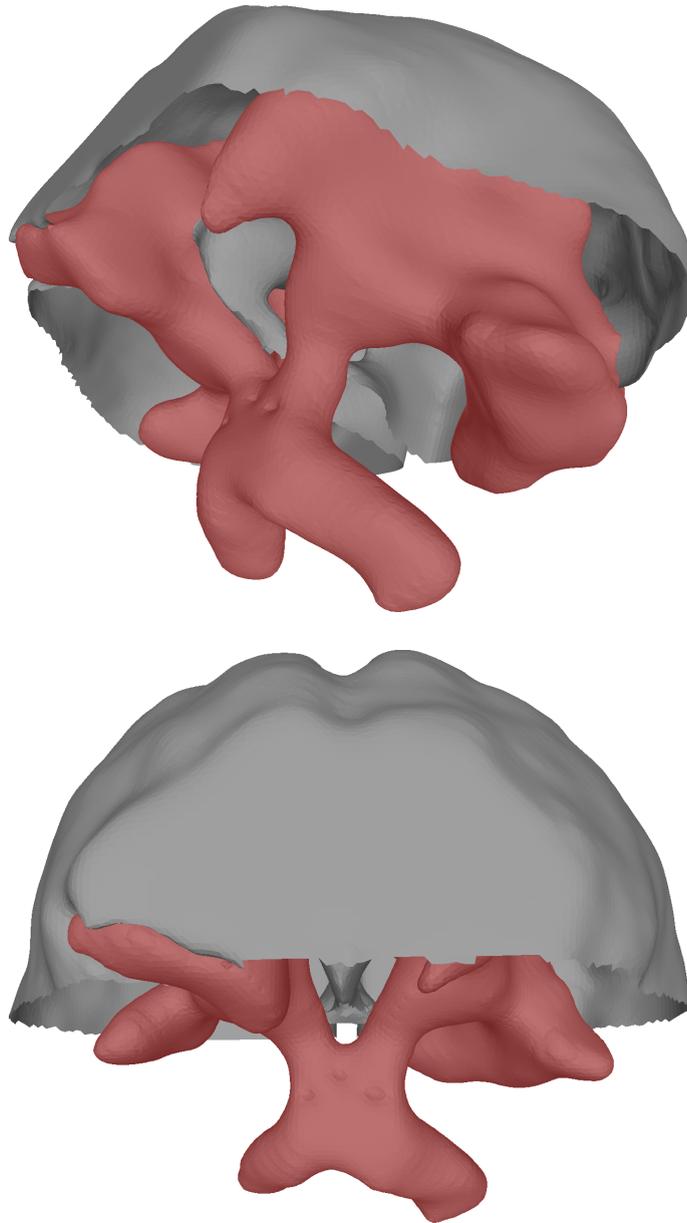


FIGURE 3.20 – Représentation sous forme de maillage de la région correspondant à la substance blanche (en rouge) et à la substance grise. Le maillage de la substance grise a été découpé afin de laisser apparaître la substance blanche qui se trouve à l'intérieur.

3.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode de segmentation d'images qui combine un classificateur supervisé à un algorithme de régularisation de graphes. L'objectif de cette approche était d'offrir une solution au problème de détermination des paramètres inhérents à la méthode de segmentation d'images texturées par régularisation de graphes présentée dans le chapitre 2.

Pour rappel, cette dernière se base sur un critère d'attache aux données inspiré des travaux de Chan et Vese [Chan *et al.*, 2000] afin d'identifier la texture des différentes régions de l'image. Comme l'ont cependant montré nos tests (voir section 2.3), ce critère nécessite la détermination d'un ensemble de paramètres afin de pondérer les différentes caractéristiques de textures d'après leur pertinence. Malheureusement, il n'existe actuellement aucune méthode supervisée permettant d'effectuer ce paramétrage.

Grâce à l'utilisation d'un classificateur supervisé, nous sommes en mesure de modéliser plus précisément le problème puisque la phase d'apprentissage tient compte de la pertinence des différentes caractéristiques. Un classificateur est donc utilisé en lieu et place du critère d'attache aux données. Les tests réalisés ont confirmé une amélioration de la précision du processus de segmentation tout en offrant un paramétrage simplifié de la méthode. Étant donné qu'aucun descripteur n'est adapté à tous types de textures, nous pouvons par ailleurs envisager d'en multiplier le nombre et ainsi d'aboutir à un outil de segmentation générique.

Conclusion générale et perspectives

Les travaux que nous présentons dans ce manuscrit de thèse prolongent une récente approche de segmentation d'images basée sur un processus de régularisation de graphes [Ta, 2009]. Cette approche considère le problème de segmentation d'images comme un problème d'optimisation bicritère. La qualité de la solution y est évaluée à l'aide d'une énergie dans laquelle la régularité de la segmentation est opposée à sa précision. Ces deux propriétés sont mesurées respectivement par une fonctionnelle de régularisation et un terme d'attache aux données. Sous réserve du respect de certaines contraintes, ce problème est un problème d'optimisation globale. Il admet donc une unique solution, et peut être résolu à l'aide d'un algorithme itératif. La segmentation ainsi calculée établit un compromis entre les critères de régularité et de précision.

L'originalité de cette approche est de représenter les images à l'aide de graphes de similarité. Comparés aux structures de grilles traditionnellement utilisées en traitement d'images, les graphes offrent une plus grande flexibilité et permettent de stocker des informations de plus haut niveau. En établissant des relations entre les pixels similaires, il est notamment possible de mettre en avant certaines structures ou certains motifs dans les images. Cette approche est ainsi qualifiée de « non-locale » puisqu'elle permet de traiter une image avec un plus haut niveau d'abstraction que les pixels qui la composent

Dans [Ta, 2009], deux modèles de segmentation par régularisation de graphes sont proposés. Ils permettent malheureusement de ne considérer qu'un unique attribut pour chaque pixel : son niveau de gris. Or, lorsque utilisée de façon « brute », cette donnée n'est pas toujours discriminante. La présence de bruit ou de textures dans l'image peut induire en erreur le critère d'attache aux données.

Afin de permettre une meilleure identification de la classe d'appartenance de chacun des pixels d'une image, il est possible d'obtenir des informations de plus haut niveau : des caractéristiques de textures. Les méthodes de caractérisation de textures sont elles aussi des méthodes non-locales, puisqu'elles requièrent d'analyser la distribution et l'agencement spatial des niveaux de gris dans une région autour de chaque pixel. Les mesures obtenues permettent de décrire des propriétés telles que l'orientation, la régularité ou encore la granularité de la texture pour chaque pixel d'une image.

L'objectif des travaux réalisés durant cette thèse a été de combiner ces deux approches non-locales : la segmentation par régularisation de graphes et la caractérisation de textures.

Dans le chapitre 2, nous avons proposé une première méthode dans laquelle la prise en compte des caractéristiques de textures est directement intégrée au sein du processus de régularisation de graphes. Dans ce dernier, le terme d'attache aux données était à l'origine basé sur les travaux de Chan et Vese [Chan et Vese, 2001] et permettait l'identification de deux régions à partir d'un critère mesurant l'homogénéité des niveaux de gris dans chacune d'entre elles. En s'inspirant d'une extension des travaux de Chan et Vese [Chan *et al.*, 2000], nous avons formulé un nouveau critère d'attache aux données capable de mesurer l'homogénéité de vecteurs d'attributs.

À l'aide de caractéristiques d'Haralick [Haralick *et al.*, 1973], cette méthode a été appliquée avec succès sur une problématique de segmentation d'images échographiques 2D. Elle s'est avérée plus précise que la méthode de segmentation par régularisation de graphes de Ta, mais aussi plus précise que la méthode de contours actifs pour image multimodales de Chan et Vese [Chan *et al.*, 2000], confirmant ainsi le bénéfice lié à la combinaison des deux approches non-locales.

Nous avons par ailleurs été en mesure d'évaluer notre méthode sur des images échographiques 3D, illustrant ainsi un avantage majeur des méthodes à base de graphes : l'indépendance vis à vis de la structure des données.

Nous avons cependant constaté que le paramétrage de notre méthode constituait une faiblesse. Afin d'optimiser l'identification des différentes textures d'une image, il est en effet nécessaire de pondérer chaque caractéristique en fonction de sa pertinence. Le réglage de ces poids aux valeurs optimales nécessite donc un *a priori* concernant les caractéristiques et reste une opération complexe.

Afin de remédier à cela, nous avons proposé dans le chapitre 3 une seconde méthode permettant de combiner les deux approches non-locales. Les différentes régions de l'image sont désormais identifiées à l'aide d'un classificateur supervisé, lequel est naturellement capable d'identifier les caractéristiques les plus pertinentes. Une fois entraîné à reconnaître certains types de textures, le classificateur permet d'attribuer à chaque pixel d'une image une valeur mesurant son potentiel d'appartenance à chacune des classes. Afin de corriger certaines erreurs de classification et de produire une segmentation plus lisse, ces valeurs sont ensuite régularisées à l'aide du processus de segmentation par régularisation de graphes. Le terme d'attache aux données est alors remplacé par la sortie du classificateur, simplifiant ainsi le paramétrage de la méthode. Nous avons par ailleurs étendu cette méthode à une problématique de segmentation multiclassées. Une stratégie « un contre tous » est alors appliquée pour permettre la régularisation de chacune des classes, et la segmentation finale est obtenue selon un principe de vote majoritaire.

Outre le fait de réduire la complexité du paramétrage, cette seconde méthode permet le développement de systèmes de segmentation experts : un classificateur peut être entraîné à résoudre un problème de segmentation connu puis utilisé pour traiter de nombreuses images similaires. Il est ainsi envisageable de développer des outils de segmentation parfaitement autonomes.

Cette méthode a été validée dans sa configuration biclassée sur des images texturées artificielles, sur des images échographiques 2D ainsi que sur des images de microscopie confocale. Comparée à la première méthode, nous avons pu constater une amélioration

de la précision de segmentation ainsi qu'une réduction du temps de calcul. Le bénéfice est cependant double puisque le paramétrage de la méthode est désormais bien moins complexe.

Nous avons enfin appliqué cette méthode sur des images d'IRM 3D et dans un contexte multiclasse. Les caractéristiques d'Haralick se sont alors révélées limitées quant à la caractérisation de régions de très petites tailles, mais les résultats obtenus restent satisfaisants.

Nous envisageons actuellement plusieurs pistes pour permettre à ces méthodes d'être à la fois plus précises et plus rapides.

La première concerne la caractérisation des textures. Dans le cadre de cette thèse, nous nous sommes limités à l'utilisation de caractéristiques d'Haralick, méthode reconnue dans la littérature comme pertinente. Elle n'est cependant pas adaptée à tous types de textures, comme nous avons pu le constater sur les images IRM 3D pour l'identification de régions de petites tailles. Notre méthode de segmentation par régularisation de classificateurs autorisant l'utilisation d'un nombre arbitraire d'attributs, il serait intéressant d'employer plusieurs méthodes de caractérisation de textures afin de combiner les avantages de chacune. Nous pensons en effet possible de concevoir un outil universel de segmentation basé textures.

L'augmentation du nombre de caractéristiques de textures soulève la question des performances du classificateur. Dans cette thèse, nous avons, par mesure de commodité, choisi d'utiliser des réseaux de neurones artificiels. Il est cependant reconnu que les machines à vecteurs supports constituent actuellement une meilleure alternative, et nous travaillons à leur intégration dans notre seconde méthode. La fusion de classificateurs est par ailleurs envisagée.

La dernière piste concerne l'optimisation de la topologie des graphes. De nombreux travaux présentent en effet la non-localité comme une source d'informations complémentaires. Cependant, si l'influence de certaines topologies sur la qualité des résultats a parfois été constatée, aucune théorie à ce sujet n'a été portée à notre connaissance. Nous nous sommes donc restreints à des topologies simples et maîtrisées, mais restons persuadés que des topologies « évoluées » permettraient d'accélérer les traitements et d'augmenter la précision de nos méthodes. Nous travaillons actuellement sur l'utilisation de mesures de pondération afin de rendre le processus de diffusion anisotrope et lui permettre de mieux identifier les régions de petite taille.

Annexes

Annexe A

Caractéristiques d’Haralick

Le but de cette annexe est de compléter les travaux d’Haralick. Dans un premier temps, nous souhaitons clarifier les formules liées au calcul des différentes caractéristiques. En effet, dans [Haralick *et al.*, 1973] – l’article de référence concernant les caractéristiques d’Haralick – plusieurs formules n’ont pas été explicitement définies. De ce fait, il est courant de trouver des travaux scientifiques ainsi que des implémentations dans lesquels les calculs sont basés sur des formules erronées, ce à quoi nous souhaitons remédier.

Par ailleurs, certaines caractéristiques sont connues sous différents noms, ou encore utilisent différentes formulations. Pour celles dont c’est le cas, nous avons recensé les différentes variations.

Enfin, nous avons souhaité fournir une information non communiquée par Haralick : le domaine de valeur des différentes caractéristiques. Cette donnée est en effet nécessaire si l’on souhaite normaliser les valeurs obtenues, étape requise par de nombreux outils afin de ne pas privilégier les caractéristiques présentant de fortes valeurs.

A.1 Notations et calculs préliminaires

Soient G l’ensemble des niveaux de gris pour lesquels une image est échantillonnée et C une matrice de co-occurrence normalisée¹. $\forall i, j \in G$, $C(i, j)$ correspond à la probabilité de trouver une co-occurrence de niveaux de gris (i, j) dans la région analysée.

C_x (resp. C_y) est définie comme étant la matrice de probabilité marginale obtenue en additionnant les lignes (resp. les colonnes) de C :

$$C_x(i) = \sum_{j \in G} C(i, j), \quad \forall i \in G, \quad C_y(j) = \sum_{i \in G} C(i, j), \quad \forall j \in G \quad (\text{A.1})$$

μ_x et σ_x (resp. μ_y et σ_y) sont la moyenne et l’écart-type de C_x (resp. C_y).

$$\mu_x = \sum_{k \in G} k \cdot C_x(k), \quad \mu_y = \sum_{k \in G} k \cdot C_y(k) \quad (\text{A.2})$$

1. Par soucis de clarté, nous ne ferons pas mention dans cette annexe du décalage pour lequel la matrice de co-occurrences est calculé, celui-ci n’entrant pas en compte dans le calcul des caractéristiques de textures.

$$\sigma_x = \sqrt{\sum_{k \in G} \mathbf{C}_x(k) \cdot (k - \mu_x)^2}, \quad \sigma_y = \sqrt{\sum_{k \in G} \mathbf{C}_y(k) \cdot (k - \mu_y)^2} \quad (\text{A.3})$$

Note 1 Lorsque la matrice de co-occurrence est symétrique, les matrices de probabilité marginale sont identiques. Elles représentent alors les probabilités d'apparition de chaque niveau de gris. Leurs moyennes et écarts-types sont eux aussi égaux.

Note 2 De nombreuses implémentations d'outils de calcul de caractéristiques d'Haralick ont recours à des formules erronées en ce qui concerne le calcul des moyennes et écarts-types des matrices de probabilités marginales, car elles ne considèrent pas que ces probabilités sont associées à des événements. Ces outils calculent alors des moyennes de probabilités, ce qui n'a aucune signification d'un point de vue logique ou mathématique.

\mathbf{C}_{x+y} et \mathbf{C}_{x-y} correspondent aux matrices de probabilités liées respectivement au niveau de gris moyen et à la différence de niveau de gris des co-occurrences de la matrice \mathbf{C} .

$$\mathbf{C}_{x+y}(k) = \sum_{\substack{i \in G \\ i+j=k}} \sum_{j \in G} \mathbf{C}(i, j), \quad \forall k \in \{2G_l, \dots, 2(G_h - G_l)\} \quad (\text{A.4})$$

$$\mathbf{C}_{x-y}(k) = \sum_{\substack{i \in G \\ |i-j|=k}} \sum_{j \in G} \mathbf{C}(i, j), \quad \forall k \in \{0, \dots, N_g - 1\} \quad (\text{A.5})$$

A.2 Propriétés utiles

Nous présentons ici deux propriétés qui seront par la suite utilisées pour calculer le domaines de valeur des caractéristiques d'Haralick.

Propriété 1. *Le domaine de valeurs de la variance d'une variable aléatoire définie sur l'intervalle $[a, b]$ est :*

$$\left[0, \frac{(b-a)^2}{4} \right] \quad (\text{A.6})$$

Démonstration. La variance mesure l'écart à la moyenne d'une variable aléatoire. Lorsque l'ensemble des valeurs prises par la variable sont identiques, l'écart à la moyenne est nul. La borne minimale de la variance est donc 0. L'écart à la moyenne est maximal lorsque les valeurs prises par la variable aléatoire sont équiprobablement réparties aux bornes de son domaine de définition.

Si l'on considère une variable aléatoire discrète X définie sur l'intervalle $[a, b]$ par la fonction de masse $p(x \in [a, b])$ avec :

$$p(a) = p(b) = 1/2 \quad (\text{A.7})$$

la moyenne m de la variable aléatoire est alors égale à :

$$\mu = (a + b)/2 \quad (\text{A.8})$$

et sa variance est égale à :

$$\begin{aligned}
 \text{var}(X) &= \sum_{x \in X} p(x) \cdot (x - \mu)^2 \\
 &= \frac{1}{2} \cdot (a - \mu)^2 + \frac{1}{2} \cdot (b - \mu)^2 \\
 &= \frac{1}{2} \left(a - \frac{a+b}{2} \right)^2 + \frac{1}{2} \left(b - \frac{a+b}{2} \right)^2 \\
 &= \frac{(b-a)^2}{4}
 \end{aligned} \tag{A.9}$$

□

Propriété 2. *Le domaine de valeurs de l'entropie d'une source de données qui comporte n symboles est :*

$$[0, \log(n)] \tag{A.10}$$

Démonstration. On considère une variable aléatoire discrète X comportant n symboles et définie par la fonction de masse $p(x \in \{x_1 \dots x_n\})$. L'entropie $H(X)$ de cette variable aléatoire est définie par :

$$H(X) = \sum_{i=1}^n -p(x_i) \cdot \log(p(x_i)) \tag{A.11}$$

L'entropie d'une source de données est minimale lorsque celle-ci retourne un unique symbole avec une probabilité de 1. Elle vaut alors :

$$\begin{aligned}
 H(X) &= 1 \cdot \log(1) \\
 &= 0
 \end{aligned} \tag{A.12}$$

L'entropie d'une source de données atteint son maximum lorsque tous les symboles sont équiprobables. Elle vaut alors :

$$\begin{aligned}
 H(X) &= \sum_{i=1}^n -\frac{1}{n} \cdot \log\left(\frac{1}{n}\right) \\
 &= \log(n)
 \end{aligned} \tag{A.13}$$

□

A.3 Caractéristiques de textures

Dans [Haralick *et al.*, 1973], Haralick considère que l'ensemble de définition des niveaux de gris est $G = \{1, \dots, N_g\}$, alors que de nombreux travaux plus récents font quant à eux référence à l'ensemble $G = \{0, \dots, N_g - 1\}$. Cette différence ne change rien en ce qui concerne le calcul ainsi que la signification de caractéristiques. En revanche, le domaine de valeurs de certaines d'entre elles sera différent, ce dont il faudra tenir compte lors de leur utilisation. Dans la suite de ce document, nous considérerons que l'ensemble de définition des niveaux de gris est $G = \{G_l, \dots, G_h\}$, avec $|G| = N_g$, et définirons l'ensemble de valeur de chacune des caractéristiques à l'aide de ces deux paramètres G_l et G_h .

A.3.1 *Angular second moment*

Nom(s)

- *Angular second moment* ou *ASM* [Haralick *et al.*, 1973, Boland, 1999, McAullife, 2008],
- *Energy* [Pincus, 2004].

Formule(s)

Dans [Haralick *et al.*, 1973, Boland, 1999, Pincus, 2004, McAullife, 2008] :

$$f_{asm} = \sum_{i \in G} \sum_{j \in G} C(i, j)^2 \quad . \quad (\text{A.14})$$

Domaine de valeurs $]0, 1]$.

Interprétation Cet indice est une mesure de l'homogénéité d'une texture. Une texture homogène présentera peu de co-occurrences de niveaux de gris différentes, qui auront donc de fortes probabilités, alors qu'une texture bruitée en présentera un nombre important, de faibles probabilités.

Note [McAullife, 2008] appelle *energy* ou encore *uniformity* la racine carrée de cet indice.

A.3.2 *Contrast*

Nom(s)

- *Contrast* [Haralick *et al.*, 1973, Boland, 1999, McAullife, 2008],
- *Inertia* [Pincus, 2004].

Formule(s)

Dans [Haralick *et al.*, 1973, Boland, 1999] :

$$f_{cont} = \sum_{k=0}^{N_g-1} k^2 \left\{ \sum_{\substack{i \in G \\ j \in G \\ |i-j|=k}} C(i, j) \right\} \quad . \quad (\text{A.15})$$

Dans [Pincus, 2004, McAullife, 2008] :

$$f_{cont} = \sum_{i \in G} \sum_{j \in G} C(i, j) \cdot (i - j)^2 \quad . \quad (\text{A.16})$$

A.3. CARACTÉRISTIQUES DE TEXTURES

Ces deux formules sont équivalentes, et peuvent être réécrites par :

$$f_{cont} = \sum_{k=0}^{N_g-1} k^2 \cdot \mathbf{C}_{x-y}(n) \quad . \quad (\text{A.17})$$

Domaine de valeurs $[0, (N_g - 1)^2]$.

Interprétation Le contraste est une mesure de la dissimilarité des intensités de niveaux de gris relativement à la relation de voisinage. Une texture présentant des pixels dont l'intensité de leurs niveaux de gris est similaire aura un faible contraste, tandis qu'une texture présentant des niveaux de gris d'intensité très différentes aura un fort contraste.

Note [McAullife, 2008] utilise l'indice *dissimilarity* dans lequel la différence entre les niveaux de gris influe linéairement et non quadratiquement.

$$f_{diss} = \sum_{k=0}^{N_g-1} k \cdot \mathbf{C}_{x-y}(n) \quad . \quad (\text{A.18})$$

Le domaine de valeur est alors réduit à $[0, N_g - 1]$.

A.3.3 *Correlation*

Nom(s)

– *Correlation* [Haralick *et al.*, 1973, Pincus, 2004, McAullife, 2008].

Formule(s)

Dans [Haralick *et al.*, 1973] :

$$f_{corr} = \frac{1}{\sigma_x \cdot \sigma_y} \cdot \left(\left(\sum_{i \in G} \sum_{j \in G} i \cdot j \cdot \mathbf{C}(i, j) \right) - \mu_x \cdot \mu_y \right) \quad . \quad (\text{A.19})$$

Dans [Pincus, 2004, McAullife, 2008] :

$$f_{corr} = \frac{1}{\sigma_x \cdot \sigma_y} \cdot \sum_{i \in G} \sum_{j \in G} (i - \mu_x) \cdot (j - \mu_y) \cdot \mathbf{C}(i, j) \quad . \quad (\text{A.20})$$

Ces deux formules sont équivalentes.

Domaine de valeurs $[-1, 1]$.

Interprétation La corrélation est une mesure de la dépendance des intensités des niveaux de gris relativement à la relation de voisinage. Une image dont la texture présente peu de variations aura un indice de corrélation de forte amplitude. À l'inverse, une texture aléatoire aura un faible indice de corrélation.

Note Au cas où au moins un des écarts-types σ_x ou σ_y serait nuls, ce qui est notamment le cas lorsque l'image ne présente aucune texture et est uniforme, aucune valeur de corrélation ne peut être donnée.

A.3.4 *Sum of squares: Variance*

Nom(s)

- *Sum of squares: Variance* [Haralick *et al.*, 1973].

Formule(s)

Dans [Haralick *et al.*, 1973] :

$$f_{var} = \sum_{i \in G} \sum_{j \in G} (i - \mu)^2 \cdot C(i, j) \quad . \quad (\text{A.21})$$

Aucune précision n'est donnée concernant la valeur de μ . Par analogie avec la formule traditionnelle de la variance, on en déduit que $\mu = \mu_i$. Cette expression peut alors être reformulée :

$$f_{var} = \sigma_x^2 \quad (\text{A.22})$$

Domaine de valeurs $[0, (N_g - 1)^2/4]$ (voir propriété 1 page 120).

Interprétation La variance est une mesure de l'étendue des valeurs des niveaux de gris par rapport au niveau de gris moyen de l'image. Une texture unie présentera une variance plus faible qu'une image bruitée.

A.3.5 *Inverse difference moment*

Nom(s)

- *Inverse difference moment* [Haralick *et al.*, 1973],
- *Local homogeneity* [Connors *et al.*, 1984],
- *Difference moment* [Pincus, 2004],
- *Homogeneity* [McAullife, 2008].

A.3. CARACTÉRISTIQUES DE TEXTURES

Formule(s)

Dans [Haralick *et al.*, 1973, Pincus, 2004, McAullife, 2008] :

$$f_{idm} = \sum_{i \in G} \sum_{j \in G} \frac{1}{1 + (i - j)^2} \cdot \mathbf{C}(i, j) \quad . \quad (\text{A.23})$$

Domaine de valeurs $]0, 1]$

Interprétation Il s'agit d'une mesure de l'homogénéité d'une texture. Les paires de pixels dont l'intensité de niveaux de gris est très différente seront faiblement pondérées par cette équation, contrairement aux paires de pixels présentant des niveaux de gris similaires. Cette mesure varie avec l'inverse du contraste.

A.3.6 *Sum average*

Nom(s)

– *Sum average* [Haralick *et al.*, 1973].

Formule(s)

Dans [Haralick *et al.*, 1973] :

$$f_{savg} = \sum_{k=2G_l}^{2(G_h - G_l)} k \cdot \mathbf{C}_{x+y}(k) \quad . \quad (\text{A.24})$$

Domaine de valeurs $[2G_l, 2(G_h - G_l)]$.

Interprétation Cet indice mesure la valeur moyenne des niveaux de gris de chaque paire de pixels considérée par la relation de voisinage.

A.3.7 *Sum variance*

Nom(s)

– *Sum variance* [Haralick *et al.*, 1973].

Formule(s)

Dans [Haralick *et al.*, 1973] :

$$f_{svar} = \sum_{k=2G_l}^{2G_h} (k - f_{sent})^2 \cdot \mathbf{C}_{x+y}(k) \quad . \quad (\text{A.25})$$

Haralick fait référence dans sa formule à la caractéristique *sum entropy*, calculée dans la section A.3.8. Cependant, par analogie avec la formule de la variance, il semble beaucoup plus logique que ce soit la caractéristique *sum average* (voir section A.3.6), à laquelle Haralick ait voulu faire référence.

$$f_{svar} = \sum_{k=2G_l}^{2G_h} (k - f_{savg})^2 \cdot \mathbf{C}_{x+y}(k) \quad (\text{A.26})$$

Domaine de valeurs $[0, (N_g - 1)^2]$ (voir propriété 1).

Interprétation Cet indice mesure l'étendue de l'intensité moyenne des co-occurrences.

A.3.8 *Sum entropy*

Nom(s)

- *Sum entropy* [Haralick et al., 1973].

Formule(s)

Dans [Haralick et al., 1973] :

$$f_{sent} = - \sum_{k=2G_l}^{2G_h} \mathbf{C}_{x+y}(k) \cdot \log(\mathbf{C}_{x+y}(k)) \quad . \quad (\text{A.27})$$

Domaine de définition $[0, \log(2(N_g - 1))]$

Interprétation Cet indice mesure le degré de prédictibilité de l'intensité moyenne des co-occurrences.

Note Étant donné que $\log(0)$ n'est pas défini, Haralick préconise d'utiliser la version régularisée de la fonction logarithme, $\log(x + \epsilon)$, ϵ étant une constante positive proche de zéro. Cependant, étant donné que la fonction $f : x \mapsto x \cdot \log(x)$ admet 0 comme limite lorsque x tend vers 0, cette régularisation peut être ignorée.

A.3.9 *Entropy*

Nom(s)

A.3. CARACTÉRISTIQUES DE TEXTURES

– *Entropy* [Haralick *et al.*, 1973, Pincus, 2004, McAullife, 2008].

Formule(s)

Dans [Haralick *et al.*, 1973, Pincus, 2004, McAullife, 2008] :

$$f_{ent} = - \sum_{i \in G} \sum_{j \in G} \mathbf{C}(i, j) \cdot \log(\mathbf{C}(i, j)) \quad . \quad (\text{A.28})$$

Domaine de valeurs $[0, 2 \log(N_g)]$

Interprétation Cet indice quantifie la prédictibilité de la texture. Il sera faible si une texture est régulière – c’est à dire si les co-occurences majoritaires sont en faible nombre – important si elle est chaotique.

A.3.10 *Difference variance*

Nom(s)

– *Difference variance* [Haralick *et al.*, 1973].

Formule(s)

Dans [Haralick *et al.*, 1973] :

$$f_{dvar} = \text{variance de } \mathbf{C}_{x-y} \quad . \quad (\text{A.29})$$

Du fait du peu d’information disponible pour le calcul de cet indice, de nombreux travaux ont là encore recours à une formule erronée. Dans [Gipp *et al.*, 2008], on peut trouver les formules suivantes, qui présentent une certaine cohérence avec celles utilisées pour le calcul des indices *sum average* et *sum variance* :

$$\mu_{x-y} = \sum_{k=0}^{N_g-1} k \cdot \mathbf{C}_{x-y}(k) \quad (\text{A.30})$$

$$f_{dvar} = \sum_{k=0}^{N_g-1} (k - \mu_{x-y})^2 \cdot \mathbf{C}_{x-y}(k) \quad (\text{A.31})$$

Domaine de valeurs $[0, (N_g - 1)^2/4]$

Interprétation La matrice C_{x-y} étant composée des probabilités d'apparitions de l'ensemble des niveaux de contraste, cet indice caractérise l'étendue des contrastes présents dans la texture. Une texture présentant peu de niveaux de contraste aura ainsi un faible indice variance.

A.3.11 *Difference entropy*

Nom(s)

– *Difference entropy* [Haralick et al., 1973].

Formule(s)

Dans [Haralick et al., 1973] :

$$f_{dent} = - \sum_{k=0}^{N_g-1} C_{x-y}(k) \cdot \log(C_{x-y}(k)) \quad . \quad (\text{A.32})$$

Domaine de valeurs $[0, \log(N_g)]$

Interprétation Cet indice mesure le degré de prédictibilité des contrastes présents dans la texture.

A.3.12 *Autres mesures de corrélation*

Haralick définit trois mesures de corrélation supplémentaires, *information measure of correlation 1* & *2* (f_{imc1} et f_{imc2}) et *maximal correlation coefficient* (f_{mcc}) :

$$f_{imc1} = \frac{f_{ent} - HXY1}{\max(HX, HY)} \quad , \quad (\text{A.33})$$

$$f_{imc2} = \sqrt{1 - \exp(-2 \cdot (HXY2 - f_{ent}))} \quad , \quad (\text{A.34})$$

$$f_{mcc} = \sqrt{\text{seconde plus grande valeur propre de } Q} \quad , \quad (\text{A.35})$$

avec HX et XY les entropies respectives de C_x et C_y et :

$$HXY1 = - \sum_{i \in G} \sum_{j \in G} C(i, j) \cdot \log(C_x(i, j) \cdot C_y(i, j)) \quad , \quad (\text{A.36})$$

$$HXY2 = - \sum_{i \in G} \sum_{j \in G} C_x(i) \cdot C_y(j) \cdot \log(C_x(i) \cdot C_y(j)) \quad , \quad (\text{A.37})$$

$$Q(i, j) = \sum_{k \in G} \frac{C(i, k) \cdot C(j, k)}{C_x(i) \cdot C_y(k)} \quad \forall i, j \in G \quad . \quad (\text{A.38})$$

Ces caractéristiques sont cependant rarement utilisées, car sources d'instabilité numérique [Boland, 1999]. En effet, au même titre que la mesure de corrélation (voir section

A.3. CARACTÉRISTIQUES DE TEXTURES

A.3.3), f_{imc1} et f_{mcc} sont indéfinies notamment lorsque la région caractérisée est composée d'un unique niveau de gris. f_{imc2} reste cependant une mesure de corrélation valide [Linfoot, 1957], son domaine de définition étant $[0, 1]$.

A.3. CARACTÉRISTIQUES DE TEXTURES

Bibliographie

- [Achanta *et al.*, 2012] ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P. et SÜSS-TRUNK, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–82.
- [Aizerman *et al.*, 1964] AIZERMAN, A., BRAVERMAN, E. M. et ROZONER, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837.
- [Akaike, 1969] AKAIKE, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21(1):243–247.
- [Alata et Olivier, 2003] ALATA, O. et OLIVIER, C. (2003). Choice of a 2-D causal autoregressive texture model using information criteria. *Pattern Recognition Letters*, 24(9-10):1191–1201.
- [Ambroise et Govaert, 2000] AMBROISE, C. et GOVAERT, G. (2000). EM Algorithm for Partially Known Labels. In *Data Analysis, Classification, and Related Methods*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 161–166. Springer Berlin Heidelberg.
- [Arbeláez et Cohen, 2004] ARBELÁEZ, P. A. et COHEN, L. D. (2004). Energy partitions and image segmentation. *Journal of Mathematical Imaging and Vision*, 20(1-2):43–57.
- [Besag, 1974] BESAG, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36(2):192–236.
- [Beyer et Goldstein, 1999] BEYER, K. et GOLDSTEIN, J. (1999). When is "nearest neighbor" meaningful? In *International Conference on Database Theory*, pages 217–235.
- [Bharati *et al.*, 2004] BHARATI, M. H., LIU, J. et MACGREGOR, J. F. (2004). Image texture analysis : methods and comparisons. *Chemometrics and Intelligent Laboratory Systems*, 72(1):57–71.
- [Boland, 1999] BOLAND, M. V. (1999). Haralick texture features. http://murphyllab.web.cmu.edu/publications/boland/boland_node26.html [2013-03-12].
- [Bougleux, 2007] BOUGLEUX, S. (2007). *Reconstruction, détection et régularisation de données discrètes*. Thèse de doctorat, Université de Caen.
- [Bougleux et Elmoataz, 2005] BOUGLEUX, S. et ELMOATAZ, A. (2005). Image smoothing and segmentation by graph regularization. *Advances in Visual Computing*, pages 745–752.
- [Bougleux *et al.*, 2007] BOUGLEUX, S., ELMOATAZ, A. et MELKEMI, M. (2007). Discrete regularization on weighted graphs for image and mesh filtering. In *Scale Space and*

- Variational Methods in Computer Vision*, volume 4485 de *Lecture Notes in Computer Science*, pages 128–139. Springer Berlin Heidelberg.
- [Bougleux *et al.*, 2008] BOUGLEUX, S., ELMOATAZ, A. et MELKEMI, M. (2008). Local and Nonlocal Discrete Regularization on Weighted Graphs for Image and Mesh Processing. *International Journal of Computer Vision*, 84(2):220–236.
- [Bouman et Liu, 1991] BOUMAN, C. et LIU, B. (1991). Multiple resolution segmentation of textured images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):99–113.
- [Brodatz, 1966] BRODATZ, P. (1966). *Textures : a photographic album for artists and designers*. Dover New York.
- [Buades *et al.*, 2005] BUADES, A., COLL, B. et MOREL, J.-M. (2005). A Non-Local Algorithm for Image Denoising. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65. IEEE.
- [Cardoso et Corte-Real, 2005] CARDOSO, J. S. et CORTE-REAL, L. (2005). Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, 14(11):1773–82.
- [Chambolle, 2005] CHAMBOLLE, A. (2005). Total variation minimization and a class of binary MRF models. In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 de *Lecture Notes in Computer Science*, pages 136–152. Springer-Verlag.
- [Chan *et al.*, 2001] CHAN, T. F., OSHER, S. et SHEN, J. (2001). The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2):231–41.
- [Chan *et al.*, 2000] CHAN, T. F., SANDBERG, B. Y. et VESE, L. A. (2000). Active Contours without Edges for Vector-Valued Images. *Journal of Visual Communication and Image Representation*, 11(2):130–141.
- [Chan et Vese, 2001] CHAN, T. F. et VESE, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277.
- [Chan et Langford, 1982] CHAN, Y. et LANGFORD, R. (1982). Spectral estimation via the high-order Yule-Walker equations. *Transactions on Acoustics, Speech, and Signal Processing*, 30(5):689–698.
- [Chang et Kuo, 1993] CHANG, T. et KUO, C.-C. J. (1993). Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, 2(4):429–41.
- [Chapelle et Schölkopf, 2006] CHAPELLE, O. et SCHÖLKOPF, B. (2006). *Semi-Supervised Learning*. MIT press Cambridge.
- [Chellappa *et al.*, 1985] CHELLAPPA, R., CHATTERJEE, S. et BAGDAZIAN, R. (1985). Texture synthesis and compression using Gaussian-Markov random field models. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15(2):298–303.
- [Chen et Chen, 1999] CHEN, C.-C. et CHEN, C.-C. (1999). Filtering methods for texture discrimination. *Pattern Recognition Letters*, 20(8):783–790.
- [Clausi, 2001] CLAUSI, D. A. (2001). Comparison and fusion of cooccurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery. *Atmosphere-Ocean*, 39(3):183–194.

- [Coifman *et al.*, 1992] COIFMAN, R. R., MEYER, Y. et WICKERHAUSER, M. V. (1992). Wavelet analysis and signal processing. *In Wavelets and their Applications*, pages 153–178.
- [Coifman et Wickerhauser, 1992] COIFMAN, R. R. et WICKERHAUSER, M. V. (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718.
- [Connors *et al.*, 1984] CONNERS, R., TRIVEDI, M. et HARLOW, C. (1984). Segmentation of a high-resolution urban scene using texture operators. *Computer Vision, Graphics, and Image Processing*, 25(3):273–310.
- [Cortes et Vapnik, 1995] CORTES, C. et VAPNIK, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.
- [Couprie *et al.*, 2013] COUPRIE, C., GRADY, L., NAJMAN, L., PESQUET, J.-C. et TALBOT, H. (2013). Dual Constrained TV-based Regularization on Graphs. *SIAM Journal on Imaging Sciences*, 6(3):1246–1273.
- [Cross et Jain, 1983] CROSS, G. R. et JAIN, A. K. (1983). Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(1):25–39.
- [Dal Moro *et al.*, 2006] DAL MORO, F., ABATE, A., LANCKRIET, G. R. G., ARANDJELOVIC, G., GASPARELLA, P., BASSI, P., MANCINI, M. et PAGANO, F. (2006). A novel approach for accurate prediction of spontaneous passage of ureteral stones : support vector machines. *Kidney international*, 69(1):157–60.
- [Daugman, 1987] DAUGMAN, J. G. (1987). Image Analysis And Compact Coding By Oriented 2D Gabor Primitives. *In BARRETT, E. B. et PEARSON, J. J., éditeurs : Proceedings of the Conference on Image Understanding and the Man-Machine Interface*, pages 19–30.
- [Davis *et al.*, 1979] DAVIS, L. S., JOHNS, S. a. et AGGARWAL, J. K. (1979). Texture analysis using generalized co-occurrence matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(3):251–9.
- [Debure et Kubato, 1998] DEBURE, K. et KUBATO, T. (1998). Autoregressive texture segmentation and synthesis for wavelet image compression. *In Proceedings of the Conference on Image and Multidimensional Digital Signal Processing*, pages 131–135.
- [Deguchi et Morishita, 1978] DEGUCHI, K. et MORISHITA, I. (1978). Texture Characterization and Texture-Based Image Partitioning Using Two-Dimensional Linear Estimation Techniques. *IEEE Transactions on Computers*, C-27(8):739–745.
- [Derin et Elliott, 1987] DERIN, H. et ELLIOTT, H. (1987). Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39–55.
- [Descombes *et al.*, 1997] DESCOMBES, X., MORRIS, R., ZERUBIA, J. et BERTHOD, M. (1997). Maximum likelihood estimation of Markov Random Field parameters using Markov Chain Monte Carlo algorithms. *In Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 1223 de *Lecture Notes in Computer Science*, pages 133–148. Springer Berlin Heidelberg.

- [DiCarlo *et al.*, 2012] DICARLO, J. J., ZOCCOLAN, D. et RUST, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415–34.
- [Efros et Leung, 1999] EFROS, A. et LEUNG, T. (1999). Texture synthesis by non-parametric sampling. *In Proceedings of the International Conference on Computer Vision*, pages 1033–1038. IEEE.
- [Elmoataz *et al.*, 2008] ELMOATAZ, A., LEZORAY, O. et BOUGLEUX, S. (2008). Nonlocal discrete regularization on weighted graphs : a framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7):1047–1060.
- [Ergen, 2012] ERGEN, B. (2012). Texture Analysis and Classification Using Autoregressive Signal Modeling Method. *In Proceedings of the International Conference on Communications and Information Technology*, pages 107–110.
- [Faucheux *et al.*, 2013] FAUCHEUX, C., OLIVIER, J. et BONÉ, R. (2013). Graph-Based Regularization of Binary Classifiers for Texture Segmentation. *In Computer Analysis of Images and Patterns*, volume 8048 de *Lecture Notes in Computer Science*, pages 310–318. Springer Berlin Heidelberg.
- [Faucheux *et al.*, 2012a] FAUCHEUX, C., OLIVIER, J., BONÉ, R. et MAKRIS, P. (2012a). Segmentation d’images échographique 2D et 3D par régularisation de graphe basée texture. *In CCompression et REprésentation des Signaux Audiovisuels*, pages 127–132.
- [Faucheux *et al.*, 2012b] FAUCHEUX, C., OLIVIER, J., BONÉ, R. et MAKRIS, P. (2012b). Texture-based graph regularization process for 2D and 3D ultrasound image segmentation. *In Proceedings of the International Conference on Image Processing*, pages 2333–2336. IEEE.
- [Fehr et Burkhardt, 2008] FEHR, J. et BURKHARDT, H. (2008). 3D rotation invariant local binary patterns. *In Proceedings of the International Conference on Pattern Recognition*, pages 1–4. Ieee.
- [Felzenszwalb et Huttenlocher, 2004] FELZENSZWALB, P. F. et HUTTENLOCHER, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181.
- [Fix et Hodges, 1951] FIX, E. et HODGES, J. (1951). Discriminatory Analysis - Nonparametric Discrimination : Consistency Properties. Rapport technique, USAF School of Aviation Medicine.
- [Gabor, 1946] GABOR, D. (1946). Theory of communication. *Journal of the Institute of Electrical Engineers*, 93:429–457.
- [Gilboa et Osher, 2007] GILBOA, G. et OSHER, S. (2007). Nonlocal Linear Image Regularization and Supervised Segmentation. *Multiscale Modeling and Simulation*, 6(2):595–630.
- [Gipp *et al.*, 2008] GIPP, M., MARCUS, G. et HARDER, N. (2008). Accelerating the Computation of Haralick’s Texture Features using Graphics Processing Units (GPUs). *In Proceedings of the World Congress on Engineering*, pages 587–592.
- [Golub et Van Loan, 1996] GOLUB, G. H. et VAN LOAN, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press.
- [Gomory et Hu, 1961] GOMORY, R. E. et HU, T. C. (1961). Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570.

- [Grady, 2006] GRADY, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783.
- [Grady et Jolly, 2008] GRADY, L. et JOLLY, M.-P. (2008). Weights and topology : a study of the effects of graph construction on 3D image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, 11:153–161.
- [Grossmann et Morlet, 1984] GROSSMANN, A. et MORLET, J. (1984). Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736.
- [Haralick, 1979] HARALICK, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804.
- [Haralick et al., 1973] HARALICK, R. M., SHANMUGAM, K. et DINSTEIN, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621.
- [Hartman et al., 1990] HARTMAN, E. J., KEELER, J. D. et KOWALSKI, J. M. (1990). Layered Neural Networks with Gaussian Hidden Units as Universal Approximations. *Neural Computation*, 2(2):210–215.
- [Haxhimusa et Kropatsch, 2004] HAXHIMUSA, Y. et KROPATSCH, W. (2004). Segmentation graph hierarchies. In *Structural, Syntactic, and Statistical Pattern Recognition*, volume 3138 de *Lecture Notes in Computer Science*, pages 343–351. Springer.
- [Hidane et al., 2012] HIDANE, M., LÉZORAY, O. et ELMOATAZ, A. (2012). Nonlinear Multilayered Representation of Graph-Signals. *Journal of Mathematical Imaging and Vision*, 45(2):114–137.
- [Hiremath et al., 2012] HIREMATH, S., TOLPEKIN, V. a., HEIJDEN, G. et STEIN, A. (2012). Segmentation of Rumex obtusifolius using Gaussian Markov random fields. *Machine Vision and Applications*, 24(4):845–854.
- [Hornik et al., 1989] HORNIK, K., STINCHCOMBE, M. et WHITE, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- [Howarth et Stefan, 2004] HOWARTH, P. et STEFAN, R. (2004). Evaluation of Texture Features for Content-Based Image Retrieval. In *Image and Video Retrieval*, volume 3115 de *Lecture Notes in Computer Science*, pages 326–334. Springer Berlin Heidelberg.
- [Jain et Farrokhnia, 1991] JAIN, A. K. et FARROKHANIA, F. (1991). Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186.
- [Jeng et Woods, 1990] JENG, F.-C. et WOODS, J. (1990). Simulated annealing in compound Gaussian random fields (image processing). *IEEE Transactions on Information Theory*, 36(1):94–107.
- [Julesz, 1962] JULESZ, B. (1962). Visual Pattern Discrimination. *IEEE Transactions on Information Theory*, 8(2):84–92.
- [Kashyap et Khotanzad, 1986] KASHYAP, R. et KHOTANZAD, A. (1986). A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):472–481.
- [Kolmogorov et Zabih, 2004] KOLMOGOROV, V. et ZABIH, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–59.

- [Kruskal, 1956] KRUSKAL, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50.
- [Kumar *et al.*, 2011] KUMAR, P., MA, X., LIU, X., JIA, J., BUCONG, H., XUE, Y., LI, Z. R., YANG, S. Y., WEI, Y. Q. et CHEN, Y. Z. (2011). Effect of training data size and noise level on support vector machines virtual screening of genotoxic compounds from large compound libraries. *Journal of Computer-Aided Molecular Design*, 25(5):455–67.
- [Laine et Fan, 1993] LAINE, A. et FAN, J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1186–1191.
- [Larsen et Hansen, 1994] LARSEN, J. et HANSEN, L. K. (1994). Generalization performance of regularized neural network models. *In Proceedings of Workshop on Neural Networks for Signal Processing*, pages 42–51. Ieee.
- [Lei *et al.*, 2013] LEI, Z., PIETIKÄINEN, M. et LI, S. Z. (2013). Learning Discriminant Face Descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [Leung et Malik, 2001] LEUNG, T. et MALIK, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44.
- [Lezoray *et al.*, 2007] LEZORAY, O., ELMOATAZ, A. et BOUGLEUX, S. (2007). Graph regularization for color image processing. *Computer Vision and Image Understanding*, 107(1-2):38–55.
- [Lezoray *et al.*, 2008] LEZORAY, O., TA, V.-T. et ELMOATAZ, A. (2008). Nonlocal graph regularization for image colorization. *In Proceedings of the International Conference on Pattern Recognition*, pages 1–4.
- [Lin *et al.*, 2008] LIN, S.-W., LEE, Z.-J., CHEN, S.-C. et TSENG, T.-Y. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing*, 8(4):1505–1512.
- [Linfoot, 1957] LINFOOT, E. (1957). An informational measure of correlation. *Information and Control*, 1(1):85–89.
- [Maier *et al.*, 2009] MAIER, M., HEIN, M. et von LUXBURG, U. (2009). Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. *Theoretical Computer Science*, 410(19):1749–1764.
- [Maier *et al.*, 2008] MAIER, M., LUXBURG, U. V. et HEIN, M. (2008). Influence of graph construction on graph-based clustering measures. *In Advances in Neural Information Processing Systems*, pages 1025–1032.
- [Malfait et Roose, 1997] MALFAIT, M. et ROOSE, D. (1997). Wavelet-based image denoising using a Markov random field a priori model. *IEEE Transactions on Image Processing*, 6(4):549–65.
- [Mallat, 1989] MALLAT, S. G. (1989). A Theory for Multiresolution Signal Decomposition : The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693.

- [Manjunath et Chellappa, 1991] MANJUNATH, B. et CHELLAPPA, R. (1991). Unsupervised texture segmentation using Markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):478–482.
- [Manjunath et Ma, 1996a] MANJUNATH, B. et MA, W. (1996a). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842.
- [Manjunath et Ma, 1996b] MANJUNATH, B. et MA, W. (1996b). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842.
- [Mao et Jain, 1992] MAO, J. et JAIN, A. K. (1992). Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188.
- [Maragos *et al.*, 1984] MARAGOS, P., SCHAFER, R. et MERSEREAU, R. (1984). Two-Dimensional Linear Prediction and Its Application to Adaptive Predictive. *Transactions on Acoustics, Speech and Signal Processing*, 32(6):1213–1229.
- [Matyas, 1965] MATYAS, J. (1965). Random optimization. *Automation and Remote Control*, 26(2):246–253.
- [McAullife, 2008] MCAULLIFE, M. (2008). MIPAV : Haralick Texture Features. <http://mipav.cit.nih.gov/documentation/HTMLAlgorithms/FilterSpatialHaralickTexture.html> [2013-03-12].
- [McCormick et Jayaramamurthy, 1974] MCCORMICK, B. H. et JAYARAMAMURTHY, S. N. (1974). Time series model for texture synthesis. *International Journal of Computer and Information Sciences*, 3(4):329–343.
- [Meyer, 1985] MEYER, Y. (1985). Principe d’incertitude, bases hilbertiennes et algèbres d’opérateurs. *Seminaire Bourbaki*, 28:209–223.
- [Ohanian et Dubes, 1992] OHANIAN, P. P. et DUBES, R. C. (1992). Performance evaluation for four classes of textural features. *Pattern Recognition*, 25(8):819–833.
- [Ojala *et al.*, 1996] OJALA, T., PIETIKÄINEN, M. et HARWOOD, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59.
- [Ojala *et al.*, 2002] OJALA, T., PIETIKAINEN, M. et MAENPAA, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.
- [Olivier *et al.*, 2008] OLIVIER, J., BONÉ, R., ROUSSELLE, J.-j. et CARDOT, H. (2008). Active Contours Driven by Supervised Binary Classifiers for Texture Segmentation. *Advances in Visual Computing*, 5358:288–297.
- [Pai et Hong, 2005] PAI, P.-F. et HONG, W.-C. (2005). Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electric Power Systems Research*, 74(3):417–425.
- [Parzen, 1968] PARZEN, E. (1968). Multiple time series modelling. Rapport technique D, Stanford University.

- [Paulhac *et al.*, 2008] PAULHAC, L., MAKRIS, P. et RAMEL, J.-y. (2008). Comparison between 2D and 3D Local Binary Pattern Methods for Characterisation of Three-Dimensional Textures. *In Image Analysis and Recognition*, volume 5112 de *Lecture Notes in Computer Science*, pages 670–679. Springer Berlin Heidelberg.
- [Pichler *et al.*, 1996] PICHLER, O., TEUNER, A. et HOSTICKA, B. J. (1996). A comparison of texture feature extraction using adaptive gabor filtering, pyramidal and tree structured wavelet transforms. *Pattern Recognition*, 29(5):733–742.
- [Pierce et Ulaby, 2002] PIERCE, L. et ULABY, F. (2002). SAR speckle reduction using wavelet denoising and Markov random field modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 40(10):2196–2212.
- [Pincus, 2004] PINCUS, Z. (2004). itkHistogramToTextureFeaturesFilter Class Template Reference. http://www.itk.org/Doxygen/html/classitk_1_1Statistics_1_1HistogramToTextureFeaturesFilter.html [2013-03-12].
- [Pons, 2005] PONS, P. (2005). Détection de structures de communautés dans les grands réseaux d’interactions. *In AlgoTel 2005 Septiemes rencontres francophones sur les Aspects Algorithmiques des Telecommunications*, pages 75–78.
- [Randen et Husoy, 1999] RANDEN, T. et HUSOY, J. H. k. n. (1999). Filtering for texture classification : a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310.
- [Rosenblatt, 1958] ROSENBLATT, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408.
- [Rubner et Tomasi, 1999] RUBNER, Y. et TOMASI, C. (1999). Texture-based image retrieval without segmentation. *Proceedings of the International Conference on Computer Vision*, 2:1018–1024.
- [Rudin *et al.*, 1992] RUDIN, L. I., OSHER, S. et FATEMI, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, 60(1-4):259–268.
- [Rumelhart *et al.*, 1986] RUMELHART, D., HINTON, G. et WILLIAMS, R. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.
- [Schmid, 2001] SCHMID, C. (2001). Constructing models for content-based image retrieval. *In Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 39–45. IEEE Comput. Soc.
- [Sharma et Singh, 2001] SHARMA, M. et SINGH, S. (2001). Evaluation of texture methods for image analysis. *In Intelligent Information Systems Conference*, pages 1–8.
- [Shi et Malik, 2000] SHI, J. et MALIK, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.
- [Smith *et al.*, 1994] SMITH, S., BOURGOIN, M., SIMS, K. et VOORHEES, H. (1994). Hand-written character classification using nearest neighbor in large databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):915–919.
- [Stachowiak *et al.*, 2005] STACHOWIAK, G., PODSIADLO, P. et STACHOWIAK, G. (2005). A Comparison of Texture Feature Extraction Methods for Machine Condition Monitoring and Failure Analysis. *Tribology Letters*, 20(2):133–147.

- [Suykens, 2001] SUYKENS, J. (2001). Nonlinear modelling and support vector machines. *In Proceedings of the Instrumentation and Measurement Technology Conference*, volume 1, pages 287–294. IEEE.
- [Ta, 2009] TA, V.-T. (2009). *Equations aux différences partielles définies sur des graphes pour le traitement d’images et de données*. Thèse de doctorat, Université de Caen Basse-Normandie.
- [Ta *et al.*, 2007a] TA, V.-T., BOUGLEUX, S., ELMOATAZ, A. et LÉZORAY, O. (2007a). Nonlocal Anisotropic Discrete Regularization for Image, Data Filtering and Clustering. Rapport technique, Université de Caen.
- [Ta *et al.*, 2007b] TA, V.-T., LEZORAY, O. et ELMOATAZ, A. (2007b). Graph Based Semi and Unsupervised Classification and Segmentation of Microscopic Images. *In Proceedings of the International Symposium on Signal Processing and Information Technology*, pages 1160–1165. IEEE.
- [Ta *et al.*, 2009] TA, V.-T., LÉZORAY, O., ELMOATAZ, A. et SCHÜPP, S. (2009). Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, 42(6):1113–1125.
- [Tang, 1998] TANG, X. (1998). Texture information in run-length matrices. *IEEE Transactions on Image Processing*, 7(11):1602–1609.
- [Tschumperlé et Brun, 2009] TSCHUMPERLÉ, D. et BRUN, L. (2009). Lissage Non Local d’Images Multi-Valuées par Diffusion Anisotrope dans l’Espace des Patches. *In Colloque GRETSI*.
- [Tuceryan et Jain, 1990] TUCERYAN, M. et JAIN, A. K. (1990). Texture segmentation using Voronoi polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):211–216.
- [Tuceryan et Jain, 1993] TUCERYAN, M. et JAIN, A. K. (1993). Texture analysis. *In Handbook of Pattern Recognition and Computer Vision*, pages 235–276. World Scientific.
- [Turner, 1986] TURNER, M. (1986). Texture discrimination by Gabor functions. *Biological Cybernetics*, 82(2-3):71–82.
- [Unser, 1995] UNSER, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing*, 4(11):1549–60.
- [Ursani *et al.*, 2007] URSANI, A. A., KPALMA, K. et RONSIN, J. (2007). Texture features based on Fourier transform and Gabor filters : an empirical comparison. *In International Conference on Machine Vision*, pages 67–72. IEEE.
- [Vapnik, 1995] VAPNIK, V. (1995). *The nature of statistical learning theory*. Springer.
- [Varma et Zisserman, 2002] VARMA, M. et ZISSERMAN, A. (2002). Classifying images of materials : Achieving viewpoint and illumination independence. *In European Conference on Computer Vision*, pages 255–271.
- [Varma et Zisserman, 2005] VARMA, M. et ZISSERMAN, A. (2005). A Statistical Approach to Texture Classification from Single Images. *International Journal of Computer Vision*, 62(1/2):61–81.
- [Véronis, 2003] VÉRONIS, J. (2003). HyperLex : Cartographie lexicale pour la recherche d’informations. *In Actes de la conférence annuelle sur le Traitement Automatique des Langues Naturelles*, pages 1–25.

- [Voisin, 2012] VOISIN, A. (2012). *Classification supervisée d'images d'observation de la Terre à haute résolution par utilisation de méthodes markoviennes*. Thèse de doctorat, Université de Nice - Sophia Antipolis.
- [von Luxburg, 2007] von LUXBURG, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.
- [Wang et al., 2013] WANG, K., BICHOT, C.-E., ZHU, C. et LI, B. (2013). Pixel to Patch Sampling Structure and Local Neighboring Intensity Relationship Patterns for Texture Classification. *Signal Processing Letters*, 20(9):853–856.
- [Wang et He, 1990] WANG, L. et HE, D.-C. (1990). Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910.
- [Wang et Liu, 1999] WANG, L. et LIU, J. (1999). Texture classification using multiresolution Markov random field models. *Pattern Recognition Letters*, 20(2):171–182.
- [Wang et Siskind, 2001] WANG, S. et SISKIND, J. (2001). Image segmentation with minimum mean cut. In *International Conference on Computer Vision*, volume 1, pages 517–524. IEEE Comput. Soc.
- [Weijer et Schmid, 2006] WEIJER, J. V. D. et SCHMID, C. (2006). Coloring Local Feature Extraction. In *Proceedings of the European Conference on Computer Vision*, volume 3952, pages 334–348.
- [Weiss, 1999] WEISS, Y. (1999). Segmentation using eigenvectors : a unifying view. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2:975–982.
- [Weszka, 1976] WESZKA, J. S. (1976). A Comparative Study of Texture Measures for Terrain Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6(4):269–285.
- [Wu et Noé, 2010] WU, H. et NOÉ, F. (2010). Maximum a posteriori estimation for Markov chains based on Gaussian Markov random fields. *Procedia Computer Science*, 1(1):1665–1673.
- [Wu et Leahy, 1993] WU, Z. et LEAHY, R. M. (1993). An optimal graph theoretic approach to data clustering : theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113.
- [Yates et al., 2003] YATES, D., GANGOPADHYAY, S., RAJAGOPALAN, B. et STRZEPEK, K. (2003). A technique for generating regional climate scenarios using a nearest-neighbor algorithm. *Water Resources Research*, 39(7).
- [Zahn, 1971] ZAHN, C. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 100(1):68–86.
- [Zhang et al., 2009] ZHANG, Q., SHAN, G., DUAN, X. et ZHANG, Z. (2009). Parameters optimization of Support Vector Machine based on Simulated Annealing and Genetic Algorithm. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 1302–1306. IEEE.
- [Zhao et Pietikäinen, 2006] ZHAO, G. et PIETIKÄINEN, M. (2006). Dynamic Texture Recognition Using Volume Local Binary Patterns. In *Dynamical Vision*, volume 4358 de *Lecture Notes in Computer Science*, pages 165–177. Springer Berlin Heidelberg.

BIBLIOGRAPHIE

- [Zhu, 2005] ZHU, X. (2005). Semi-Supervised Learning Literature Survey Contents. Rapport technique, Computer Sciences, University of Wisconsin-Madison.
- [Zou *et al.*, 1995] ZOU, C. R., PLOTKIN, E. I., SWAMY, M. N. S. et HE, Z. Y. (1995). Recursive-in-order least-squares parameter estimation algorithm for 2-D noncausal Gaussian Markov random field model. *Circuits Systems and Signal Processing*, 14(1):87–110.

Résumé :

Dans cette thèse, nous nous intéressons à un récent algorithme de segmentation d'images basé sur un processus de régularisation de graphes. L'objectif d'un tel algorithme est de calculer une fonction indicatrice de la segmentation qui satisfait un critère de régularité ainsi qu'un critère d'attache aux données. La particularité de cette approche est de représenter les images à l'aide de graphes de similarité. Ceux-ci permettent d'établir des relations entre des pixels non-adjacents, et ainsi de procéder à un traitement non-local des images. Afin d'en améliorer la précision, nous combinons cet algorithme à une seconde approche non-locale : des caractéristiques de textures. Un nouveau terme d'attache aux données est dans un premier temps développé. Inspiré des travaux de Chan et Vese, celui-ci permet d'évaluer l'homogénéité d'un ensemble de caractéristiques de textures. Dans un second temps, nous déléguons le calcul de l'attache aux données à un classificateur supervisé. Entraîné à reconnaître certaines classes de textures, ce classificateur permet d'identifier les caractéristiques les plus pertinentes, et ainsi de fournir une modélisation plus aboutie du problème. Cette seconde approche permet par ailleurs une segmentation multiclasse. Ces deux méthodes ont été appliquées à la segmentation d'images texturées 2D et 3D.

Mots clés :

Segmentation, analyse de textures, graphe, régularisation, classificateur supervisé, réseau de neurones artificiels, analyse d'images médicales, images 3D.

Abstract:

In this thesis, we improve a recent image segmentation algorithm based on a graph regularization process. The goal of this method is to compute an indicator function that satisfies a regularity and a fidelity criteria. Its particularity is to represent images with similarity graphs. This data structure allows relations to be established between similar pixels, leading to non-local processing of the data. In order to improve this approach, combine it with another non-local one: the texture features. Two solutions are developed, both based on Haralick features. In the first one, we propose a new fidelity term which is based on the work of Chan and Vese and is able to evaluate the homogeneity of texture features. In the second method, we propose to replace the fidelity criteria by the output of a supervised classifier. Trained to recognize several textures, the classifier is able to produce a better modelization of the problem by identifying the most relevant texture features. This method is also extended to multiclass segmentation problems. Both are applied to 2D and 3D textured images.

Keywords:

Segmentation, textures analysis, graph, regularization, supervised classifier, artificial neural network, medical images analysis, 3D images.