



HAL
open science

Analyse de textures dans l'espace hyperspectral par des méthodes probabilistes

Guillaume Rellier

► **To cite this version:**

Guillaume Rellier. Analyse de textures dans l'espace hyperspectral par des méthodes probabilistes. Interface homme-machine [cs.HC]. Université Nice Sophia Antipolis, 2002. Français. NNT: . tel-00505898

HAL Id: tel-00505898

<https://theses.hal.science/tel-00505898>

Submitted on 26 Jul 2010

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.

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS - UFR SCIENCES

Ecole Doctorale : Sciences et Technologies de l'Information et de la
Communication

THESE

présentée pour obtenir le titre de

DOCTEUR en sciences de l'Université de Nice Sophia-Antipolis

Spécialité : Traitement des images

par

Guillaume RELLIER

Analyse de textures dans l'espace hyperspectral par des
méthodes probabilistes

Soutenue publiquement le 8 Novembre 2002 devant le jury composé de :

A. Bijaoui	OCA	Président
F. Heitz	ENSPS	Rapporteur
J. M. F. Moura	Carnegie Mellon Univ., USA	Rapporteur
X. Descombes	INRIA	Examineur
F. Falzon	Alcatel Space	Examineur
J. Zerubia	INRIA	Examineur

14h30 à L'ESSI

Remerciements

Avant de commencer ce mémoire, quelques lignes pour remercier ceux qui ont permis qu'il se termine aussi.

Les premières personnes que je tiens à remercier sont les membres de mon jury. Merci à monsieur Albert Bijaoui d'avoir bien voulu présider celui-ci, et à messieurs Fabrice Heitz et José Moura d'avoir accepté de faire le travail sans doute peu reposant de lire ce mémoire et d'en rédiger les rapports. Quant aux examinateurs, Xavier Descombes, Josiane Zerubia et Frédéric Falzon, je les remercie à double titre : d'une part pour avoir participé à l'examen de ma soutenance, et d'autre part pour avoir encadré mon travail pendant les trois ans de préparation.

D'un point de vue pratique, je tiens également à remercier Alcatel Space Cannes et la région Provence-Alpes-Côte d'Azur qui ont cofinancé ma bourse de thèse.

Merci aux membres du projet Ariana pour leur apport scientifique, et plus si affinités. Merci à Corinne, et désolé pour les ordres de missions faits au dernier moment.

Je remercie chaleureusement les gens de l'Inria avec qui j'ai partagé de bons moments, sans doute autour d'un verre : Radu, Oscar, Nath, Lolo, Frank, Alex, Laurent, Caroline, Mathias, Laurențiu, Christophe, Caroline, Dédé, et d'autres... La même chose pour les gens *du dehors* que j'ai croisés après 18h : Eva, Steph, Zboub, Matthieu, Paul et Hugues, les traboucaïres, et une pensée pour l'Adapa.

Enfin, bien sûr, un très gros merci à ma famille : mes parents, Olivier, Nathalie et Valentin, les mamies (et les papis) et tous les autres... Pour l'ensemble de leur œuvre.

Table des matières

Introduction	1
Chapitre 1. Contexte de l'étude et problématique	5
1.1 Nature et caractéristiques des images de télédétection	6
1.2 Analyse de la texture	9
1.2.1 Analyse de la texture: pourquoi?	11
1.2.2 Analyse de la texture: comment?	12
1.3 Classification	15
1.4 Nécessité et justification de la réduction de l'espace	18
1.4.1 Nécessité	18
Efficacité algorithmique	18
Problèmes d'estimation	19
1.4.2 Justification	19
Renforcement de l'hypothèse de normalité	19
Corrélation	19
1.5 Conclusion	20
Chapitre 2. Analyse de textures multivariées par des modèles markoviens	23
2.1 Modèles markoviens	24
2.2 Modèle markovien gaussien	25
2.3 Le champ de Markov gaussien multivarié	25
2.3.1 Loi conditionnelle	26
2.3.2 Loi globale	27
2.3.3 Caractéristiques des espaces de grande dimension	28
2.3.4 Domaine de définition des paramètres	30
2.3.5 Domaine de définition pour le modèle en 4-connexité anisotrope . . .	31
2.3.6 Corrélations du modèle	33
2.3.7 Estimation des paramètres du champ	34

	Maximum de vraisemblance	35
	Maximum de pseudo-vraisemblance	37
	Estimateur MMSE	38
2.3.8	Simulations	39
	Qualité des estimateurs	39
	Adéquation du modèle	40
2.4	Modèle markovien gaussien 3D	43
	2.4.1 Loi conditionnelle et loi globale	44
	2.4.2 Estimation des paramètres	46
	2.4.3 Restriction à un modèle spectral	47
2.5	Partition du spectre en zones homogènes	49
	2.5.1 Elimination des bandes non informatives	50
	2.5.2 Ajout de bornes	50
	2.5.3 Algorithme	51
2.6	Conclusion	53
Chapitre 3. Réduction d'espace		55
3.1	Approches générales	56
	3.1.1 Analyse en composantes principales	56
	3.1.2 <i>Minimum Noise Fraction</i>	57
3.2	Approches pour la classification	58
	3.2.1 Analyse discriminante	59
	3.2.2 Poursuite de projection	60
3.3	Méthode sémantique-statistique	60
	3.3.1 Analyse de mélange	60
	3.3.2 Résolution de l'analyse de mélange à A connue	61
	3.3.3 Calcul de \hat{y}	62
	3.3.4 Utilisation de l'analyse de mélange pour la réduction d'espace	64
	3.3.5 Recherche des spectres purs	64
	3.3.6 Utilisation d'un espace privilégié	65
	3.3.7 Résultats	67
	Images de synthèse	67
	Images réelles	67
3.4	Conclusion	75

Chapitre 4. Poursuite de projection	77
4.1 Présentation générale	78
4.2 Algorithmes approchés	79
4.2.1 Poursuite de projection dite séquentielle	79
4.2.2 Recherche par croissance d'espace	82
4.2.3 Comparaison des deux algorithmes	83
4.3 Choix de l'indice de projection : modélisation gaussienne non contextuelle	84
Mise en relation de la règle de classification et de l'indice de projection	85
4.4 Choix de l'indice de projection : modélisation markovienne gaussienne	86
4.4.1 Mise en relation de la règle de classification et de l'indice de projection	86
4.4.2 Problème à plusieurs classes	88
4.4.3 Calcul de l'indice de projection	88
4.4.4 Indices de projection globaux	89
4.5 Conclusion	90
Chapitre 5. Application	91
5.1 Classification de texture	92
5.2 Classification : Maximum de vraisemblance	92
5.2.1 Méthode locale	92
5.2.2 Méthode par blocs	94
5.2.3 Réduction de dimension	95
5.3 Classification : Ajout de bandes de textures	95
5.3.1 Description de l'algorithme	95
5.3.2 Paramètres envisageables	95
5.3.3 Calcul des paramètres	97
5.3.4 Retour sur le calcul de l'indice de projection pour la poursuite de projection	99
Approximation de la loi de probabilité du paramètre de texture	101
Distance	105
Poursuite de projection sur les paramètres de texture	105
5.3.5 Règle de classification	106
5.4 Résultats et comparaisons	107
5.4.1 Analyse de texture grâce aux matrices de cooccurrence	107
5.4.2 Récapitulatif des algorithmes testés	109
5.4.3 Images de synthèse	109
Résultats de ABmg	110
Résultats de MVmg	110

Comparaisons	110
5.4.4 Collage de textures extraites d'images AVIRIS	114
5.4.5 Images réelles AVIRIS	119
5.5 Conclusion	126
Conclusion et perspectives	129
Annexe A. Propriétés de la matrice A	133
A.1 Valeurs propres et déterminant	133
A.1.1 Cas monospectral	133
A.1.2 Cas multispectral	134
A.2 Dérivée	135
A.3 Matrice B du champ 3D	136
Bibliographie	137

Table des figures

1.1	Concept illustré du capteur AVIRIS (©JPL).	8
1.2	2 réponses spectrales d'une image du capteur AVIRIS : intensité radiométrique en fonction de la bande de longueur d'onde mesurée. En trait plein : pixel de végétation ; en pointillés : pixel d'eau.	9
1.3	4 bandes d'un extrait d'image AVIRIS : a) bande 20, b) bande 50, c) bande 100, d) bande 170.	10
1.4	2 exemples de texture : a) micro-texture, b) macro-texture.	11
1.5	Image de la matrice de covariance d'une scène AVIRIS : l'intensité en un point (i, j) de l'image est la covariance entre les mesures dans les bandes de longueur d'onde i et j , $c_{ij} = E\{x_s^i x_s^j\}$	20
2.1	Domaine de définition pour les coefficients $a_{1,0}$ et $a_{0,1}$. En pointillés, le cas de dimensions infinies ou paires. En traits plein, un cas quelconque.	32
2.2	Données simulées à différents rapports Signal sur Bruit (S/B) (colonne de gauche), simulation à partir des paramètres estimés pour chaque quart de l'image (colonne de droite).	41
2.3	a) Image initiale, b) Image simulée avec les paramètres extraits de chaque quart de l'image a), par l'estimateur MV.	43
2.4	Un site s et ses différents voisins en 6-connexité dans un modèle 3D.	45
2.5	Corrélation théorique entre deux bandes en fonction de leur distance en nombre de bandes, et approximation de celle-ci par le modèle spectral, pour des nombre de paramètres allant de 1 à 5.	49
2.6	Image de la matrice de corrélation d'une scène AVIRIS.	50
2.7	a) Matrice de corrélation de l'image analysée, b) représentation bandes retenues pour la partition, c) et d) représentation des blocs homogènes pour des partitions à 2 groupes (c) et 3 groupes (d).	52

3.1	Exemple en 3 dimensions de 2 spectres purs (flèches bleues) définissant un domaine de mélange (pointillés bleus), sous ensemble de l'espace engendré par ceux-ci (le plan figuré par le parallélogramme en noir).	62
3.2	a) Evolution des valeurs propres pour une image de synthèse (S/B=5.35 dB), b) évolution du cumul de celles-ci pour la même image (en pourcentage du total), c) évolution des valeurs propres pour une image de synthèse (S/B=8.72 dB), d) évolution du cumul de celles-ci pour la même image (en pourcentage du total).	68
3.3	Exemple de nuages de points issus de mélanges : a) 3 composantes pures, b) 4 composantes pures, c) 5 composantes pures, d) 5 composantes pures avec des coefficients non-uniformément distribués.	69
3.4	Extrait d'une scène réelle en zone rurale (affichage de 3 bandes du visible).	71
3.5	Projection du nuage de points correspondant à la figure 3.4 dans trois plans différents : a) plan défini par les bandes 10 et 15, b) plan défini par les bandes 15 et 35, c) plan défini par les bandes 20 et 30.	71
3.6	a) Proportion du spectre pur correspondant à l'eau, b) proportion du spectre pur correspondant à la forêt, c) proportion du spectre pur correspondant au sol nu, d) image en couleur des trois proportions (rouge : sol nu, vert : forêt, bleu : eau).	72
3.7	Extrait d'une scène réelle en zone urbaine (affichage de 3 bandes du visible).	73
3.8	Projection du nuage de points de l'image 3.7 et des 4 spectres purs, sur un plan de projection arbitraire.	73
3.9	Projection du nuage de points S_i : a) les proportions dans le plan 1-2, b) dans le plan 1-4, c) dans le plan 2-4, d) dans le plan 3-4.	74
3.10	Proportions des 4 spectres purs estimés.	75
4.1	Illustration de l'algorithme d'ajustement des bornes : pour z_{k-1} et z_{k+1} fixés, la valeur optimale \hat{z}_k de z_k est celle minimisant la fonction $\sigma_{I,k}(z_k)$	81
4.2	Distance optimale estimée en fonction du nombre de bandes pour les deux mises en œuvre de la poursuite de projection séquentielle, dans le cas de deux textures d'une même scène AVIRIS. En bleu : mise en œuvre avec groupes variables, en vert : mise en œuvre avec groupes fixes.	82
4.3	Evolution de la distance optimale trouvée par les deux algorithmes de poursuite, avec le même indice de projection. En bleu, la courbe correspondant à l'algorithme par croissance d'espace, et en vert, l'algorithme de poursuite de projection séquentielle.	84

4.4	Evolution du temps de calcul pour les deux algorithmes de poursuite de projection. En bleu, la courbe correspondant à l'algorithme par croissance d'espace, et en vert, l'algorithme de poursuite de projection séquentielle. . .	85
5.1	Algorithme de classification avec ajout de bandes.	96
5.2	Illustration de l'heuristique : dans le cas du haut, une seule texture est présente, on calcule les paramètres sur tous les échantillons, dans le cas du bas, il y a deux classes, on calcule les paramètres sur la classe la plus présente. .	99
5.3	a) Collage de textures à 3 bandes spectrales simulées par MGMRF, b) image en fausses couleurs des éléments diagonaux de $\hat{\Sigma}$ sans euristique, c) transitions détectées par l'heuristique, d) éléments diagonaux de $\hat{\Sigma}$ avec l'heuristique.	100
5.4	a) Extrait de 3 bandes visibles de l'image AVIRIS de Moffett Field, b) variance conditionnelle σ^2 extraite d'une bande de cette image, c) paramètres d'interaction a extrait d'une bande de cette image.	100
5.5	2 histogrammes du paramètre σ^2 calculés sur une bande du spectre visible de 2 zones de l'image AVIRIS de Moffett Field.	101
5.6	a) Une bande de l'image analysée par une matrice de cooccurrence, b) Moyenne, c) Variance, d) Homogénéité, e) Contraste, f) Dissimilarité, g) Entropie, h) Second Moment, i) Corrélation.	108
5.7	Données simulées à différents rapports S/B (colonne de gauche), classification obtenue par la méthode <i>ABmg</i>	111
5.8	Projection du nuage de points correspondant à une zone de transition entre deux textures : a) les deux zones sont distinctes, b) les deux zones sont entremêlées. A noter que les points de vue, différents, ont été choisis parce qu'ils présentaient les nuages de la façon la plus séparée possible.	112
5.9	Données simulées à différents rapports S/B (colonne de gauche), classification obtenue par la méthode <i>MVmg</i>	113
5.10	Les 3 bandes en fausses couleurs de paramètres de moyenne (haut) et dissimilarité (bas) pour les images <i>colSynth1</i> et <i>colSynth10</i> , $d = (0,1)$	115
5.11	3 bandes extraites du spectre visible d'un collage de 4 textures AVIRIS. . .	116
5.12	a) Algorithme <i>PP/MV</i> , b) Algorithme <i>PP/MVmg</i>	117
5.13	Algorithme <i>PP/ABmg</i>	118
5.14	Algorithme <i>ACP/Cooc</i>	118
5.15	Evolution du taux de bonne classification pour les 4 méthodes comparées sur la mosaïque de la figure 5.11.	119
5.16	Evolution du taux de bonne classification pour les 4 méthodes comparées sur l'extrait d'image AVIRIS de Moffett Field.	120

5.17 Classification par 3 méthodes de l'image AVIRIS, en gardant 10 bandes de radiométrie : a) 3 bandes de l'image originale, b) *PP/MV*, c) *PP/ABmg* (5 bandes de texture), d) *ACP/Cooc*. 121

5.18 Influence du nombre de bandes de texture et du nombre de degrés de liberté dans l'approximation de la loi du paramètre de texture. a) Classification convenable (40 bandes de radiométrie, 3 bandes de texture), b) classification avec un nombre de degrés de liberté élevé, c) classification avec un nombre de bandes de texture élevé (20 bandes de texture). 123

5.19 a) affichage en fausse couleur des canaux {30,20,10} de la scène AVIRIS, b) zones d'apprentissage, c) zones de validation. 124

5.20 a) affichage en fausse couleur des canaux {30,20,10} de la scène AVIRIS, b) résultat obtenu par *PP/MV*, c) *PP/ABmg* (2 bandes de texture), d) *ACP/Cooc*. 125

5.21 a) Zoom sur une partie de la classification par *PP/ABmg* sans euristique b) *PP/ABmg* avec euristique. 126

Liste des tableaux

2.1	Moyenne et écart type des quotients entre coefficients des matrices de corrélation avec les plus proches voisins et de la matrice de covariance, pour différentes textures et sélections de bandes spectrales	34
2.2	Paramètres réels et estimés des 4 parties (NO, SO, NE, SE) de l'image <i>col-Synth2</i>	42
3.1	Résultat de l'estimation des spectres purs dans le cas d'un mélange à 3 composantes. 5 tests sont effectués à différents rapport S/B. Les deux premières colonnes de résultats donnent la valeur de l'indice pour les spectres purs réels et estimés, la dernière colonne donne l'erreur d'estimation des spectres purs, en pourcentage de la norme moyenne du signal.	70
3.2	Résultat de l'estimation des spectres purs dans le cas d'un mélange à 4 composantes. 5 tests sont effectués à différents rapport S/B. Les deux premières colonnes de résultats donnent la valeur de l'indice pour les spectres purs réels et estimés, la dernière colonne donne l'erreur d'estimation des spectres purs, en pourcentage de la norme moyenne du signal.	70
5.1	Taux de bonne classification et indice κ pour les classifications obtenues par 4 méthodes différentes à partir des images de synthèse à différents rapports S/B.	114

Introduction

Dans cette thèse, on aborde l'analyse de texture pour l'étude de zones urbaines. La texture est une notion complexe dont il n'existe pas vraiment de définition unifiée, acceptée par tous. En général, on appelle texture ce qui, en dehors de la couleur ou du niveau de gris, caractérise l'homogénéité visuelle d'une zone donnée d'une image. C'est un concept spatial, impliquant les variations locales de niveau de gris (ou de couleur), qui devient incontournable quand l'information que l'on cherche à extraire se situe dans les relations entre les pixels de l'image (ie. l'information contextuelle). Or, c'est le cas des zones qui sont l'objet de ce travail : les zones urbaines.

Le but de cette étude est d'établir un modèle qui permette une analyse de texture prenant en compte conjointement l'aspect spatial et l'aspect spectral. En effet, le développement des instruments spectro-imageurs permet aujourd'hui d'avoir à disposition des images, dites *hyperspectrales*, caractérisées par un nombre de canaux largement supérieur à celui des images multispectrales (qui ont en général trois à sept canaux). On désire tirer parti des capacités de ces spectro-imageurs pour améliorer l'analyse spatiale de la texture. On va donc chercher à enrichir la vision habituelle uniquement spatiale de la texture, par l'apport du point de vue spectral, que permet le grand nombre de bandes des images hyperspectrales.

Le premier chapitre de ce mémoire est consacré à la description du contexte particulier de ce travail. On décrit, dans un premier temps, ce que sont les images traitées d'un point de vue pratique : nombre de bandes (ou canaux), résolution, etc. . . Dans un deuxième temps, sont rappelés les principes de l'analyse de texture ainsi que les méthodes les plus populaires pour l'effectuer. On répète cette démarche en ce qui concerne le thème de la classification, qui sera l'objet des applications de l'analyse de texture. Enfin, la dernière partie de ce chapitre traite d'un problème propre au traitement des images ayant un grand nombre de composantes spectrales : la réduction d'espace. Ici, on justifie l'étape de réduction de dimension en s'appuyant sur des propriétés statistiques et physiques de ces images.

Le choix de l'outil des champs de Markov quant à l'analyse de texture est fait en raison de la capacité de ceux-ci à intégrer aisément et intuitivement les notions spatiales et spectrales des images multi-composantes. Le deuxième chapitre est donc consacré d'une part, à un rappel sur les champs de Markov, en particulier sur les champ de Markov gaussiens, et d'autre part, à la présentation d'une adaptation aux images hyperspectrales d'un champ de Markov gaussien multivarié appelé MGMRF, et de différents estimateurs pour les paramètres de ce champ. On rappelle également une adaptation du modèle consistant en un champ markovien gaussien à trois dimensions, dont les variables sont scalaires. La dernière partie de ce chapitre traite d'une méthode de partition du spectre en groupes de bandes adjacentes, permettant de satisfaire les conditions de validité des hypothèses faites en choisissant les modèles.

Le thème de la réduction d'espace, évoqué et justifié à la fin du premier chapitre, est détaillé dans le troisième chapitre. On s'attache, dans une première partie, à décrire des méthodes parmi les plus utilisées pour la détermination d'un sous-espace de projection, en particulier l'analyse en composantes principales et la poursuite de projection. Cette dernière méthode est analysée au quatrième chapitre. Le troisième chapitre s'achève par la description d'une méthode de réduction d'espace originale, fondée sur l'analyse de mélange linéaire, dans laquelle aucune connaissance *a priori* sur les composantes du mélange n'est nécessaire.

La poursuite de projection est un cadre général de recherche de sous-espaces de projection ayant les propriétés désirées par l'utilisateur. Ces propriétés sont traduites sous la forme d'une grandeur appelée *indice de projection*, fonction des données et du sous-espace de projection. Le principe de cette famille d'algorithmes est de déterminer le sous-espace dans lequel l'indice de projection est optimisé. Le quatrième chapitre détaille deux algorithmes de poursuite de projection, aboutissant à des solutions approchées du problème d'optimisation de l'indice de projection. La fin de ce chapitre est consacrée au choix de cet indice. On propose dans ce manuscrit un indice de projection s'appuyant sur la modélisation par champ de Markov gaussien, dérivé de celui proposé par Jimenez et Landgrebe dans [52].

Enfin, le dernier chapitre aborde une application des modèles et des algorithmes étudiés. Deux algorithmes de classification supervisée de textures sont proposés. Le premier est fondé sur le critère du maximum de vraisemblance supposant un modèle markovien gaussien pour les données. L'approche adoptée pour le deuxième algorithme est plus classique, puisqu'il s'agit d'ajouter de nouvelles bandes, caractérisant la texture, aux bandes initiales (ou obtenues par réduction d'espace), et d'opérer la classification sur cette image « augmentée »,

enrichie d'information de texture. Les résultats présentés montrent que le premier algorithme n'est efficace que quand les textures ont des moyennes proches. Quant au deuxième algorithme, il donne de bons taux de classification en comparaison d'algorithmes ne prenant pas en compte la texture, ou la prenant en compte à travers des paramètres de cooccurrence.

Chapitre 1

Contexte de l'étude et problématique

Ce travail s'articule autour de deux aspects que l'on aborde conjointement : l'analyse spectrale et spatiale de la texture. Une analyse spectrale fine est rendue envisageable grâce à l'apparition d'images dites hyperspectrales, que l'on présente dans ce chapitre, ainsi que leurs caractéristiques particulières.

Ensuite, le thème de l'analyse spatiale de la texture est présenté, de même qu'une revue de quelques méthodes efficaces pour le traiter. On se penche également sur le problème de la classification, but final de l'analyse de la texture dans notre cas, en présentant quelques méthodes de classification.

Enfin, la dernière partie de ce chapitre est destinée à justifier une partie du travail présenté par la suite, celle de la réduction d'espace des images hyperspectrales. Des caractéristiques particulières (géométriques, statistiques) des images hyperspectrales sont utilisées à cette fin.

1.1 Nature et caractéristiques des images de télédétection

Une image de télédétection est une image échantillonnée spatialement, acquise grâce à un capteur survolant une scène. Les images monospectrales, ou de niveaux de gris, fournissent, pour chaque échantillon spatial (pixel), une valeur scalaire correspondant au rayonnement électromagnétique incident sur le capteur intégré sur la bande de longueur d'onde mesurée. On parle alors d'images à une seule « bande ». Outre les images monospectrales, il existe des images fournissant des mesures sur plusieurs bandes. C'est le cas des images dites multispectrales, superspectrales et hyperspectrales.

Quand les premières sont apparues, elles désignaient toutes les images qui n'étaient pas monospectrales, sans restriction. On disposait alors de capteurs multispectraux du type SPOT XS ou du *Thematic Mapper* (TM) de Landsat qui ont respectivement 3 et 7 bandes d'acquisition. Le terme multispectral désigne désormais ce type d'image, au nombre de canaux inférieur à 10. De nouvelles images ayant un nombre de composantes supérieur, issues de capteurs récents, aussi bien aéroportés que satellitaires, sont à présent qualifiées de superspectrales ou hyperspectrales. Les images issues des capteurs CASI (capteur aéroporté) [2] ou MERIS (satellitaire) [69] sont des données superspectrales, et celles des capteurs AVIRIS (aéroporté) [79] ou MIVIS (satellitaire) des données hyperspectrales.

La différence entre les images superspectrales et hyperspectrales réside dans l'ordre de grandeur du nombre de bandes qu'elles comportent (respectivement : une dizaine et une centaine), mais la frontière est assez floue.

L'analyse conjointement spatiale et spectrale de texture, que l'on a mentionné en introduction, repose sur la possibilité de disposer d'images très finement échantillonnées spectralement, ce qui est le cas des images hyperspectrales. Celles-ci seront donc le support de notre analyse.

Pour chaque échantillon spatial, elles ne fournissent plus seulement une valeur scalaire, mais un ensemble de valeurs (qui peut être vu comme un vecteur), correspondant aux

mesures du rayonnement faites dans un certain nombre de longueurs d'onde. Ces images constituent par conséquent un « pavé » hyperspectral dans lequel chaque point peut être indexé par ses deux coordonnées spatiales et sa coordonnée spectrale (ie. la longueur d'onde correspondante).

En chaque site s de la grille spatiale, c'est-à-dire en chaque pixel, la mesure est un vecteur x_s de dimension N_b égale au nombre de bandes. La valeur x_s^i pour chaque coordonnée de x_s est en théorie la mesure du rayonnement incident sur le capteur dans la longueur d'onde nominative λ_i . En pratique, il s'agit en fait de l'intégration du rayonnement sur une bande très étroite centrée sur λ_i .

La réponse spectrale x_s est, tout de même, une bonne approximation discrète du rayonnement continu réel incident sur le capteur. Les bandes sont en général régulièrement espacées du visible au proche infrarouge, on obtient alors avec x_s une approximation du spectre réfléchi par le matériau imagé dans cette plage de longueurs d'onde, qui doit être caractéristique du matériau présent dans la zone correspondant au pixel observé s .

Les données hyperspectrales sur lesquelles sont testés les méthodes et algorithmes présentés dans ce mémoire sont des images produites par le capteur aéroporté AVIRIS. Celles-ci sont d'une très grande précision spectrale, puisque le spectre du rayonnement en chaque pixel, allant de 380 à 2500 nm, est échantillonné en 224 valeurs correspondant à des plages de longueurs d'onde d'environ 10 nm de largeur. Par comparaison, en ce qui concerne les données superspectrales du capteur satellitaire MERIS, ces dernières présentent un nombre de bandes égal à 15, allant de 390 à 1040 nm. L'échantillonnage est donc plus approximatif pour les images superspectrales.

La figure 1.1 est une illustration des propriétés des images fournies par le capteur AVIRIS, où l'accent est mis sur la précision de l'échantillonnage du spectre mesuré en chaque pixel, à l'origine d'un fort pouvoir discriminant.

La figure 1.2 montre deux exemples de réponses spectrales données par une image AVIRIS, avec en abscisse le canal, c'est-à-dire le numéro de la bande spectrale mesurée, et en ordonnée le rayonnement mesuré. On voit que la mesure spectrale est d'une très bonne précision, et que la partie de spectre captée est bien discriminante. Cette figure montre également la forme habituelle des signaux traités. Leur caractéristique principale est de présenter à certains endroits fixes de profonds creux, qui correspondent à des longueurs d'ondes d'absorption de certaines molécules de l'atmosphère comme l'eau ou l'oxygène. Ces signaux ont aussi la particularité de décroître progressivement quand la longueur d'onde augmente, jusqu'à devenir pratiquement nuls. La raison en est que ce signal est la réflexion du rayonnement solaire sur la surface terrestre. Par conséquent, il garde l'enveloppe du rayonnement incident. Or le rayonnement solaire possède une forme en cloche qui atteint un maximum

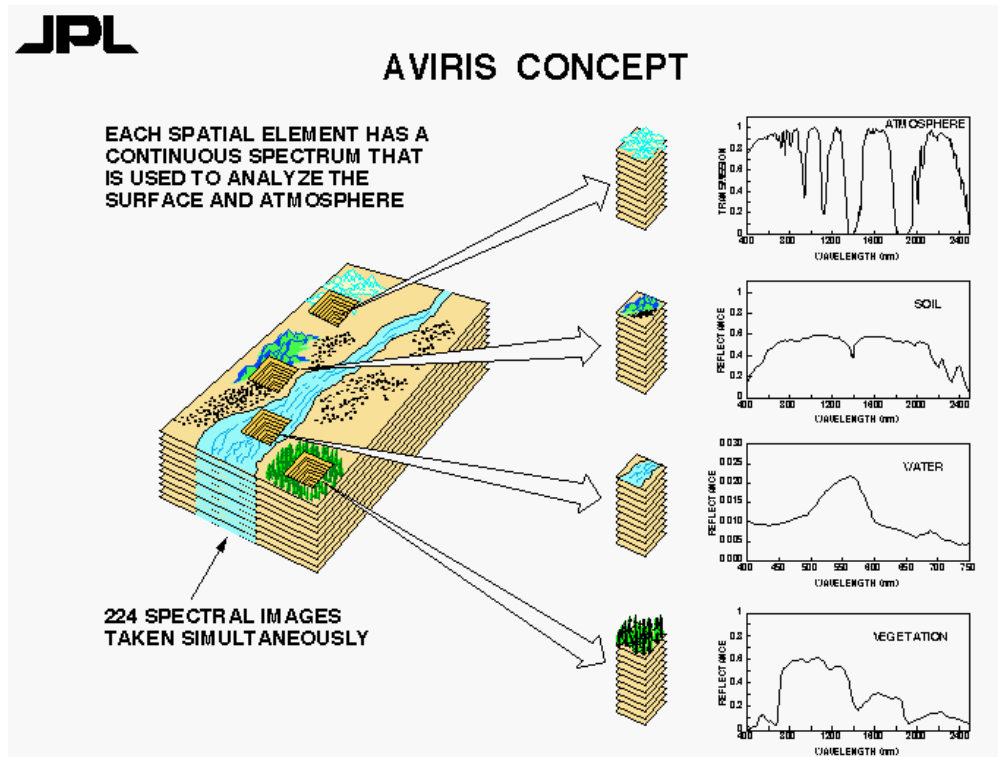


FIG. 1.1 – Concept illustré du capteur AVIRIS (© JPL).

dans la longueur d'onde du vert (environ 500 nm) et qui décroît dans l'infrarouge.

En contrepartie de cette grande précision spectrale, ces images sont peu précises spatialement. En effet, puisque les bandes spectrales sont très étroites, il faut intégrer le rayonnement sur des cellules plus grandes pour obtenir l'énergie suffisante à la mesure. Par exemple, pour le capteur aéroporté AVIRIS, on dispose de 224 mesures spectrales, pour une résolution au sol de l'ordre de 20 mètres, variable selon l'altitude du capteur.

La figure 1.3 montre un exemple d'images représentant différentes bandes d'une image AVIRIS. Il s'agit d'un extrait d'une scène de Moffett Field (Etats-Unis), contenant à la fois des zones urbaines variées et des zones rurales, ainsi qu'un plan d'eau. La résolution apparaît peu élevée, mais il apparaît également que les différentes zones ont des comportements distincts selon la longueur d'onde considérée. Par conséquent, la multiplication des bandes accompagnée d'une analyse spectrale pourrait se révéler utile.

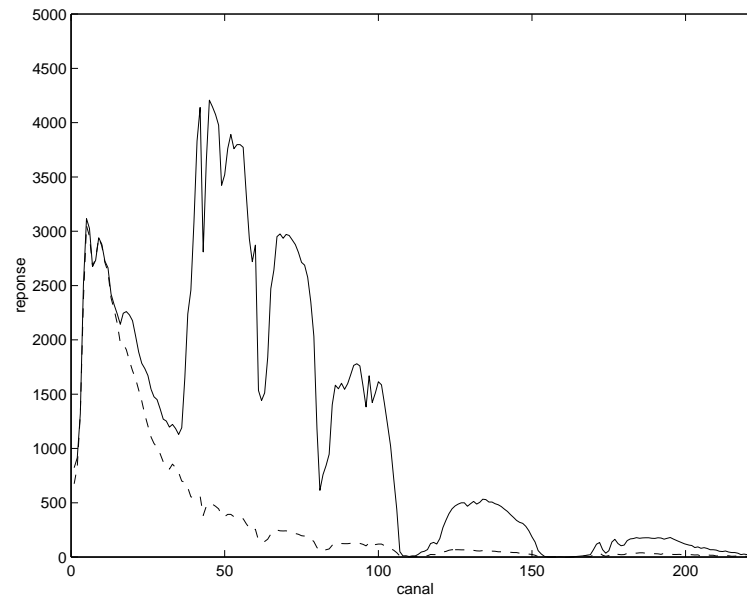


FIG. 1.2 – 2 réponses spectrales d'une image du capteur AVIRIS : intensité radiométrique en fonction de la bande de longueur d'onde mesurée. En trait plein : pixel de végétation ; en pointillés : pixel d'eau.

1.2 Analyse de la texture

Les images présentées ci-dessus sont l'objet de notre analyse car elles se prêtent naturellement à une analyse spectrale. Le deuxième aspect du traitement abordé dans ce travail est l'analyse spatiale de la texture.

L'analyse de la texture est un sujet complexe sur lequel portent de nombreux travaux. Complexe, avant tout parce que la texture en traitement d'image est un concept assez flou, et que sa définition elle-même diffère selon les points de vue. Il est cependant en général accepté de dire que la texture est ce qui, en dehors de la couleur, caractérise l'homogénéité visuelle d'une zone d'une image, comme expliqué en introduction. Il s'agit donc d'une notion impliquant les variations spatiales de l'image. On distingue deux sortes de textures, les micro-textures et les macro-textures, les méthodes de traitement des unes n'étant, en général, pas réellement efficaces pour les autres. Les micro-textures sont caractérisées par une cohérence visuelle de leurs variations de radiométrie au niveau « microscopique », c'est-à-dire au niveau du pixel. Par opposition aux précédentes, c'est au niveau d'ensembles de pixels que se situe l'homogénéité des macro-textures, par exemple par la présence de motifs qui se répètent dans l'image. Ceci est illustré par la figure 1.4, où deux exemples, une micro-

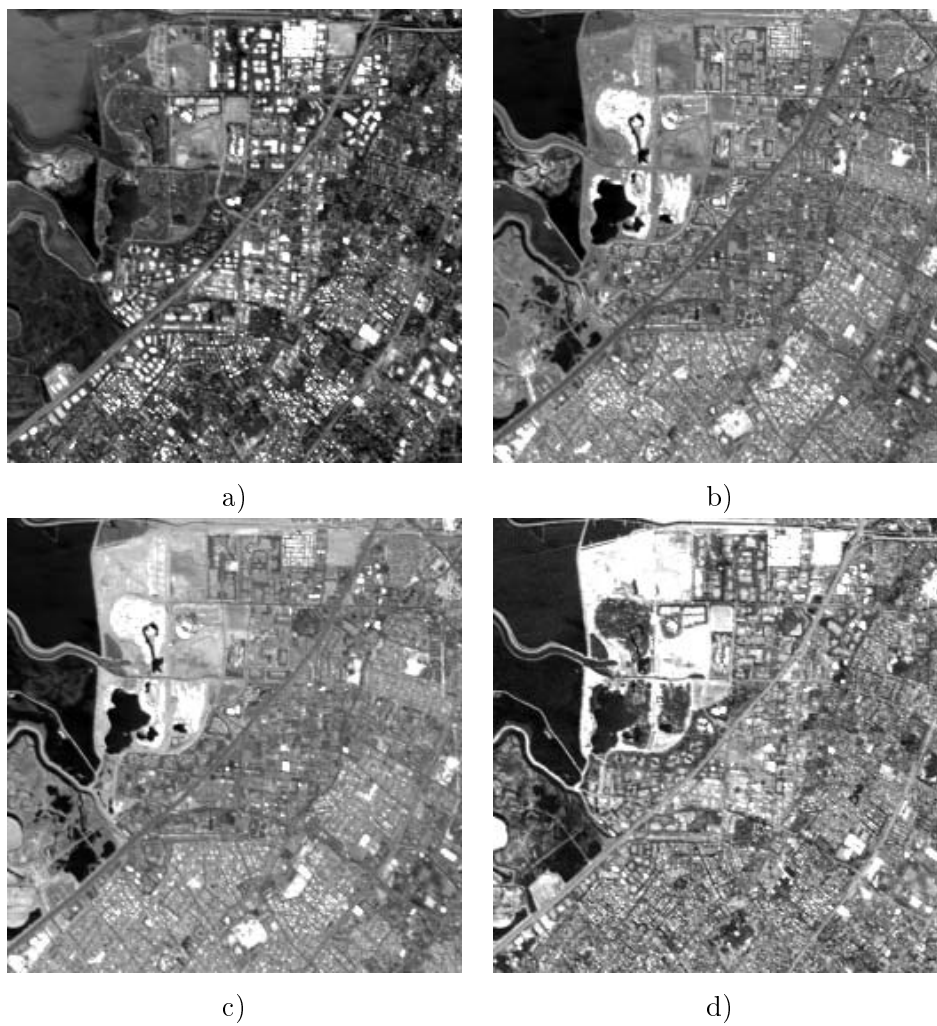


FIG. 1.3 – 4 bandes d'un extrait d'image AVIRIS: a) bande 20, b) bande 50, c) bande 100, d) bande 170.

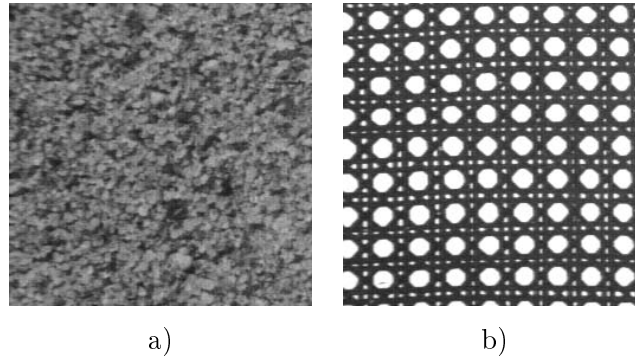


FIG. 1.4 – 2 exemples de texture : a) *micro-texture*, b) *macro-texture*.

texture et une macro-texture, sont donnés. Cette figure montre en outre le côté aléatoire ou déterministe que peut revêtir la structure spatiale des textures. En effet, dans cet exemple, la macro-texture a une organisation quasiment déterministe, alors que la micro-texture est aléatoire.

1.2.1 Analyse de la texture : pourquoi ?

L'application dans laquelle s'inscrit les travaux de cette thèse est l'analyse des zones urbaines : celle-ci se fait par analyse de la texture dans la majorité de la littérature concernant ce problème [18, 51, 62]. Les raisons pour lesquelles une simple analyse radiométrique du niveau de gris, de la couleur ou du spectre réfléchi par un pixel ne suffisent pas dans le cadre de l'analyse urbaine sont nombreuses. Parmi ces raisons, on trouve principalement les suivantes (qui se rejoignent, pour certaines d'entre elles) :

- les zones urbaines sont caractérisées par une grande variabilité de la radiométrie, bien supérieure à celle des zones rurales, ou plus encore à celle des zones maritimes,
- les différentes classes sont souvent mélangées au niveau radiométrique (i.e. les nuages s'intersectent), car composées souvent des mêmes matériaux ou de matériaux similaires dans des proportions similaires,
- les zones urbaines présentent un aspect structuré, qui peut être dû à la répétition de bâtiments ou de routes,
- les classes urbaines ne sont pas caractérisées par un unique composant spectral,
- les distances caractéristiques de certaines zones urbaines sont supérieures à la résolution du pixel.

Ces raisons font que les zones urbaines doivent être analysées à travers leur texture. Mais au delà d'une analyse de la texture uniquement spatiale, minimum nécessaire à l'étude

des zones urbaines, on cherche ici à faire une analyse conjointe spatiale et spectrale, cette dernière étant possible grâce à l'utilisation des images hyperspectrales. Or, l'analyse de texture est un thème de recherche qui fait l'objet de nombreux travaux dans le cadre spatial pour des images monospectrales, voire multispectrales, mais peu de travaux cherchant à faire coopérer les analyses spatiale et verticale (spectrale), ont pu être recensés dans la littérature.

1.2.2 Analyse de la texture : comment ?

Les approches en matière d'analyse de la texture sont nombreuses et il serait difficile d'en faire une revue exhaustive. Néanmoins, on peut les partitionner en deux grandes classes de méthodes : celles qui concernent plus particulièrement l'analyse des micro-textures et celles qui sont plus adaptées aux macro-textures. On s'attache à détailler dans ce chapitre des méthodes d'analyse de micro-textures, car on considère que les zones urbaines, celles que l'on souhaite étudier, correspondent à cette classe de texture, hypothèse due notamment à la faible résolution des images hyperspectrales.

En ce qui concerne l'analyse de micro-textures, il existe un grand nombre de méthodes parmi lesquelles on va détailler les plus courantes et efficaces, famille par famille.

Les méthodes que l'on présente ont initialement été mises au point sur des images scalaires, c'est-à-dire de niveau de gris. On cherchera donc également à montrer pour chacune de ces familles de méthodes la possibilité d'intégrer l'aspect radiométrique dans l'analyse de texture.

Matrices de cooccurrence

Parmi les plus populaires, et sans doute les plus efficaces en analyse de textures monodimensionnelles, on trouve les approches utilisant les matrices de cooccurrence permettant d'extraire des paramètres caractéristiques de la texture observée [38, 39]. Les matrices de cooccurrence sont des matrices M_d dans lesquelles chaque coefficient $m_{i,j}^d$ est égal à la probabilité (mesurée empiriquement) d'avoir en même temps la valeur i en un pixel, et la valeur j en un pixel décalé par rapport au premier d'un vecteur d (arbitrairement choisi). Il s'agit donc de la probabilité de cooccurrence jointe de i et j en deux points séparés de d . On peut extraire, de ces matrices, une série de paramètres caractérisant la texture examinée. L'analyse de la texture se fait alors à travers l'analyse de ces paramètres, calculés sur des fenêtres glissant sur toute l'image. Ces approches donnent de bons résultats pour les images monodimensionnelles, mais leur adaptation aux images multidimensionnelles paraît difficilement réalisable. En effet, une matrice de cooccurrence est une matrice de dimension $N \times N$, où N est le nombre de niveaux de gris de l'image après quantification (on passe

en général à un nombre inférieur à quelques dizaines). Pour adapter une telle méthode aux images hyperspectrales, il faut soit travailler bande par bande de façon scalaire, auquel cas l'information inter-bande est perdue, soit travailler sur le cube image. Mais ce dernier cas est pratiquement impossible à traiter, car il impliquerait de travailler sur des matrices gigantesques $N_q N_b \times N_q N_b$, où N_q est le nombre de niveaux de gris quantifiés, et N_b le nombre de bandes. Or les images AVIRIS ont 224 bandes et sont codées sur 16 bits ! Même en utilisant une quantification avec très peu de niveaux de gris et en projetant les données pour réduire le nombre de bandes, le nombre d'échantillons nécessaires pour estimer les paramètres de ces matrices serait bien trop élevé (cf. problèmes dus à la dimensionnalité, paragraphe 1.4.1).

Méthodes de filtrage

Les approches par filtrage sont également populaires dans le domaine de l'analyse de la texture, et permettent l'analyse des macro-textures aussi bien que des micro-textures. L'idée de ces méthodes, dans lesquelles on utilise des bancs de filtres multi-canaux, est inspirée par l'analyse du système de perception visuelle humain [10, 44], qui décompose une image rétinienne en plusieurs réponses filtrées à différentes échelles et orientations. Un exemple de traitement par banc de filtres est le filtrage de Gabor [9, 17, 47, 49]. Un filtre de Gabor bidimensionnel consiste en une onde sinusoïdale plane (d'une certaine orientation et fréquence) modulée par une enveloppe gaussienne. Le filtrage correspond à une convolution par des filtres de réponse impulsionnelle de la forme suivante :

$$h(x,y) = \exp \left\{ -\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right\} \cos (2\pi(u_x x + u_y y) + \phi) \quad (1.1)$$

où $u = (u_x, u_y)^t$ est le vecteur d'onde de l'onde sinusoïdale, et ϕ la phase de cette onde. Les paramètres de texture sont des paramètres énergétiques calculés sur les images filtrées par des filtres de Gabor de vecteur d'onde à différentes orientations et normes (cette dernière correspondant à la fréquence de l'onde plane sinusoïdale) [57].

Dans le cadre multidimensionnel des images hyperspectrales, l'adaptation directe des filtres de Gabor consisterait en l'application de ces filtres sur chacune des bandes prises indépendamment. Or, effectuer une analyse indépendante des différentes bandes ferait perdre la notion d'analyse de texture à la fois spatiale et spectrale que l'on souhaite mettre en œuvre.

Ondelettes

Le formalisme des ondelettes peut également fournir un outil d'analyse de textures. Une transformée en ondelettes est la décomposition d'un signal dans une base de fonctions ap-

pelées ondelettes, caractérisant ses variations locales à différentes échelles et à différents moments (ou points de l'espace dans le cas d'une image). Cette décomposition peut être vue comme un filtrage multi-échelle du signal, dans lequel on divise successivement le signal résiduel en une sous-bande de basse fréquence (signal résiduel courant à la prochaine itération) et une sous-bande de haute fréquence. Cet outil très bien formalisé en une dimension a été étendu en deux dimensions de façon simple en opérant des filtrages alternés sur les lignes et les colonnes. Une description très complète de cet outil pour les signaux à une et deux dimensions peut être trouvée dans [63, 81].

Les images filtrées obtenues, c'est-à-dire la décomposition de l'image sur la base d'ondelettes, constituent alors des paramètres de texture, puisqu'il s'agit de mesures indiquant le comportement du signal dans le voisinage des pixels, plus ou moins immédiat selon l'échelle. C'est l'approche qu'adopte Unser dans [77], où il décompose le signal sur une famille d'ondelettes redondantes, rendant la transformation invariante par translation, et en tire en chaque point un vecteur de ces paramètres pour caractériser la texture.

La transformation en ondelettes décompose récursivement les images de basse fréquence, or, l'information de texture ne se situe pas forcément dans cette partie du spectre. Un outil plus adéquat pour l'analyse de texture demeure donc la transformation en paquets d'ondelettes, où les sous-bandes de hautes fréquences sont elles-mêmes à nouveau décomposées. Cette approche est adoptée dans [58], où les auteurs décomposent l'image en paquets d'ondelettes, puis calculent l'énergie présente dans chaque sous-bande, pour former un vecteur qui est classifié par un réseau de neurones. Une autre possibilité pour prendre en compte les sous-bandes de haute fréquence est celle de la transformée en ondelette à structure d'arbre [13], où les sous-bandes de hautes et basses fréquences ne sont décomposées que si leur énergie dépasse un certain seuil.

Comme les filtres de Gabor, les ondelettes sont essentiellement scalaires, la possibilité d'opérer une analyse globale de texture passerait, par exemple, par l'introduction d'un modèle probabiliste sur les sous-bandes de la transformée.

Modèles fractals

L'irrégularité et le comportement multi-échelle des variations d'un grand nombre de textures naturelles sont le point de départ de la modélisation fractale de celles-ci. Ces modèles sont caractérisés par une grandeur appelée dimension fractale qui caractérise la variabilité globale de l'image. Il existe plusieurs définitions de cette dimension : on peut voir cela comme une mesure de la similitude du comportement à différentes échelles de la texture, ou comme une mesure de son irrégularité [21]. Plusieurs manières de calculer cette dimension fractale peuvent être utilisées, comme par exemple la méthode du spectre de puissance [68],

la méthode de la variation [26] ou la méthode par « box-counting » [55].

L'outil des fractales est très bien formalisé mais, en analyse de texture, la dimension fractale est en général la seule grandeur utilisée pour la caractérisation de celles-ci. La compacité de ce modèle en est un atout, mais si la modélisation n'est fondée que sur ce paramètre, celui-ci peut ne pas être suffisant pour bien décrire et caractériser une texture un peu complexe. Pour remédier à cela, une solution est de trouver la dimension fractale d'images dérivées de l'image initiale : obtenues par filtrage de cette dernière par un banc de filtres de Gabor [54] ou par seuillages de l'histogramme [14].

La même remarque que pour les ondelettes peut être faite concernant l'adaptation de cet outil aux données multicomposantes.

Modèles markoviens

L'analyse de texture fondée sur la modélisation par champ de Markov permet non seulement de caractériser les textures, mais est aussi un outil de simulation, de segmentation, de restauration ou de compression. Ces modèles permettent de décrire de façon probabiliste les dépendances spatiales entre sites du champ. Il existe de nombreux modèles dans le cadre de ces champs, que l'on présente au chapitre suivant. La modélisation par champs de Markov est intéressante du fait que l'on travaille sur des champs de variables aléatoires qui peuvent être des vecteurs aléatoires. Or, le fait d'avoir une méthode d'analyse fondée sur une modélisation vectorielle, donc travaillant conjointement sur l'ensemble des bandes, permet de prendre en compte les relations inter-bandes, en même temps que les relations spatiales.

1.3 Classification

Un élément important du contexte de ce travail est la finalité de ce dernier. Le but de l'analyse de la texture est la classification, c'est-à-dire la partition de l'image en ensembles de pixels représentant une même occupation du sol, dont on dira appartenir à une même *classe*. Ici, on cherche plus spécialement à partitionner des zones urbaines. Le processus de classification est souvent rapproché de celui de la segmentation, mais les différences entre les deux sont grandes : le but de la segmentation est d'opérer une partition de l'image en zones connexes homogènes sans aucune sémantique, alors que celui de la classification est de déterminer, pour chaque pixel ou groupe de pixels de l'image à traiter, la classe de référence à laquelle il appartient.

Une partition des méthodes de classification peut aussi être faite, entre les méthodes dites de classification *supervisées* et les méthodes *non supervisées*. Dans le premier cas,

l'algorithme nécessite à un (ou plusieurs) stade(s) l'intervention d'un utilisateur (le cas le plus fréquent étant l'apport d'échantillons d'apprentissage), alors que dans le second cas, aucune intervention n'est nécessaire.

La classification est un problème souvent étudié en télédétection, et il existe diverses façons d'effectuer cette opération. En ce qui concerne l'imagerie hyperspectrale, on distingue principalement les approches de classification statistiques, les méthodes non paramétriques et celles fondées sur des modèles physiques, travaillant par l'analyse du spectre.

Classification par analyse du spectre

Les images hyperspectrales, du fait de leurs caractéristiques, sont souvent utilisées pour la détection ou la reconnaissance de matériaux que l'on appelle *cibles*. En effet, les images fournissent une approximation du spectre du rayonnement réfléchi par le sol, et par conséquent, moyennant des traitements adéquats, on peut retrouver le matériau responsable du spectre mesuré (ou éventuellement plusieurs matériaux dans le cas d'un mélange). Les algorithmes de classification par analyse du spectre, développés en imagerie hyperspectrale se font généralement en deux étapes :

- la première étape [36] consiste à résoudre le problème inverse permettant de passer de la mesure au niveau du capteur, entachée de distortions, d'atténuations et de bruit, à une mesure intrinsèque au sol imagé, la réflectance. Pour cela, de nombreux codes de correction ont été mis au point (modtran, lowtran, 6s).
- la deuxième consiste en la détection ou la quantification de spectres de rayonnement de référence dans les pixels d'une image hyperspectrale [12, 11, 40, 75].

Parfois, seule la deuxième étape est effectuée, et la détection se fait sur le spectre non corrigé.

Classification statistique

Quant aux méthodes statistiques, elles consistent à déterminer des frontières de décision partitionnant l'espace des pixels en différentes zones étiquetées grâce aux échantillons d'apprentissage. La détermination des frontières de décision est faite à travers une modélisation probabiliste des pixels, et les notions de distances associées à ces modélisations. Ensuite, la classification peut se faire simplement pour chaque pixel en considérant la zone à laquelle il appartient, ou par des méthodes plus complexes impliquant le contexte de chaque pixel, à travers des champs de Markov, par exemple, ce qui permet une prise en compte du contexte spatial pour aboutir à des classifications plus régulières.

Une méthode très courante (sans doute la plus courante) est la classification par maximum de vraisemblance, avec une modélisation gaussienne. On calcule dans un premier

temps la loi de probabilité conditionnelle (supposée gaussienne) pour chacune des classes de référence. Ensuite, on associe à chacune de ces classes une zone où la probabilité conditionnellement à cette classe est la plus grande (par rapport aux autres classes). Lors de la classification, tous les pixels d'une zone sont assignés à la classe correspondante.

Classification non paramétrique

Les méthodes non paramétriques consistent également à déterminer des frontières de décisions grâce aux échantillons d'apprentissage, mais ne se basent pas sur une quelconque modélisation de la formation des pixels. La méthode des k plus proches voisins ($k - NN$ [19, 76]) consiste à assigner au pixel étudié l'étiquette de la classe la plus présente parmi les k échantillons d'apprentissage les plus proches. Deux variantes, proposées dans [74], permettent de régler les conflits quand plusieurs classes sont représentées par le même nombre de voisins. Elles se fondent sur les distances entre l'échantillon étudié et les classes en conflit.

La classification par *Support Vector Machines* est une autre méthode non paramétrique récente [80]. Elle consiste à résoudre un problème de classification binaire en plaçant un hyperplan dans l'espace des données comme frontière de décision de manière à ce que :

- cet hyperplan maximise le taux de bonne classification des échantillons d'apprentissage,
- la distance entre le plan et le pixel le plus proche est maximisée.

Ce plan est alors défini par une combinaison des échantillons les plus proches de ce plan, qui sont appelés les *vecteurs supports*. Cette approche est elle aussi intéressante, puisque l'optimisation est supposée maximiser directement la classification, mais ses développements concernent principalement le problème à deux classes, et la généralisation n'est pas directe. Néanmoins, les résultats sont satisfaisants, comme le montrent certaines applications sur des images de télédétection non-texturées [43, 84].

Une autre classe de méthodes de classification non paramétriques s'appuie sur les réseaux de neurones. Il existe une très vaste littérature décrivant essentiellement deux types de réseaux : les réseaux multi-couches, et les cartes à auto-organisation (en anglais, *self organizing maps*), comme les cartes de Kohonen. Les cartes de Kohonen sont des systèmes non linéaires et non supervisés dont l'apprentissage se fait automatiquement à l'aide d'échantillons non étiquetés. Le résultat de l'apprentissage est une carte dont chaque cellule comprend une sortie possible du système. Chaque sortie est une combinaison d'un certain nombre des entrées utilisées pour l'apprentissage. Il s'agit donc, pour résumer grossièrement,

d'une technique d'agrégation (en anglais *clustering*). Les réseaux multi-couches sont, quant à eux, des systèmes qui permettent la classification supervisée. En entrée de ces réseaux, on place les vecteurs à classifier, et les sorties sont en général associées aux différentes classes. Les sorties sont la combinaison des réponses des différentes couches cachées de neurones, associées de façon non-linéaire (par le biais d'une fonction sigmoïde, par exemple). Dans la plupart des exemples, on obtient les paramètres par un algorithme de propagation de gradient. La combinaison des deux types d'algorithmes, comme dans [61], peut procurer des classifications assez efficaces.

1.4 Nécessité et justification de la réduction de l'espace

Une partie de ce mémoire sera consacrée au problème de la réduction d'espace. Il s'agit de méthodes permettant de passer des mesures faites dans l'espace initial de dimension égale au nombre de bandes à un espace de dimension réduite. Cette réduction est faite, comme on le verra alors, par une projection des données dans un sous-espace. La détermination d'un bon sous-espace final ayant les propriétés désirées est le sujet du chapitre 3.

Dans ce paragraphe, on explique pourquoi l'on a besoin d'effectuer cette réduction d'espace, et pourquoi, dans le cadre de notre application de classification de textures, la perte d'information liée à cette projection des données peut être assez faible.

1.4.1 Nécessité

Efficacité algorithmique

Un premier motif de la réduction de l'espace d'observation est d'ordre pratique : le volume de mémoire nécessaire au stockage et à la manipulation d'une image hyperspectrale est très grand, parfois jusqu'à 200 fois plus important qu'une image monospectrale de même dimension. Par conséquent, la manipulation de ces images est très lourde. Les traitements sont aussi plus longs, du fait des calculs plus nombreux : par exemple, estimer la variance d'une image monospectrale consiste en l'estimation d'une unique variable, alors qu'estimer la matrice de covariance d'une image à N bandes consiste en l'estimation de $\frac{N(N-1)}{2}$ variables. D'autre part, si les données sont de trop grande dimension, il faut procéder à de fréquentes et coûteuses lectures et écritures sur disque. Or, l'acceptabilité d'un traitement dépend non seulement de la qualité du résultat obtenu, mais aussi de sa vitesse de calcul.

Problèmes d'estimation

Lors de tout traitement statistique paramétrique, on doit passer par une ou plusieurs étapes d'estimation de paramètres. Dans le cas de la classification, par exemple, il est nécessaire d'estimer les paramètres des lois de probabilité des différentes classes.

Or, étant donné un nombre d'échantillons d'apprentissage fixé, la qualité de l'estimation de paramètres statistiques diminue avec le nombre de bandes car en même temps, le nombre de paramètres à estimer (i.e. d'inconnues) croît. Dans le cas de l'estimation de matrices de covariance, on peut même en arriver à obtenir des matrices singulières. En classification, ceci se traduit par une diminution du taux de bonnes décisions quand on augmente la dimension du vecteur observé tout en gardant constant le nombre de pixels d'apprentissage : c'est l'effet de Hughes [46].

1.4.2 Justification

La réduction de la dimension de l'espace d'observation, à travers l'extraction d'un sous-espace de l'espace initial peut donc présenter de gros avantages, si elle n'est pas trop pénalisante en terme de qualité du traitement envisagé.

Or, plusieurs caractéristiques des espaces de grande dimension en général, et des images hyperspectrales en particulier, font que cette réduction est possible au prix d'une perte d'information minimale.

Renforcement de l'hypothèse de normalité

Une caractéristique des espaces de grande dimension est que les projections linéaires dans des sous-espaces ont tendance à rendre les distributions gaussiennes. Plus précisément, un jeu de données de distribution quelconque dans un espace de dimension initiale N voit sa projection dans un sous-espace de dimension M fixée tendre vers un modèle gaussien quand N tend vers l'infini [37].

Ceci permet d'utiliser des algorithmes de classification paramétriques se fondant sur une hypothèse de normalité. En particulier, si l'on réalise une classification par maximum de vraisemblance, les équations à résoudre sont très simples.

Corrélation

La figure 1.5 représente la matrice de covariance d'une scène AVIRIS. La matrice montre de fortes corrélations entre bandes adjacentes, par blocs. Les matrices de covariance de chaque classe présentent également la même structure très corrélée par blocs. Ceci dénote



FIG. 1.5 – Image de la matrice de covariance d'une scène AVIRIS : l'intensité en un point (i, j) de l'image est la covariance entre les mesures dans les bandes de longueur d'onde i et j , $c_{ij} = E\{x_s^i x_s^j\}$.

une grande redondance entre les différents canaux, rendant inutile l'utilisation de l'observation dans toute sa dimension. La structure par bloc vient du fait que les propriétés physiques des matériaux sont communes par groupes de longueurs d'ondes proches, et changent radicalement pour certains paliers, comme par exemple les longueurs d'ondes d'absorption du rayonnement par certaines molécules de l'atmosphère (eau, oxygène...).

D'autre part, ces matrices présentent de grandes disparités eu égard à leurs valeurs propres. Ceci traduit une forte anisotropie des nuages considérés, lesquels sont donc très aplatis et orientés.

1.5 Conclusion

Dans la première partie de ce chapitre, on a présenté le contexte et la problématique du travail. Tout d'abord, on a abordé les deux aspects clés que constituent d'une part, la nature originale et caractéristique des données de télédétection que l'on souhaite utiliser, les *images hyperspectrales*, et d'autre part, le traitement, en lui-même (l'analyse de la texture) et son objectif (analyse urbaine).

Un rapide état de l'art concernant l'analyse de texture a été fait, mettant en évidence

les possibilités de chacune des méthodes à pouvoir concilier une analyse conjointement spectrale et spatiale des textures. La plus directe et intuitive est la modélisation par champ de Markov.

On a également évoqué le problème de la classification, en rappelant quelques grandes familles de méthodes.

La deuxième partie du chapitre a été consacrée à un problème spécifique au traitement des images hyperspectrales : la réduction d'espace. On a rappelé en quoi cette réduction d'espace pouvait être à la fois nécessaire et relativement peu gênante. Les justifications de cette réduction ont été données, quant aux méthodes qui permettent cette réduction d'espace, on y reviendra en détails aux chapitres 3 et 4 de ce manuscrit.

Chapitre 2

Analyse de textures multivariées par des modèles markoviens

Ce chapitre est consacré à la modélisation des textures fines par des champs de Markov gaussiens. Ces champs sont un outil probabiliste de modélisation spatiale des textures fréquemment utilisé en traitement d'image couleur ou de niveaux de gris.

Ici, on s'intéresse plus précisément à l'adaptation du champ markovien gaussien, déjà utilisé en analyse urbaine, aux images hyperspectrales. On propose une adaptation vectorielle de ce champ, ainsi que des méthodes d'estimation de ses paramètres. Des tests de simulation sont menés en fin de chapitre afin de vérifier la pertinence de ce champ et de ses méthodes d'estimation dans le contexte qui nous occupe. On présente également une autre adaptation consistant en un champ markovien 3D de variables scalaires.

Finalement, on propose une méthode de partage du spectre en zones homogènes considérées comme indépendantes, de façon à vérifier les conditions de validité des hypothèses simplificatrices posées lors de l'adaptation du champ.

2.1 Modèles markoviens

Les champs de Markov sont un outil très utilisé en traitement d'image. Leur intérêt est principalement de pouvoir modéliser des interactions entre pixels de façon simple, à travers un modèle probabiliste. Considérons un champ aléatoire X constitué de variables aléatoires X_s en chaque site $s \in S$ (ici, s est le pixel, et S la grille de pixels). Ces variables aléatoires prennent leurs valeurs dans un espace Ω continu ou discret. On désigne par X^s le champ X privé de X_s . On note V_s le voisinage de s , un système de voisinage V se définissant comme suit :

$$V_s = \{t\} \text{ tels que } \begin{cases} s \notin V_s \\ t \in V_s \Rightarrow s \in V_t \end{cases} \quad (2.1)$$

Les lettres majuscules désignent les variables et champs aléatoires, les lettres minuscules les valeurs des réalisations respectives. Un champ de Markov X est un champ aléatoire présentant la propriété suivante :

$$P(X_s = x_s | X^s = x^s) = P(X_s = x_s | \{x_t\}, t \in V_s) \quad (2.2)$$

Ceci signifie que la probabilité en un site s (un pixel) conditionnellement au reste du champ est égale à la probabilité en ce site connaissant uniquement ses voisins. Cette modélisation permet d'introduire explicitement des interactions à courte portée, et induit implicitement des interactions à plus longue portée par "transitivité". Si l'on ajoute la contrainte de positivité $P(X = x) > 0, \forall x$, le théorème de Hammersley-Clifford établit qu'un champ de Markov est aussi un champ de Gibbs associé au même système de voisinage, c'est-à-dire que sa loi peut s'écrire de la façon suivante :

$$P(X = x) = \frac{1}{Z} \exp(-U(x)) = \frac{1}{Z} \exp\left(-\sum_{c \in C} V_c(x)\right) \quad (2.3)$$

où Z est une constante de normalisation, et C l'ensemble des cliques associées au voisinage V , c'est-à-dire regroupant tous les singletons et ensembles de sites tous voisins.

Grâce à cette modélisation contextuelle, les champs de Markov sont efficacement utilisés en segmentation [56], en classification [24] ou en restauration d'image, pour régulariser les étiquettes ou les niveaux de gris.

Mais, outre la régularisation, les champs de Markov peuvent aussi servir pour la modélisation des images, en particulier des textures fines [23, 82]. Dans ce cadre, il existe de nombreux modèles markoviens de texture, comme par exemple le modèle auto-binomial [6, 20], le modèle de Derin-Elliott [22] ou le modèle markovien gaussien aussi appelé auto-normal [15], que l'on détaille par la suite. Bien qu'il existe des modèles markoviens de texture plus sophistiqués [34], le modèle markovien gaussien est bien adapté au problème d'analyse de texture [15], notamment urbaine [62], et est assez simple.

2.2 Modèle markovien gaussien

Le modèle markovien gaussien a déjà été utilisé en imagerie monodimensionnelle pour l'analyse de la texture, notamment urbaine [25, 62]. Cette modélisation peut s'étendre dans le cadre de l'imagerie multispectrale. Pour cela, deux approches peuvent être envisagées.

La première approche, qu'a notamment utilisé Hazel dans [41], consiste à définir un modèle à deux dimensions sur les données vectorielles que constituent les observations aux différentes longueurs d'onde d'un même site. Dans cette approche, les interactions entre les valeurs à différentes longueurs d'onde ne sont pas modélisées explicitement, mais découlent des paramètres, ceux-ci étant estimés à partir des données. Il n'est donc pas nécessaire de faire d'hypothèse concernant la portée de ces interactions. Les méthodes présentées dans le chapitre 2.3 sont reliées à une modélisation découlant de la première approche, que l'on appelle MGMRF (en anglais *Multivariate Gaussian Markov Random Field*).

Dans la deuxième approche [67, 73], les données couleur ou hyperspectrales sont considérées comme une collection de variables scalaires dans un modèle à trois dimensions : deux dimensions spatiales et une dimension spectrale. Les interactions spectrales sont donc explicitement modélisées, mais ont une portée limitée par le voisinage choisi. On présente ce modèle, ainsi que des méthodes d'estimation de ses paramètres, au chapitre 2.4.

2.3 Le champ de Markov gaussien multivarié

Dans ce paragraphe, on présente un modèle markovien gaussien similaire à celui présenté dans [41], adapté aux images hyperspectrales, appelé champ de Markov gaussien multivarié, ou MGMRF. Cela permet, moyennant des hypothèses simplificatrices, réalistes si l'on prend

les précautions que l'on évoquera, d'éviter le phénomène de Hughes (cf. paragraphe 1.4.1) et d'obtenir des estimateurs simples et relativement robustes.

2.3.1 Loi conditionnelle

Soit une image x de dimensions spatiales $N_h \times N_v$ ayant un nombre de bandes N_b . Cette image est supposée être une réalisation d'un champ X . Comme on l'a vu au paragraphe 2.2, on considère un champ à 2 dimensions (les dimensions spatiales) composé de variables vectorielles de dimension égale au nombre de bandes (les réponses spectrales). La modélisation markovienne implique que la probabilité qu'un vecteur aléatoire X_s en un site s prenne la valeur x_s connaissant le reste de l'image est égale à la probabilité en ce site connaissant ses voisins. La modélisation markovienne gaussienne ajoute une contrainte sur cette probabilité, supposée être de la forme suivante :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_{t \in V_s} \theta_{t-s} x_t\|_{\Sigma}^2 \right\} \quad (2.4)$$

La notation $\|v\|_{\Sigma}^2$ signifie $v^t \Sigma^{-1} v$, la norme du vecteur v par rapport à Σ . Les θ_i sont des matrices de transformation, ou *paramètres d'interaction matriciels* et Σ désigne la *matrice de covariance conditionnelle*. Ce sont les paramètres du modèle. Cette loi est donc une loi gaussienne dont la covariance est constante et la moyenne dépendante du voisinage.

Cette expression est valide si l'on considère des données centrées, c'est-à-dire de moyenne nulle. Dans le cas de moyennes non nulles, l'expression est légèrement modifiée :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_{t \in V_s} \theta_{t-s} x_t - (I_{N_b} - \sum_{t \in V_s} \theta_{t-s}) \mu\|_{\Sigma}^2 \right\} \quad (2.5)$$

où I_{N_b} désigne la matrice identité de dimension N_b , et μ le vecteur moyen du champ.

Par la suite, on considérera que la texture homogène considérée a été centrée, la généralisation étant triviale.

La formulation (2.4) permet d'écrire pour chaque pixel s du champ :

$$x_s = \sum_{t \in V_s} \theta_{t-s} x_t + u_s \quad (2.6)$$

où l'erreur u_s est un bruit gaussien, centré, de matrice de covariance Σ mentionnée plus haut. De plus, ces erreurs u_s sont corrélées dans des mesures associées aux différents paramètres θ_t , à travers la matrice structurante A , mentionnée plus loin.

De plus, pour des contraintes de cohérence et de stationnarité du champ, on doit avoir $\theta_i = \theta_{-i}$, ce qui conduit à une formulation de la probabilité conditionnée par un nombre moindre de variables, à savoir les moyennes de voisins :

$$x_s = \sum_{k=0}^K \theta'_k m_s^k + u_s \quad (2.7)$$

où k désigne chaque type de voisinage (horizontal, vertical, ...), K le nombre de types de voisins, m_s^k la moyenne des voisins de type k , et $\theta'_k = \theta_k n_k$, avec n_k le nombre de voisins de type k . Un *type* de voisins désigne ici l'ensemble des voisins t_i d'un pixel s ayant la même contribution dans le modèle, ce qui se traduit par le même paramètre θ_{s-t_i} . À titre d'exemple, dans un système de voisinage en 4-connexité isotrope, les 4 voisins sont considérés comme étant du même type, tandis que dans un système de voisinage en 8-connexité anisotrope, on peut considérer pour chaque pixel quatre types de voisins : les voisins horizontaux, verticaux, et les deux types de voisins correspondant aux deux diagonales.

2.3.2 Loi globale

On s'intéresse maintenant à la loi globale d'un champ markovien gaussien. Pour obtenir la formule de la loi d'un champ de ce type, on revient sur la formule (2.6) :

$$u_s = x_s - \sum_{t \in V_s} \theta_{t-s} x_t \quad (2.8)$$

Soit x l'image entière réorganisée en un vecteur colonne, $x = (x_{(0,0)}^t \dots x_{(N_v, N_h)}^t)^t$, de même $u = (u_{(0,0)}^t \dots u_{(N_v, N_h)}^t)^t$, où l'exposant t signifie l'opération de transposition. Il vient alors :

$$u = Ax \quad (2.9)$$

où la structure de A est déterminée par les différents paramètres $\{\theta_t\}$ et le système de voisinage considéré (cf. annexe A). On considère des conditions périodiques aux bords, ce qui permet l'utilisation de la transformée de Fourier discrète pour les calculs concernant la matrice A , mais il peut-être intéressant de considérer différents types de conditions aux bords, comme c'est le cas dans [3, 65], où les auteurs proposent les modèles et estimateurs pour des conditions aux bords symétriques, asymétriques ou de Dirichlet.

Puisque le vecteur u suit une loi normale et que x est une transformation linéaire de u , le vecteur x suit également une loi normale. On peut donc écrire :

$$P(x) = \frac{1}{Z(S)} \exp \left(-\frac{1}{2} x^t S^{-1} x \right) \quad (2.10)$$

avec, par définition, La matrice symétrique S telle que $S = E\{xx^t\} = A^{-1}E\{ux^t\}$ et $Z(S)$ la constante de normalisation, appelée fonction de partition.

On peut prouver, d'après le principe d'orthogonalité, que l'on peut écrire [41] :

$$E\{ux^t\} = I_{N_h} \otimes I_{N_v} \otimes \Sigma \quad (2.11)$$

où \otimes désigne le produit de Kronecker¹ [48], et I_n la matrice identité de dimension n . Il s'agit donc d'une matrice où l'on trouve la matrice Σ répétée sur la diagonale.

Donc, l'équation (2.10) peut être réécrite:

$$P(x) = \frac{1}{Z(S)} \exp \left(-\frac{1}{2} x^t (I_{N_h} \otimes I_{N_v} \otimes \Sigma^{-1}) Ax \right) \quad (2.12)$$

2.3.3 Caractéristiques des espaces de grande dimension

Après avoir présenté ce modèle dans sa forme générale, on s'intéresse plus particulièrement à son application dans le cadre de l'analyse d'images hyperspectrales, notamment des images AVIRIS à 224 bandes spectrales.

Le nombre de paramètres du modèle markovien gaussien dépend de l'ordre du voisinage, de son anisotropie, et des contraintes appliquées aux matrices θ_t . Dans le cas d'un modèle anisotrope à 4 voisins par exemple, il est nécessaire de calculer les deux matrices θ_v et θ_h correspondant respectivement au voisinage vertical et horizontal, ainsi que la matrice de covariance conditionnelle Σ (soit $3N^2$ paramètres, si le vecteur d'observation est de dimension N). Or, l'analyse statistique de texture à travers un modèle probabiliste passe forcément par une (ou plusieurs) phase(s) d'estimation des paramètres. Dans le cas de données AVIRIS à 224 bandes, l'estimation robuste d'un nombre aussi élevé de paramètres est très difficile, étant donné le nombre d'échantillons disponibles pour effectuer le calcul, qui ne peut excéder quelques milliers, dans le cas de l'apprentissage des paramètres d'une classe dans un contexte de classification supervisée, par exemple. Or plus le nombre de paramètres est élevé, plus le phénomène de Hughes [46] est présent.

Deux solutions peuvent être apportées à ce problème. D'une part, comme évoqué en introduction, il est possible de choisir des critères et des méthodes permettant de réduire la dimension des observations de façon à améliorer estimation et classification. D'autre part,

1. le produit de Kronecker entre deux matrices $A = \{a_{ij}\}$ de dimension (n,n) et $B = \{b_{ij}\}$ de dimension (m,m) est une matrice $A \otimes B = C = \{c_{ij}\}$ de dimension (mn,mn) telle que $c_{im+k,jm+l} = a_{i,j}b_{k,l}$

il est possible de simplifier le modèle de façon à ce que le nombre de paramètres à estimer diminue radicalement.

Dans le cas de l'application qui nous intéresse, on met en œuvre les deux solutions. La simplification consistera à donner une certaine forme *a priori* aux matrices de transformation θ_i :

$$\theta_i = \text{diag}(a_i) = a_i I_{N_b} \quad (2.13)$$

Ceci consiste à considérer que l'on a d'une part, indépendance entre les bandes dans les interactions entre pixels, et d'autre part, que ces interactions sont identiques dans les différentes longueurs d'onde du spectre. Comme on le verra plus tard au paragraphe 2.3.6, cela se traduit par une même vitesse de décroissance des corrélations dans les différentes bandes. C'est grâce à cette propriété de la vitesse de décroissance que l'on sera capable de déterminer dans quelle mesure l'hypothèse faite ici est valable en ce qui concerne les données réelles.

Cette hypothèse a également pour conséquence de simplifier la forme de la probabilité conditionnelle :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_{t \in V_s} a_{t-s} x_t\|_{\Sigma}^2 \right\} \quad (2.14)$$

où les a_t sont des scalaires.

On peut alors écrire en termes de potentiels par paires de pixels voisins :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z'} \exp \left\{ -\sum_{t \in V_s} a_{t-s} \|x_s - x_t\|_{\Sigma}^2 - \beta \|x_s\|_{\Sigma}^2 \right\} \quad (2.15)$$

où $\beta = 1 - \sum_t a_{t-s}$.

Enfin, notons une autre manière d'écrire cette probabilité conditionnelle que l'on utilisera dans certains algorithmes d'estimation des paramètres faisant appel aux moyennes de pixels du même type dans le voisinage :

$$P(x_s | \{m_k\}) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_k a'_k m_s^k\|_{\Sigma}^2 \right\} \quad (2.16)$$

où les a'_k sont obtenus à partir des a_k comme les θ' à partir des θ , c'est-à-dire en multipliant les a_k par le nombre de pixels entrant en jeu dans le calcul de la moyenne m_s^k .

2.3.4 Domaine de définition des paramètres

Pour que le champ X de loi (2.10) existe, il faut que la matrice S , définie à l'équation 2.10, soit définie positive. Par conséquent, supposant que Σ est définie positive, il faut que la matrice A le soit également. Or, l'hypothèse concernant Σ est vraisemblable, si l'on considère un nombre d'échantillons suffisamment élevé. En effet, Σ est calculée comme $\Sigma = \sum_s u_s u_s^t$, c'est-à-dire comme une simple matrice de covariance. Celle-ci ne peut donc qu'être singulière dans des cas particuliers très peu probables, ou quand le nombre d'échantillons servant à l'estimation est trop faible.

La matrice A , quant à elle, correspond à une convolution (circulante par blocs, symétrique), dont on peut connaître les valeurs propres, donc le déterminant, en passant par sa transformée de Fourier. Cette matrice est de dimensions $N_v N_h N_b \times N_v N_h N_b$, et a donc $N_v N_h N_b$ valeurs propres. Sa structure, et le fait que les θ_t soient des scalaires multipliés par la matrice identité, font que l'on a $N_v N_h$ valeurs propres distinctes, et que chaque valeur propre l'est N_b fois (cf. annexe A). Chaque valeur propre correspond à un point du plan fréquentiel, celles-ci peuvent donc être paramétrées par des coordonnées $s = (s_1, s_2)$, évoluant dans la grille S .

Le résultat du calcul des valeurs propres λ_s (détaillé en annexe A) est le suivant :

$$\lambda_s = 1 - \sum_{m=1, n=1}^{\frac{N_h}{2}, \frac{N_v}{2}} 2a_{m,n} \cos \left(2\pi \left(\frac{s_1 m}{N_h} + \frac{s_2 n}{N_v} \right) \right) \quad (2.17)$$

où les $a_{m,n}$ sont les paramètres d'interaction correspondant à un décalage d'un vecteur (m, n) .

La matrice A est définie positive si et seulement si toutes ses valeurs propres sont positives.

Une condition suffisante pour que toutes les valeurs propres soient positives, et que le champ existe, est que :

$$\sum_{m=1, n=1}^{\frac{N_h}{2}, \frac{N_v}{2}} |a_{m,n}| < \frac{1}{2} \quad (2.18)$$

Ce domaine est un sous-ensemble du domaine de définition des paramètres. Ce dernier étant difficile à déterminer, on le caractérise plus en détails pour le cas particulier simple du modèle en 4-connexité anisotrope.

2.3.5 Domaine de définition pour le modèle en 4-connexité anisotrope

Pour le modèle en 4-connexité anisotrope, les valeurs propres λ_s s'écrivent de la façon suivante :

$$\lambda_s = 1 - 2a_{1,0} \cos\left(2\pi \frac{s_1}{N_h}\right) - 2a_{0,1} \cos\left(2\pi \frac{s_2}{N_v}\right) \quad (2.19)$$

Pour déterminer le domaine de définition, il convient donc de distinguer 3 cas :

- $a_{1,0} \geq 0, a_{0,1} \geq 0$

Dans ce cas, la valeur propre minimale est obtenue en $\hat{s}_1 = \hat{s}_2 = 0$, et la condition nécessaire et suffisante d'appartenance au domaine est que :

$$a_{1,0} + a_{0,1} < \frac{1}{2} \quad (2.20)$$

- $a_{1,0} < 0, a_{0,1} < 0$

Dans ce cas, la valeur propre minimale est obtenue en :

$$\begin{cases} \hat{s}_1 = \arg \min \cos\left(2\pi \frac{s_1}{N_h}\right) \\ \hat{s}_2 = \arg \min \cos\left(2\pi \frac{s_2}{N_v}\right) \end{cases} \quad (2.21)$$

C'est-à-dire, si N_h (resp. N_v) est pair, $\hat{s}_1 = \frac{N_h}{2}$ (resp. $\hat{s}_2 = \frac{N_v}{2}$), et si N_h (resp. N_v) est impair, $\hat{s}_1 = \mathcal{E}\left(\frac{N_h}{2}\right)$ (resp. $\hat{s}_2 = \mathcal{E}\left(\frac{N_v}{2}\right)$), où $\mathcal{E}(\cdot)$ désigne la partie entière.

La condition nécessaire et suffisante d'appartenance au domaine est :

$$a_{1,0} + a_{0,1} \frac{\cos\left(2\pi \frac{\hat{s}_2}{N_v}\right)}{\cos\left(2\pi \frac{\hat{s}_1}{N_h}\right)} > -\frac{1}{2 \cos\left(2\pi \frac{\hat{s}_1}{N_h}\right)} \quad (2.22)$$

Si l'on tend vers la limite, c'est-à-dire un champ à dimensions infinies, le membre de droite tend vers 0.5, et la pente de la droite limite du domaine tend vers 1.

- $a_{1,0} \geq 0, a_{0,1} < 0$

Dans ce cas, la valeur propre minimum est obtenue en :

$$\begin{cases} \hat{s}_1 = \arg \max \cos\left(2\pi \frac{s_1}{N_h}\right) \\ \hat{s}_2 = \arg \min \cos\left(2\pi \frac{s_2}{N_v}\right) \end{cases} \quad (2.23)$$

C'est-à-dire $\hat{s}_1 = 0$, et $\hat{s}_2 = \frac{N_v}{2}$.

La condition nécessaire et suffisante d'appartenance au domaine est :

$$a_{1,0} + a_{0,1} \cos\left(2\pi \frac{\hat{s}_2}{N_v}\right) < \frac{1}{2} \quad (2.24)$$

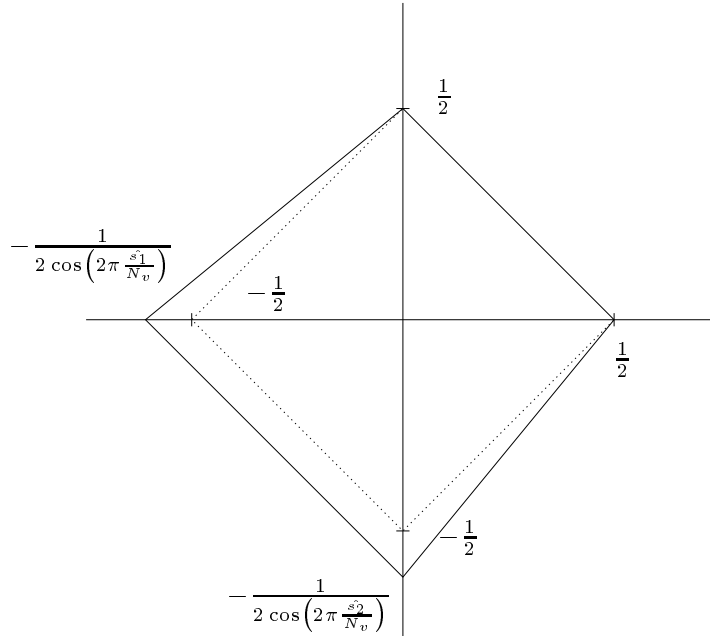


FIG. 2.1 – *Domaine de définition pour les coefficients $a_{1,0}$ et $a_{0,1}$. En pointillés, le cas de dimensions infinies ou paires. En traits plein, un cas quelconque.*

La limite du domaine est une droite de pente $-\cos\left(2\pi\frac{s_2}{N_v}\right)$, positive.

Le cas $a_{0,1} \geq 0, a_{1,0} < 0$ se retrouve aisément à partir de celui ci.

Le domaine de définition est donc l'intérieur d'un quadrilatère contenant le point $(0,0)$, délimité par les droites définies ci-dessus. Si les dimensions du champ sont paires, alors les cosinus minimaux et maximaux obtenus pour tous les \hat{s} sont égaux soit à 1 soit à -1 . Dans ce cas, la condition suffisante (2.18) est aussi nécessaire, et le domaine de définition est le carré centré en zéro en pointillés sur la figure 2.1. C'est aussi vrai dans le cas limite où les dimensions tendent vers l'infini par valeurs impaires. En revanche, si les dimensions du champ sont réduites, le domaine de définition n'est plus le même carré, mais un quadrilatère incluant ce carré, comme illustré en traits pleins sur la figure 2.1.

C'est la raison pour laquelle, pour ce cas particulier du modèle (ainsi que pour le modèle 4-connexe isotrope), il n'est pas très pénalisant, voire pas du tout, en terme de perte de candidats potentiels, de se restreindre au domaine défini par la condition (2.18).

2.3.6 Corrélations du modèle

Le calcul des matrices de corrélation du MGMRF est utile pour deux raisons. D'une part, il nous permet de justifier le choix de ce modèle pour les scènes que l'on veut analyser, et d'autre part, une caractéristique des matrices de corrélation permet de proposer un estimateur efficace des paramètres de ce modèle (cf. paragraphe 2.3.7).

La loi de probabilité globale du MGMRF décrite en (2.12) vient de la formule (2.11), que l'on peut récrire comme suit :

$$E\{xx^t\} = A^{-1} (I_{N_h} \otimes I_{N_v} \otimes \Sigma) \quad (2.25)$$

On en déduit une première information sur les matrices de corrélation entre « hyperpixels » x_s :

$$C_d = A_{(i,i+d)}^{-1} I_{N_b} \Sigma = A_{(i,i+d)}^{-1} \Sigma \quad (2.26)$$

où $C_d = E\{x_s x_{s+d}^t\}$.

En effet, comme expliqué en annexe A, la matrice A peut être écrite $A = A' \otimes I_{N_b}$, où A' est la matrice structurante du champ markovien gaussien dans le cas monodimensionnel. Ici, $A_{(i,i+d)}^{-1}$ est donc le coefficient correspondant à un décalage spatial de d , de l'inverse de cette matrice A' .

Les matrices de corrélation C_d sont donc proportionnelles à la matrice de covariance conditionnelle, avec un coefficient de proportionnalité dépendant du déplacement d . Ce coefficient peut être calculé en inversant la matrice A , par exemple en passant par la transformation de Fourier (cf. annexe A).

On peut, grâce à cette propriété, vérifier bien-fondé de l'hypothèse simplificatrice sur les paramètres d'interaction : si les matrices de corrélation calculées sur des images possèdent la propriété d'être proportionnelles entre elles, on peut considérer que la simplification n'est pas préjudiciable à la précision du modèle.

Dans le tableau 2.1, quelques textures sur lesquelles nous avons cherché à vérifier le bien-fondé de l'hypothèse faite sont recensées. Pour ce faire, on calcule la matrice de covariance et les matrices de corrélation entre un pixel et ses deux plus proches voisins horizontaux et verticaux. Ce tableau fournit la valeur moyenne ainsi que l'écart type du quotient Q_{01} (resp. Q_{10}) entre un terme de la matrice de covariance et son homologue dans la matrice de corrélation verticale (resp. horizontale).

Les cas 1 et 2 concernent des extraits de deux zones urbaines provenant de l'image de Moffett Field prise par le capteur AVIRIS. Dans les deux cas, on a gardé 25 bandes du spectre visible. On remarque que la variation des quotients Q est assez faible en pourcentage

Cas n^o	Texture	Nb Bandes	Moyenne \bar{Q}_{01}	Ecart type $\sigma_{Q_{01}}$ (% \bar{Q}_{01})	Moyenne \bar{Q}_{10}	Ecart type $\sigma_{Q_{10}}$ (% \bar{Q}_{10})
1	AVIRIS 1	25 (V)	0.83	0.20 %	0.79	0.29 %
2	AVIRIS 2	25 (V)	0.60	0.93 %	0.55	0.87 %
3	AVIRIS 1	25 (IR)	0.84	0.12 %	0.80	0.17 %
4	AVIRIS 2	25 (IR)	0.51	0.90 %	0.48	0.41 %
5	AVIRIS 1	25 (V/IR)	0.84	4.00 %	0.79	3.14 %
6	AVIRIS 2	25 (V/IR)	0.42	353.12 %	0.34	447.66 %

TAB. 2.1 – *Moyenne et écart type des quotients entre coefficients des matrices de corrélation avec les plus proches voisins et de la matrice de covariance, pour différentes textures et sélections de bandes spectrales*

(moins de 1%). Par conséquent, dans le cas de textures urbaines à faible résolution, l'hypothèse simplificatrice sur les paramètres est réaliste. Pour les cas 3 et 4, les tests concernent, pour les mêmes textures 25 bandes provenant cette fois-ci de l'infrarouge. Dans ces deux cas, la même constatation peut être faite: on a également une très faible variabilité des quotients Q , également inférieure à 1%. L'hypothèse est donc toujours réaliste. Enfin, les cas 5 et 6 concernent toujours les mêmes textures, mais cette fois-ci, les bandes gardées sont une sélection de bandes prises sur l'ensemble du spectre, du visible à l'infrarouge. Dans ces cas, on constate que l'hypothèse n'est plus vérifiée, car la variabilité du quotient Q est très grande.

Il semble donc que l'hypothèse faite n'est vérifiée que lorsque les bandes choisies sont dans une même zone spectrale (visible, proche infrarouge...).

Par la suite, on considérera que ces conditions sont respectées. Les images hyperspectrales couvrant en général diverses parties du spectre, il faudra découper le spectre en zones aux propriétés spectrales similaires, et les modéliser séparément. Le problème de la partition du spectre en zones de caractéristiques homogènes est traité au paragraphe 2.5.

2.3.7 Estimation des paramètres du champ

L'estimation de paramètres est une étape nécessaire dans le cadre des traitements par des méthodes fondées sur un modèle paramétrique. Dans notre cas, on se trouve en présence d'une réalisation d'un champ supposé markovien gaussien, et l'on cherche à retrouver le modèle dont est issu cette réalisation.

Considérons une image x de taille $N_v \times N_h$, de nombre de bandes N_b , de moyenne nulle, supposée avoir été générée par le modèle décrit par l'équation (2.14), dont les paramètres

sont inconnus. Notre objectif est estimer ces paramètres d'une façon rapide et précise.

Maximum de vraisemblance

Dans le cas de ce modèle, on s'intéresse au calcul de l'estimateur correspondant au critère du maximum de vraisemblance des paramètres. Satisfaire ce critère consiste à choisir les paramètres les plus « vraisemblables », c'est-à-dire ceux pour lesquels la probabilité de la réalisation du champ x *connaissant les paramètres* est maximale.

Le problème consiste à retenir le vecteur de paramètres $\hat{\phi}$ (dans le cas présent $\phi = (\{a_k\}, \Sigma)$), tel que :

$$\hat{\phi} = \arg \max_{\phi} (P(x|\phi)) \quad (2.27)$$

En revenant à l'équation (2.10) et en explicitant la constante de normalisation, il vient :

$$P(x|\phi) = \frac{(2\pi)^{-\frac{N_v N_h N_h}{2}}}{|S|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}x^t S^{-1}x\right) \quad (2.28)$$

où $|S|$ désigne le déterminant de S .

La log-vraisemblance $LV(\phi)$ s'écrit donc :

$$LV(\phi) = C - \frac{1}{2} \ln |S| - \frac{1}{2} x^t S^{-1} x \quad (2.29)$$

où C est une constante. Par conséquent :

$$\arg \max_{\phi} (LV(\phi)) = \arg \min_{\phi} (\ln |S| + x^t S^{-1} x) \quad (2.30)$$

D'où :

$$\hat{\phi} = \arg \min_{\phi} (\ln |S| + x^t S^{-1} x) \quad (2.31)$$

L'optimisation de cette expression est complexe, du fait du calcul du déterminant de S . Néanmoins, cette fonction de log-vraisemblance est calculable, et on peut l'optimiser, du moins par des méthodes numériques. D'après l'équation (2.12), on a :

$$S^{-1} = (I_{N_h} \otimes I_{N_v} \otimes \Sigma^{-1}) A \quad (2.32)$$

Donc :

$$|S|^{-1} = |\Sigma|^{-N_h N_v} |A| \quad (2.33)$$

Or, $|A|$ peut s'écrire (cf. annexe A pour plus de détails) :

$$|A| = \prod_{x=0, y=0}^{N_h, N_v} \sum_{m=0, n=0}^{N_h, N_v} a_{m,n} \exp \left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v} \right) \right) \quad (2.34)$$

Le déterminant $|S|$ est donc calculable en fonction de ϕ .

D'autre part, on peut réduire le jeu de paramètres. En effet, il est possible de calculer le $\hat{\Sigma}(\{a_k\})$, correspondant au maximum de vraisemblance de Σ connaissant les $\{a_k\}$.

$$\hat{\Sigma}(\{a_k\}) = \frac{1}{N} \sum_i x_i u_i^t \quad (2.35)$$

La fonction $LV(\phi)$, à $\{a_k\}$ fixé, présente un maximum en $\hat{\Sigma}(\{a_k\})$. Par conséquent :

$$\max_{\phi} LV(\phi) = \max_{\{a_k\}} LV \left(\{a_k\}, \hat{\Sigma}(\{a_k\}) \right) \quad (2.36)$$

L'espace de recherche du maximum est réduit de façon considérable, puisqu'il s'agit de l'espace de définition des $\{a_k\}$, et la recherche est plus efficace. Celle-ci est effectuée par l'algorithme du gradient conjugué

Influence sur les corrélations

On peut caractériser en quoi consiste en pratique l'obtention de l'estimateur du maximum de vraisemblance. En effet, l'estimée des paramètres d'interaction est la valeur que ces paramètres prennent quand les dérivées partielles de la fonction de vraisemblance s'annulent. En combinant (2.36) et (2.33), on obtient l'équation suivante :

$$\hat{a} = \arg \max_a (LV(a)) = \arg \min_a \left(\sum_s x_s^t \hat{\Sigma}^{-1} u_s + N_h N_v \ln \left| \hat{\Sigma} \right| - \ln |A| \right) \quad (2.37)$$

où a est le vecteur des paramètres a_k , et $\hat{\Sigma} = \hat{\Sigma}(a)$ est une fonction du vecteur a , cf. équation (2.35).

Par définition de $\hat{\Sigma}$, $\sum_s x_s^t \hat{\Sigma}^{-1} u_s = N$. Par conséquent :

$$\frac{\partial}{\partial a_k} LV = \frac{\partial}{\partial a_k} \ln |A| - N_h N_v \frac{\partial}{\partial a_k} \ln \left| \hat{\Sigma} \right| \quad (2.38)$$

Or, l'on sait que $\Sigma(\hat{a}) = C_0 - \sum_{t \in V_0} a_t C_t$, les C_t représentant les matrices de corrélation. Donc, d'après le résultat suivant sur la dérivation de déterminant, connu sous le nom de formule de Jacobi :

$$\frac{\partial |M|}{\partial x} = |M| \text{trace} \left(M^{-1} \frac{\partial M}{\partial x} \right) \quad (2.39)$$

les dérivées partielles du deuxième terme s'écrivent :

$$\frac{\partial}{\partial a_k} \ln |\hat{\Sigma}| = \text{trace}(\hat{\Sigma}^{-1} C_k) \quad (2.40)$$

Par ailleurs, l'on peut montrer (cf. annexe A), que la dérivée partielle du logarithme du déterminant de la matrice A par rapport à a_k est égale au coefficient de A^{-1} correspondant au déplacement de type k .

Le maximum de vraisemblance étant atteint quand la dérivée de la log-vraisemblance est nulle, il vient :

$$A_{m,n}^{-1} = \frac{1}{N_b} \text{trace}(\hat{\Sigma}^{-1} C_{m,n}), \forall (m,n) \in V_0 \quad (2.41)$$

L'estimateur du maximum de vraisemblance fait donc coïncider la corrélation spatiale théorique du champ avec une moyenne de la corrélation empirique, ceci pour les corrélation entre un pixel *et son voisinage*.

Maximum de pseudo-vraisemblance

Un critère permettant une estimation plus rapide est celui du maximum de pseudo-vraisemblance (MPV) [6]. Ce critère est fondé sur une hypothèse d'indépendance des probabilités conditionnelles. Il mène à l'approximation suivante concernant la probabilité du champ x :

$$P(x) \approx \prod_{s \in \Omega} P(x_s | \{x_t\}, t \in V_s) \quad (2.42)$$

L'estimateur de maximum de pseudo-vraisemblance est :

$$\phi = \arg \max_{\phi} \left(\prod_{s \in \Omega} P(x_s | \{x_t\}, \phi) \right) \quad (2.43)$$

$$\phi = \arg \min_{\phi} \left(\sum_s \|x_s - \sum_k a'_k m_s^k\|_{\Sigma}^2 + N_v N_h \ln |\Sigma| \right) \quad (2.44)$$

Une fois de plus, la minimisation ne se ramène pas à une formule simple, notamment parce que la matrice Σ dépend des $\{a'_k\}$, ce qui signifie que la minimisation doit se faire sur les $\{a'_k\}$. En considérant la matrice Σ indépendante de ces coefficients, la minimisation selon les $\{a'_k\}$ est ramenée à un calcul matriciel :

$$\begin{pmatrix} \langle (m_s^0)^t \Sigma^{-1} m_s^0 \rangle & \dots & \langle (m_s^0)^t \Sigma^{-1} m_s^K \rangle \\ \vdots & & \vdots \\ \langle (m_s^K)^t \Sigma^{-1} m_s^0 \rangle & \dots & \langle (m_s^K)^t \Sigma^{-1} m_s^K \rangle \end{pmatrix} \begin{pmatrix} a'_0 \\ \vdots \\ a'_K \end{pmatrix} = \begin{pmatrix} \langle x_s^t \Sigma^{-1} m_s^0 \rangle \\ \vdots \\ \langle x_s^t \Sigma^{-1} m_s^K \rangle \end{pmatrix} \quad (2.45)$$

On approxime alors la matrice de covariance conditionnelle par la matrice identité (ce qui ramène le problème à une simple estimation aux moindres carrés) ou par la matrice de covariance générale, de façon à pouvoir calculer les paramètres en deux étapes rapides : d'abord les $\{a'_k\}$, ensuite Σ .

Une fois les $\{a'_k\}$ calculés, le calcul de Σ est un simple calcul de covariance :

$$\Sigma = \langle u_s u_s^t \rangle = \left\langle \left(x_s - \sum_k a'_k m_s^k \right) \left(x_s - \sum_k a'_k m_s^k \right)^t \right\rangle \quad (2.46)$$

Estimateur MMSE

On peut également dériver un estimateur à partir d'un autre critère : l'erreur quadratique moyenne. L'estimateur MMSE (*Minimum Mean Square Error* en anglais) est l'estimateur minimisant ce critère.

Le MMSE obéit au principe d'orthogonalité, c'est-à-dire que l'erreur d'estimation u_s est orthogonale aux données servant à l'estimation. Cela se traduit par l'équation suivante :

$$E\{x_r^t u_s\} = 0, \quad \forall s \neq r \quad (2.47)$$

En explicitant u_s et en développant, l'équation (2.47) devient :

$$E\{x_r^t x_s\} - \sum_{t \in V_s} a_{t-s} E\{x_r^t x_t\} = 0, \quad \forall s \neq r \quad (2.48)$$

En notant $c_d = E\{x_s^t x_{s+d}\}$, l'équation peut se récrire :

$$c_d - \sum_{t \in V_0} a_t c_{d+t} = 0, \quad \forall d \neq 0 \quad (2.49)$$

Il faut autant d'équations que d'inconnues pour résoudre le système. Un choix naturel est donc de prendre le système formé des équations (2.49) pour $d \in V_0$.

Pour estimer les paramètres a_k , il suffit donc d'estimer les coefficients c_d et c_{d+t} du champ nécessaires à la résolution du système d'équations, et de le résoudre. Il est à noter que le système est très proche de celui obtenu en utilisant le critère du maximum de pseudo-vraisemblance (2.45).

Par exemple, dans le cas d'un champ isotrope en 4-connexité, il suffit de calculer $c_{0,0}$, $c_{0,1}(= c_{1,0} = c_{0,-1} = c_{-1,0})$, $c_{0,2}(= c_{2,0} = c_{0,-2} = c_{-2,0})$ et $c_{1,1}(= c_{1,-1} = c_{-1,1} = c_{-1,-1})$.

Finalement, il vient :

$$a' = 4a = \frac{16c_{0,1}}{4c_{0,0} + 8c_{1,1} + 4c_{0,2}} \quad (2.50)$$

Il suffit donc pour cet estimateur de calculer les 4 coefficients de corrélation pour estimer a , c'est à dire un calcul de complexité $\mathcal{O}(N_e N_b)$. Ensuite, il faut calculer le matrice Σ , par un calcul de complexité $\mathcal{O}(N_e N_b^2)$. A titre de comparaison, le calcul de Σ est fait à chaque évaluation de la fonction de vraisemblance. Or le calcul de la vraisemblance est en général fait quelques centaines de fois lors de l'optimisation numérique de celle-ci. Le gain en temps de calcul est donc très grand.

2.3.8 Simulations

Les simulations vont nous permettre de vérifier deux points. D'une part, des tests d'estimation sur des images simulées dont on connaît les paramètres permettront la validation de la qualité des estimateurs, et d'autre part, ces images simulées vont montrer l'adéquation du modèle aux données dont nous disposons.

Pour atteindre ce second but, la procédure adoptée est la suivante :

- Dans un premier temps, on réduit la dimension de la texture proposée par analyse en composantes principales (ACP) [27], de façon à limiter la taille de l'espace dans lequel se trouvent les pixels, pour faciliter l'échantillonnage (par l'algorithme de Metropolis mentionné ci-dessous).
- Ensuite, on analyse la texture en estimant \hat{a} et $\hat{\Sigma}$ correspondant à un modèle isotrope 4-connexe.
- Dans un troisième temps, on échantillonne une texture ayant ces paramètres par l'algorithme de Metropolis [64].
- Enfin, on replonge la texture dans l'espace initial (en effectuant la rotation inverse), de façon à avoir une évaluation visuelle du résultat.

Qualité des estimateurs

La figure 2.2 montre les résultats obtenus pour des images de synthèse. Ces images ont été simulées par échantillonnage grâce à l'algorithme de Metropolis, selon le modèle MGMRF, avec des paramètres connus. Les paramètres choisis pour les simulations ne correspondent pas à des situations réelles, mais ils permettent de tester l'efficacité de l'estima-

tion. Ici, on ne passe pas par l'ACP, car les données image ont été simulées avec 3 bandes spectrales, afin de faciliter la simulation.

Non seulement les résultats sont satisfaisants visuellement, car la simulation est très proche des données, mais on peut également les valider mathématiquement, car les paramètres sont connus. Les paramètres extraits et réels pour les données simulées de la figure 2.2 sont donnés dans le tableau 2.2. Les résultats numériques sont très satisfaisants, puisqu'ils montrent que pour ces images "idéales" (ie. générées par le modèle) l'erreur d'estimation (pour les critères du MV et du MPV) est de moins de 1% pour le paramètre d'interaction, et d'au maximum 8.6% (de façon isolée) pour les coefficients de la matrice Σ (qui sont d'ordre 2). D'autre part, on ne constate pas de différence particulière entre les deux critères pour ces images, excepté une erreur plus importante dans l'estimation du paramètre d'interaction pour la zone en haut à droite (0.843 au lieu de 0.9). L'estimateur du MPV permet des estimations beaucoup plus rapides, puisque dans le cas d'images à 3 bandes, et pour un nombre d'échantillons de 4096, celui-ci donne un résultat en 0.2 secondes, alors que l'estimateur du MV met 4.5 à 5.5 secondes, selon le nombre d'itérations, sur une station Sun Ultra 10.

En revanche, il est à noter qu'un problème majeur de l'estimateur du MPV est que, puisqu'il résulte de la résolution d'un système d'équations linéaires, aucune précaution n'est prise pour que l'estimée des paramètres d'interaction reste située dans leur domaine de validité. Si les coefficients du système sont mal estimés (par exemple à cause d'irrégularités du champ dans la fenêtre), on peut obtenir des paramètres inacceptables, en dehors du domaine de définition du champ.

Adéquation du modèle

Pour ce qui est de la deuxième série de simulations, on montre des résultats d'analyse/synthèse sur des images réelles dans la figure 2.3. Sur cet exemple, on remarque bien que le modèle n'est pas capable de saisir la structure complexe de macro-textures comme celle présente dans le quart supérieur droit de la figure 2.3, notamment concernant la périodicité et l'orientation. Néanmoins, on est capable de bien prendre en compte l'information relative à la longueur de corrélation et aux variations de couleur. Les textures plus fines des trois autres quarts sont bien rendues, excepté aux endroits où sont présentes des anomalies (quart supérieur gauche). Notons enfin que ces simulations ont été faites avec le modèle spatial le plus simple, c'est-à-dire 4-connexe isotrope. Or, il serait en théorie possible de mieux approcher les textures avec des interactions à plus longue portée (cf. [34] sur les textures de Brodatz), mais ceci pourrait causer plusieurs problèmes. Le premier est que l'augmentation de la portée des interactions, et donc des types de voisinage, implique un nombre

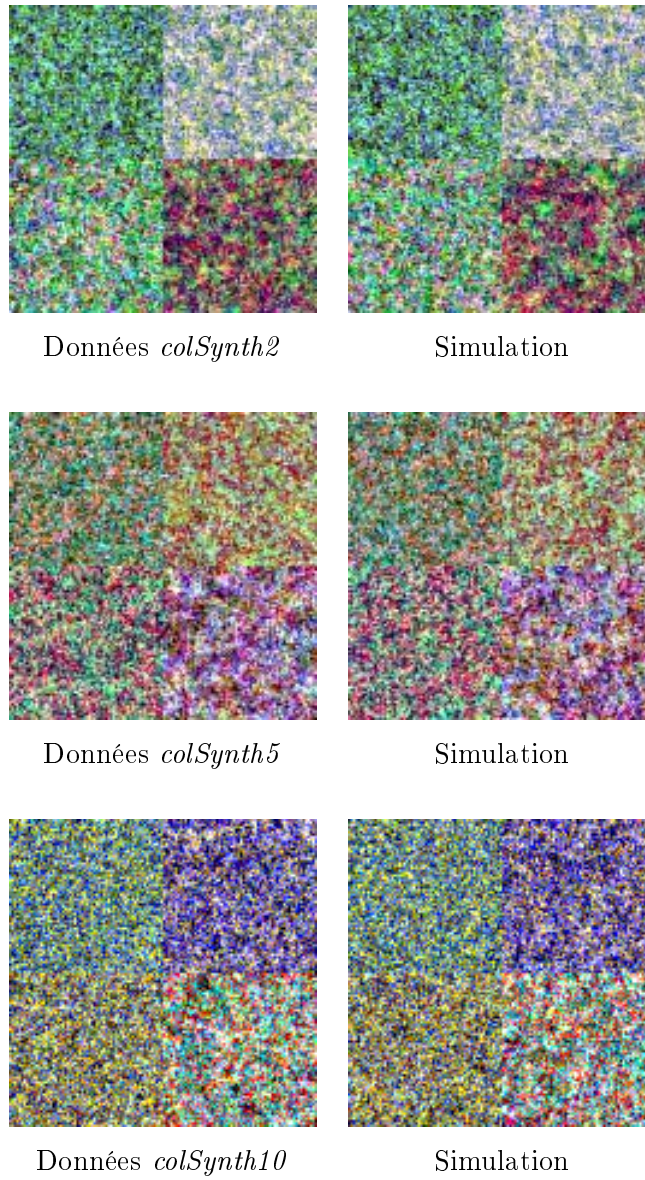


FIG. 2.2 – Données simulées à différents rapports Signal sur Bruit (S/B) (colonne de gauche), simulation à partir des paramètres estimés pour chaque quart de l'image (colonne de droite).

	<i>colSynth2</i> N-O		
	Réel	Estimé MV	Estimé MPV
Paramètre a	0.8	0.805	0.770
Matrice Σ : écart moyen des coefficients (%)	-	2.5	3.8
Matrice Σ : écart maximal en %	-	6	7.3

	<i>colSynth2</i> S-O		
	Réel	Estimé MV	Estimé MPV
Paramètre a	0.85	0.855	0.852
Matrice Σ : écart moyen en %	-	3	3
Matrice Σ : écart maximal en %	-	8.6	8.3

	<i>colSynth2</i> N-E		
	Réel	Estimé MV	Estimé MPV
Paramètre a	0.9	0.906	0.843
Matrice Σ : écart moyen en %	-	1	2.7
Matrice Σ : écart maximal en %	-	2.6	4.2

	<i>colSynth2</i> S-E		
	Réel	Estimé MV	Estimé MPV
Paramètre a	0.95	0.959	0.958
Matrice Σ : écart moyen en %	-	1.3	1.3
Matrice Σ : écart maximal en %	-	3.8	3.8

TAB. 2.2 – Paramètres réels et estimés des 4 parties (NO, SO, NE, SE) de l'image col-Synth2.

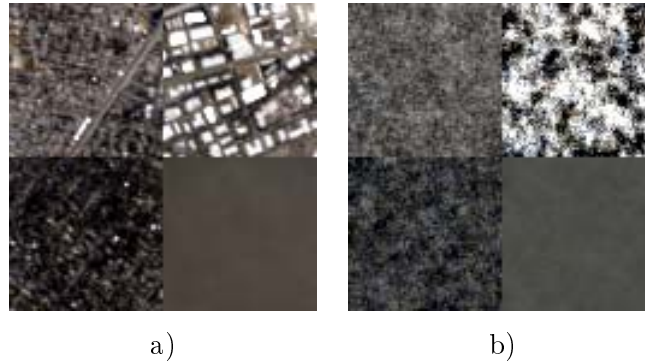


FIG. 2.3 – a) Image initiale, b) Image simulée avec les paramètres extraits de chaque quart de l'image a), par l'estimateur MV.

plus grand de paramètres à estimer, et donc demande un nombre très grand d'échantillons d'apprentissage. De plus, si l'algorithme fonctionne avec une fenêtre glissante pour le calcul des paramètres sur toute l'image, cette fenêtre devra avoir une très grande taille, ce qui diminue la précision en terme de localisation (des bords, notamment). Un autre problème est que l'augmentation du nombre de paramètres rend l'estimation de ceux-ci plus coûteuse et moins robuste, en particulier pour ce qui est du critère MV, où l'optimisation est effectuée numériquement.

2.4 Modèle markovien gaussien 3D

La modélisation markovienne gaussienne peut prendre plusieurs formes, dont le MGMRF que l'on a détaillé au paragraphe précédent. Dans ce dernier, l'image est considérée comme un champ à 2 dimensions spatiales de variables vectorielles que sont les "hyperpixels", ceux-ci étant les mesures en chaque site, de dimension égale au nombre de bandes spectrales. Les interactions explicitement modélisées sont donc uniquement les interactions spatiales. Les interactions spectrales sont prises en compte à travers la matrice de covariance conditionnelle, de façon implicite.

Le type d'images que l'on utilise dans ce travail, les images hyperspectrales, sont caractérisées par un échantillonnage spectral très fin. On peut donc avoir, pour les images du capteur AVIRIS par exemple, plus de 200 bandes spectrales, un nombre du même ordre de grandeur que les dimensions spatiales. Dans ces conditions, il paraît concevable d'imaginer une modélisation où la dimension spectrale est une dimension à part entière. Il existe de tels modèles, où l'image est considérée comme un champ à trois dimensions de variables scalaires. Un site est donc paramétré par ses deux coordonnées spatiales et sa coordonnée

spectrale. Les interactions sont explicites entre les sites voisins, au sens spectral comme au sens spatial.

Si ces interactions sont explicites, elles sont aussi définies par un jeu de paramètres réduit, du fait de l'hypothèse markovienne. Cela mène à des modèles d'images hyperspectrales caractérisés par des paramètres dont le nombre est plus faible et qui par conséquent échappent aux problèmes d'estimation dans les espaces de grande dimension.

Dans ce chapitre, on présente un modèle gaussien de ce type [5, 67, 73].

Une limitation de ce modèle est qu'il suppose une stationnarité spectrale qui n'est en général pas vérifiée. Le paragraphe 2.5 permet d'apporter un début de solution à ce problème, en proposant une méthode de partition du spectre en groupes de bandes où la stationnarité est une hypothèse plus réaliste. Le modèle peut alors être appliqué sur chacun des groupes indépendamment.

2.4.1 Loi conditionnelle et loi globale

Considérons une image x de dimensions spatiales $N_h \times N_v$ et de dimension spectrale N_b . Cette image est supposée être une réalisation du champ X . On suppose donc que X est un champ à 3 dimensions de variables scalaires. Supposant une moyenne nulle pour le champ considéré (l'adaptation à un champ de moyenne non nulle est directe), la modélisation markovienne gaussienne implique que la probabilité d'un vecteur en un site $s = (i, j, k)$ connaissant le reste de l'image est égale à la probabilité en ce site connaissant ses voisins, c'est-à-dire :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2\sigma^2} \left(x_s - \sum_{t \in V_s} \beta_{t-s} x_t \right)^2 \right\} \quad (2.51)$$

où σ^2 est un paramètre du modèle appelé *variance conditionnelle*, et les paramètres $\{\beta_t\}$ sont les *paramètres d'interaction* entre un site et un de ses voisins. Dans le cas, par exemple, d'un voisinage au premier ordre, les voisins d'un site $s = (i, j, k)$, sont les points t paramétrés de la façon suivante :

$$t_i = (i \pm 1, j, k), t_j = (i, j \pm 1, k), t_k = (i, j, k \pm 1) \quad (2.52)$$

Ce voisinage en « 6-connexité » est l'équivalent du voisinage en 4-connexité pour un réseau à deux dimensions. Ces points sont représentés sur la figure 2.4.

On peut écrire, d'après l'équation (2.51) :

$$x_s = \sum_{t \in V_s} \beta_{t-s} x_t + u_s \quad (2.53)$$

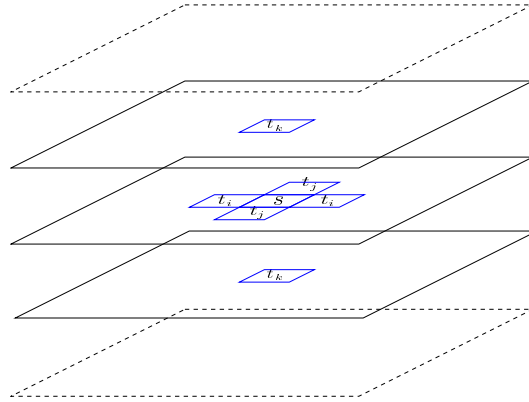


FIG. 2.4 – Un site s et ses différents voisins en 6-connexité dans un modèle 3D.

où la variable aléatoire scalaire u_s est un bruit gaussien, centré, de variance σ^2 .

On peut obtenir la loi globale de ce champ, utile pour le calcul de la vraisemblance des paramètres. Pour cela, on écrit, à partir de la formule (2.53), avec x qui représente l'image réorganisée en vecteur colonne : $x = (x_{(0,0,0)}, x_{(0,0,1)}, \dots, x_{(N_h, N_v, N_b)})$; et de même $u = (u_{(0,0,0)}, u_{(0,0,1)}, \dots, u_{(N_h, N_v, N_b)})$:

$$u = Bx \quad (2.54)$$

où B est une matrice formée des paramètres d'interaction β_t , organisés comme expliqué en annexe A.

De plus, u suit une loi normale donc x également, car c'est une transformation linéaire de u . On peut donc écrire pour x :

$$P(x) = \frac{1}{Z(S)} \exp\left(-\frac{1}{2}x^t S^{-1}x\right) \quad (2.55)$$

avec par définition $S = E\{xx^t\} = B^{-1}E\{uu^t\}$, et $Z(S)$ constante de normalisation.

Or, d'après le principe d'orthogonalité, on doit avoir :

$$E\{u_i x_j\} = \delta_{i,j} \sigma^2, \quad \forall i, j \in S \quad (2.56)$$

Cela se traduit matriciellement par $E\{ux^t\} = \sigma^2 I$, où I représente la matrice identité de dimension $N_h \times N_v \times N_b$, et donc $S = \sigma^2 B^{-1}$. L'expression de la loi globale devient :

$$P(x) = \frac{1}{Z(S)} \exp\left(-\frac{1}{2\sigma^2}x^t Bx\right) \quad (2.57)$$

Cette loi est donc entièrement caractérisée par la variance conditionnelle σ et la matrice B . Or, cette matrice B est elle-même entièrement définie par le jeu de paramètres β_i . Le modèle est donc paramétré par un jeu réduit de paramètres, ce qui le rend robuste face au phénomène de Hughes.

2.4.2 Estimation des paramètres

Un critère que l'on a évoqué au paragraphe 2.3.7, très utilisé pour l'estimation de paramètres, est celui du Maximum de Vraisemblance (MV).

Le problème consiste donc à trouver le jeu de paramètres $\phi = (\sigma^2, \{\beta_t\})$, tel que :

$$\phi = \arg \max_{\phi} (P(x|\phi)) \quad (2.58)$$

Or, en explicitant la constante de normalisation dans l'équation (2.57), on peut exprimer la loi globale du champ comme :

$$P(x|\phi) = \frac{|B|^{\frac{1}{2}}}{(2\pi\sigma^2)^{\frac{N_v N_h N_b}{2}}} \exp\left(-\frac{1}{2\sigma^2} x^t B x\right) \quad (2.59)$$

La log-vraisemblance $LV(\phi)$ s'écrit donc, en omettant la constante et en multipliant par 2 :

$$LV(\phi) = \ln |B| - N_h N_v N_b \ln \sigma^2 - \frac{1}{\sigma^2} x^t B x \quad (2.60)$$

Tout comme le calcul de la vraisemblance au paragraphe précédent, il est possible de calculer la log-vraisemblance des paramètres et de l'optimiser. Pour l'optimiser, il est possible d'utiliser la vraisemblance réelle et une méthode de gradient avec une réduction des paramètres comme pour le MGMRF. Il est également possible d'utiliser des approximations, comme dans [73], où les auteurs utilisent un développement limité pour approcher l'expression du déterminant de B .

Pour déterminer le maximum de vraisemblance, on commence par diminuer le nombre de paramètres, comme pour le MGMRF. On va donc calculer, pour un jeu de paramètres $\{\beta_t\}$ donné, la variance conditionnelle minimisant la vraisemblance.

$$\frac{\partial}{\partial \sigma^2} LV = -N_h N_v N_b \frac{1}{\sigma^2} + \frac{1}{(\sigma^2)^2} x^t B x \quad (2.61)$$

On reconnaît la forme connue de la dérivée partielle pour un champ gaussien sans interaction. La valeur de $\hat{\sigma}^2$, annulant cette dérivée et minimisant l'expression, est :

$$\hat{\sigma}^2 = \frac{x^t B x}{N_h N_v N_b} = \frac{1}{N_h N_v N_b} \sum_s x_s u_s \quad (2.62)$$

L'expression de la log-vraisemblance (2.60) en $\phi = (\{b_t\}, \hat{\sigma}^2(\{b_t\}))$ devient :

$$LV(\{b_t\}) = \ln |B| - N_h N_v N_b \ln \frac{x^t B x}{N_h N_v N_b} - N_h N_v N_b \quad (2.63)$$

Le jeu de paramètres est donc réduit, et il suffit alors de minimiser cette expression par rapport aux $\{b_t\}$. On peut effectuer la minimisation par un algorithme de gradient conjugué, puisque le gradient du champ peut être calculé de la même façon que pour le MGMRF. En effet, une expression de la dérivée du deuxième terme est :

$$\begin{aligned} \frac{1}{N_h N_v N_b} \frac{\partial}{\partial b_k} (x^t B x) &= \frac{1}{N_h N_v N_b} \frac{\partial}{\partial b_k} (\sum_s x_s u_s) \\ &= -\frac{1}{N_h N_v N_b} (\sum_s x_s x_{t_k-s}) \\ &= -C_k \end{aligned} \quad (2.64)$$

où C_k est la corrélation entre un pixel et son voisin de type k .

Quant à la dérivée du premier terme, le déterminant de B , le calcul fait en annexe A permet d'écrire :

$$\frac{\partial}{\partial b_k} \ln |B| = (B^{-1})_k \quad (2.65)$$

où $(B^{-1})_k$ est le coefficient de B^{-1} correspondant à un déplacement dans le champ de type k . Il s'agit, comme pour le MGMRF, de la corrélation théorique entre un pixel et son voisin de type k , comme le montre la formule (2.59).

Quand la vraisemblance est minimale, il y a égalité entre la corrélation empirique et la corrélation théorique pour chacun des types de voisinage.

2.4.3 Restriction à un modèle spectral

Un cas particulier du modèle précédent est celui où les interactions spatiales ne sont pas prises en compte. Les paramètres d'interaction correspondant aux voisinages spatiaux sont donc nuls, et on se retrouve avec uniquement des interactions spectrales. Cela revient donc à une modélisation simplement de la réponse spectrale.

$$x_{ijk} = \sum_{n=-N}^N \beta_n x_{ij(k+n)} + u_{ijk} \quad (2.66)$$

où N est l'ordre du voisinage.

De même que précédemment, la loi globale du champ est :

$$P(x) = \frac{1}{Z(S)} \exp \left(-\frac{1}{2\sigma^2} x^t B x \right) \quad (2.67)$$

Or B est telle qu'il n'y a pas d'interaction entre les voisins spatiaux.

On peut donc écrire B de la façon suivante :

$$B = I_{N_v N_h} \otimes C \quad (2.68)$$

avec

$$C = I_{N_b} - \sum_{n=1}^N \beta_n C_{N_b}^n \quad (2.69)$$

où $C_{N_b}^n = (c_{kl}^n)$, avec $c_{kl}^n = \delta_{(|k-l|=n)}$.

Donc, si l'on appelle $\mathbf{x}_{ij} = (x_{ij0}, \dots, x_{ij(N_b-1)})^t$ le vecteur représentant l'« hyperpixel » situé aux coordonnées spatiales (i, j) , on peut écrire :

$$P(\mathbf{x}_{ij}) = \frac{1}{Z(B)} \exp \left(-\frac{1}{2\sigma^2} \mathbf{x}_{ij}^t B \mathbf{x}_{ij} \right) \quad (2.70)$$

Ceci est donc un modèle gaussien non-contextuel classique, avec une seule nuance : la matrice de covariance est structurée par l'équation (2.69).

Il s'agit donc d'une manière de régulariser les matrices de covariances, méthodes souvent utilisées comme des alternatives à la réduction de dimension pour éliminer le phénomène de Hughes [29, 42]. Dans les cas cités, la seule différence est que l'on cherche à régulariser la matrice de covariance en la supposant combinaison linéaire de différentes matrices plus ou moins proches de la matrice de covariance estimée (matrice identité, matrice de covariance réduite à sa diagonale, matrice de covariance générale de l'image...).

Ici, la modélisation impose une structure à la matrice inverse de la matrice de covariance (à travers la matrice B). En fait, la matrice B est l'approximation de la matrice inverse de la matrice de covariance. Et puisque le champ (dans sa dimension spectrale) est supposé stationnaire, c'est une approximation de la matrice de corrélation. D'autre part, on sait que l'estimation des paramètres par maximum de vraisemblance aboutit à des valeurs de paramètres tels que la matrice est inversible, ce qui est aussi le but de cette régularisation.

Il est facile de prouver, car c'est un cas particulier du cas présent, que le jeu de paramètres $\{\beta_n\}$ correspondant au maximum de vraisemblance est celui pour lequel les corrélations théoriques, c'est-à-dire les coefficients de la matrice B^{-1} (ou, de façon équivalente C^{-1}), sont égaux aux corrélations mesurées dans l'image entre un pixel et ces N premiers voisins.

La figure 2.5 montre les différences entre les coefficients de B^{-1} et les corrélations mesurées pour l'ensemble d'apprentissage d'une image AVIRIS correspondant à de l'eau. Il apparaît que l'ajout progressif de paramètres permet d'affiner progressivement l'approximation de la corrélation, grossière au départ (avec un seul paramètre) jusqu'à obtenir une approximation assez proche de la corrélation réelle.

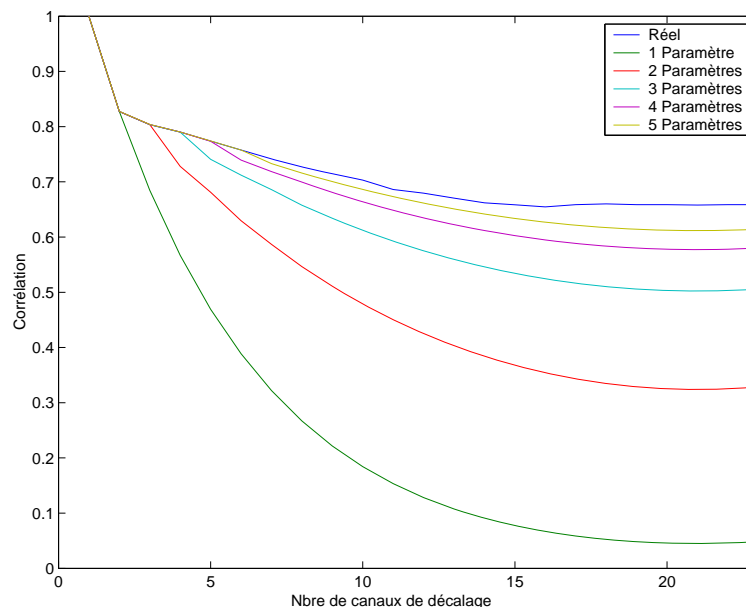


FIG. 2.5 – Corrélation théorique entre deux bandes en fonction de leur distance en nombre de bandes, et approximation de celle-ci par le modèle spectral, pour des nombre de paramètres allant de 1 à 5.

2.5 Partition du spectre en zones homogènes

Comme expliqué au paragraphe 2.3.6, l'utilisation du spectre en entier dans le cadre du modèle MGMRF est problématique, car cela rend infondées les hypothèses simplificatrices du modèle. D'autre part, dans le cadre du modèle 3D présenté au paragraphe 2.4, il est également peu recommandable de construire celui-ci sur l'ensemble du spectre, car cela suppose la stationnarité du signal dans le domaine spectral, hypothèse qui ne se vérifie en général pas. En effet, la figure 2.6, où l'on a affiché la matrice de corrélation (normalisée) correspondant à une images AVIRIS, fait apparaître clairement des zones où les corrélations sont homogènes. Mais on a, d'autre part, des bandes qui ne sont pas corrélées avec les autres, qui correspondent aux images de bruit que l'on a dans les longueurs d'onde d'absorption de certains gaz présents dans l'atmosphère.

On voit clairement que 4 groupes de bandes adjacentes aux caractéristiques spectrales homogènes se détachent de l'image de corrélation de la figure 2.6. Ces groupes se détachent en formant des blocs de fortes corrélation dans la diagonale de la matrice de corrélation. Ces groupes ont donc la propriété d'être composés de bandes très corrélées entre elles et beaucoup moins avec les bandes des autres groupes.

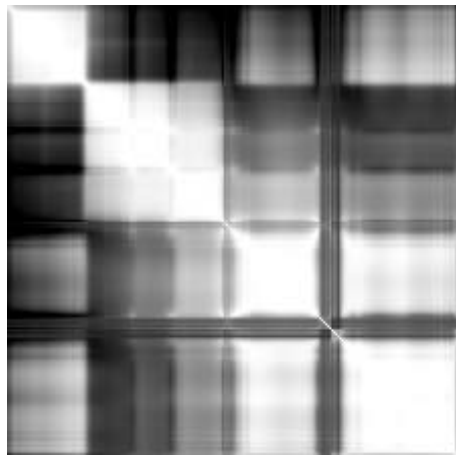


FIG. 2.6 – Image de la matrice de corrélation d'une scène AVIRIS.

L'algorithme de partition du spectre consistera, dans un premier temps, à éliminer les bandes non significatives, puis dans un deuxième temps, à diviser itérativement un des groupes de bandes de manière à augmenter au maximum l'homogénéité de ceux-ci.

2.5.1 Elimination des bandes non informatives

Dans un premier temps, les bandes dont la variance est inférieure à un seuil η sont éliminées, car il s'agit de bandes correspondant aux longueurs d'ondes que certains gaz présents dans l'atmosphère (oxygène, vapeur d'eau) absorbent. Il résulte qu'une bande où le signal est faible est une bande qui a subi une très forte atténuation du rayonnement par les gaz. Les images correspondant à ces longueurs d'onde ne contiennent que du bruit.

2.5.2 Ajout de bornes

Une fois obtenu un spectre réduit où toutes les bandes sont informatives, il reste à le partitionner en zones homogènes de bandes adjacentes.

Le processus est itératif. Il vise à ajouter à chaque étape une nouvelle borne de séparation de manière à créer deux groupes à partir d'un seul, afin d'augmenter au maximum l'homogénéité des groupes résultants.

L'homogénéité de la partition se calcule comme la moyenne de l'homogénéité des différents groupes considérés indépendamment. Quant à l'homogénéité d'un groupe de bandes adjacentes, il s'agit de la moyenne des coefficients de la matrice de corrélation de ce groupe.

Pour chaque position possible de la nouvelle borne de séparation, cet indice est calculé, et la position retenue est celle qui le maximise.

2.5.3 Algorithme

L'algorithme de partition du spectre est celui décrit ci-dessous (cf. algorithme 1). Le nombre final de groupes N est choisi arbitrairement, en s'appuyant, par exemple, sur la connaissance que l'on a du domaine du spectre imagé par le capteur. L'initialisation est faite en considérant un unique groupe de bandes, constitué de celles gardées après élimination des bandes non-informatives. Ensuite, on ajoute une borne de séparation à chaque itération jusqu'à obtention du nombre de groupes N désiré.

Algorithme 1 Partition du spectre en groupes homogènes

Calcul de la matrice de corrélation $C = \{c_{ij}\}$

Elimination des bandes dont la variance est inférieure à un seuil η .

Soit N le nombre de groupes désiré et n_b le nombre de bandes restant après l'élimination.

pour $n = 2$ à N **faire**

$\sigma_{min} \leftarrow \infty$

pour $i = 1$ à n_b **faire**

i nouvelle borne de la partition en groupes $\{G_k\}_{k=1,\dots,n}$, contenant chacun n_k bandes

Calculer l'homogénéité $\sigma = \frac{1}{n} \sum_{k=1}^n \sigma_k$

avec $\sigma_k = \frac{1}{n_k^2} \sum_{l,m \in G_k} |c_{lm}|$

si $\sigma < \sigma_{min}$ **alors**

$\sigma_{min} \leftarrow \sigma$

$i_{min} \leftarrow i$

fin si

fin pour

i_{min} choisi comme nouvelle borne définitive de la partition en groupes $\{G_k\}_{k=1,\dots,n}$

fin pour

La figure 2.7 illustre les itérations de l'algorithme, pour l'image dont la matrice de corrélation est représentée en a). Les résultats donnés sont les traces sur la matrice de covariance des différents blocs de bandes adjacentes retenus. L'image 2.7.b) est la première étape de l'algorithme où les bandes non significatives sont enlevées, et les images c) et d) montrent les partitions à 2 et 3 groupes respectivement. On voit que, lors du passage de 2 à 3 groupes de bandes, c'est le premier groupe qui a été divisé en deux. Les partitions semblent bien refléter la structure du spectre original. Et la partition à trois groupes paraît suffisante pour dégager des groupes homogènes.

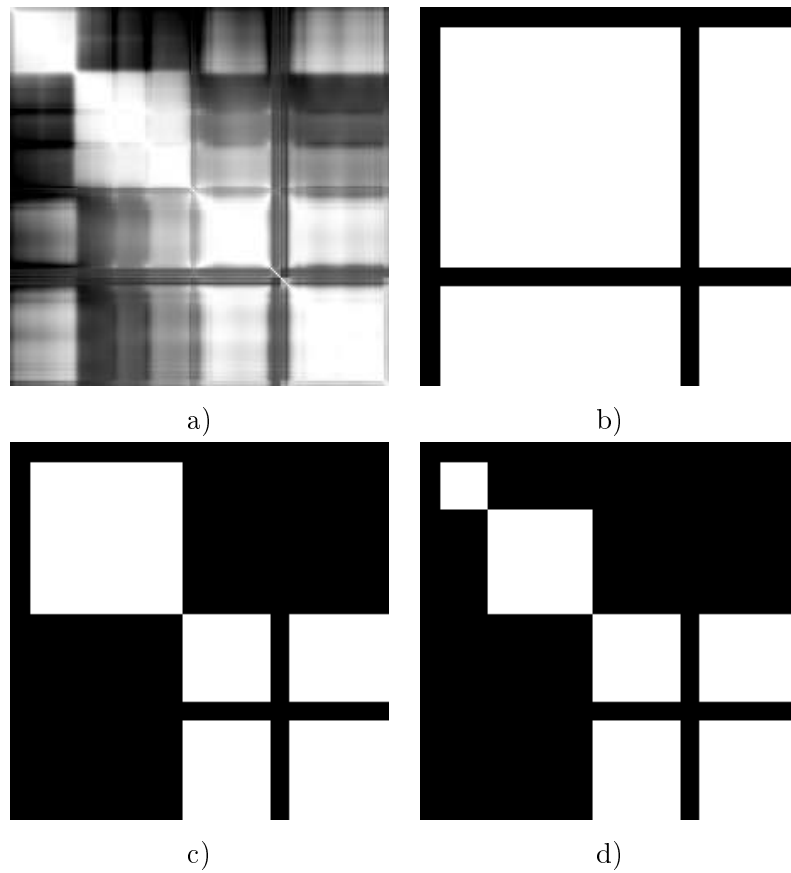


FIG. 2.7 – a) Matrice de corrélation de l'image analysée, b) représentation bandes retenues pour la partition, c) et d) représentation des blocs homogènes pour des partitions à 2 groupes (c) et 3 groupes (d).

2.6 Conclusion

Dans ce chapitre, on détaille les modèles markoviens gaussiens applicables aux images 3D que constituent les images hyperspectrales, afin d'en analyser la texture spatialement et spectralement. On propose une adaptation du champ appelé MGMRF, fondée sur la simplification des paramètres d'interaction, qui permet d'une part, d'éviter l'effet de Hughes et d'autre part, de dériver des estimateurs simples et plus stables. Il s'avère que cette simplification, moyennant des hypothèses sur le domaine des longueurs d'ondes de l'image, est réaliste.

On rappelle également un modèle markovien 3D, c'est-à-dire avec une modélisation markovienne explicite sur la dimension spectrale. On montre comment cette modélisation peut être utilisée pour la régularisation des matrices de covariance.

Enfin, la dernière partie de ce chapitre est consacrée à la présentation d'un algorithme de partition du spectre en zones homogènes, permettant de se mettre dans les conditions de validité des hypothèses simplificatrices du premier modèle.

Chapitre 3

Réduction d'espace

La nécessité et la justification de la réduction de l'espace de définition des données ont été détaillées au premier chapitre. Dans ce chapitre, on présente différentes méthodes pour effectuer cette réduction.

La première partie du chapitre aborde des méthodes récurrentes dans la littérature concernant ce problème et, dans la deuxième partie, une méthode de réduction d'espace, fondée sur l'analyse de mélange linéaire, est proposée.

On a montré, au chapitre 1, qu'il était intéressant d'utiliser pour le traitement des images hyperspectrales, non pas les mesures faites dans l'espace initial de grande dimension, mais une projection de celles-ci dans un espace de plus petite dimension. Dans cette partie, on examine différents critères et différentes méthodes de projection.

3.1 Approches générales

Les méthodes suivantes considèrent un critère énergétique calculé sur l'image dans son ensemble, et non un critère relatif à la distinction des classes. Contrairement aux méthodes que l'on verra plus tard, elles ne sont pas optimales par rapport à des critères de classification, mais elles sont simples et donnent tout de même de bons résultats.

3.1.1 Analyse en composantes principales

Parmi les méthodes existantes, la méthode la plus simple, mais qui reste efficace, est l'analyse en composantes principales (ACP) [27, 50]. Le principe de cette méthode est de déterminer les « composantes principales » (CP) et de représenter le signal dans l'espace généré par les n premières CP, espace qui sera donc de dimension n . Les composantes principales sont un ensemble de vecteurs ayant 3 particularités intéressantes : premièrement, elles forment une base orthogonale de l'espace de départ ; deuxièmement, chaque composante principale est décorrélée de toutes les autres au regard du nuage de points total de l'image ; troisièmement, elles sont ordonnées (de façon décroissante) selon la variance que présente le nuage de points dans la direction des composantes.

Dans la base initiale, la matrice de covariance peut s'exprimer :

$$\Sigma = P^t \Lambda P \tag{3.1}$$

Avec Λ la matrice de covariance dans la base après rotation des axes telle que :

$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{pmatrix} \quad (3.2)$$

avec les valeurs propres λ_i telles que $\lambda_1 > \lambda_2 > \dots > \lambda_N$.

En prenant comme espace de représentation les n premières CP (qui correspondent aux n plus grandes valeurs propres), on est assuré d'avoir une approximation du signal telle que la moyenne de l'erreur sur l'image entière est nulle et sa variance minimale.

Cette méthode est optimale au sens de l'énergie du signal, c'est-à-dire que pour une valeur de n donnée, il est impossible de déterminer une base dans laquelle l'erreur quadratique moyenne entre le signal et son approximation soit inférieure. Pour n fixé, la projection sur les n premières CP est celle garantissant une perte d'énergie minimale. L'inconvénient de cette méthode est le temps de calcul qu'elle nécessite. Les composantes principales sont les vecteurs propres de la matrice de covariance *totale* de l'image, et cette matrice est longue à calculer. Ensuite, le calcul des vecteurs propres de la matrice peut être long, si le nombre de bandes N est très grand. Pour ce calcul, il est possible de profiter de la structure par blocs de la matrice pour diviser le problème en une somme de sous-problèmes plus simples [50].

Le choix de la dimension d'arrivée n peut être fait selon un critère énergétique, c'est-à-dire que si l'on choisit un rapport $\eta \in [0,1]$, on prendra n tel que dans la base d'arrivée :

$$\sum_{i=0}^n E\{y_i^2\} > \eta \sum_{i=0}^N E\{y_i^2\} \quad (3.3)$$

où y_i sont les coordonnées transformées et centrées en zéro, et $E\{\cdot\}$ représente l'espérance mathématique. Le paramètre η est choisi en fonction de la perte énergétique de signal que l'on s'autorise à avoir.

3.1.2 *Minimum Noise Fraction*

La transformation appelée *Minimum Noise Fraction* (MNF) est une variante de l'ACP, ajustée au bruit [35]. Cette transformation consiste en une rotation de la base afin d'obtenir une nouvelle base orthogonale où les vecteurs sont ordonnés suivant le rapport signal sur bruit (S/B), et non suivant la variance comme pour l'ACP.

Sa mise en œuvre pratique se décompose en deux ACP successives. La première ACP se fait par rapport aux statistiques du bruit, supposées connues. A l'issue de cette étape, on connaît une base orthogonale dans laquelle le bruit est décorrélé. La base intermédiaire obtenue à l'issue de cette étape est la base de vecteurs propres de la matrice de covariance

du bruit *normalisés* de sorte que cette matrice dans la nouvelle base soit la matrice identité. On a donc *sphérisé* le bruit.

La matrice de covariance du bruit Σ_n peut s'écrire comme suit :

$$\Sigma_n = P^{-1}IP \quad (3.4)$$

où la matrice de passage P est constituée de vecteurs orthogonaux mais n'ayant pas forcément une norme égale à 1, et I est la matrice identité.

Une fois dans cette base, on effectue la deuxième ACP, cette fois en considérant la matrice de covariance des données (incluant le bruit). On obtient une base de vecteurs propres, telle que la valeur propre associée à un vecteur de base est le rapport S/B des données selon ce vecteur.

$$\Sigma_s = P^{-1}Q^t\Lambda QP \quad (3.5)$$

où la matrice de passage Q est une matrice orthonormée, et Λ une matrice diagonale.

La matrice de passage totale pour passer de la base initiale à la base MNF est donc QP .

On est capable avec cette transformation de déterminer la dimension intrinsèque des données. En effet, si un vecteur de la base finale a pour valeur propre associée une valeur bien supérieure à 1, c'est que cette composante contient surtout de l'information et de la variance due à la variabilité propre des données. Tandis que si ce vecteur de base a pour valeur propre associée une valeur proche de 1, c'est que sa variance est principalement due au bruit. Ceci peut être une manière de déterminer une valeur de n adéquate, puisque les bandes contenant principalement du bruit ne sont pas intéressantes.

Le problème de cette méthode est qu'il faut avoir une connaissance *a priori* sur le bruit, ce qui n'est pas forcément possible si l'on ne connaît pas les caractéristiques du capteur, ou les conditions d'acquisition des données.

3.2 Approches pour la classification

Pour la classification, d'autres méthodes peuvent être envisagées considérant non plus un critère énergétique comme l'ACP ou de rapport signal sur bruit comme la MNF, mais étant fondées sur la distinction entre classes.

Celles-ci sont mises en œuvre dans le cadre de la segmentation supervisée, puisqu'elles nécessitent une définition préalable des statistiques des classes. Or, calculer des statistiques devient moins trivial quand la dimension de l'observation devient de plus en plus grande. C'est l'effet de Hughes [46].

Avec ces approches, il est possible de perdre beaucoup d'information, si celle-ci n'est pas facteur de discrimination entre les classe définies. Parmi elles, deux méthodes nous ont paru intéressantes : l'*analyse discriminante* et l'algorithme de *poursuite de projection*.

3.2.1 Analyse discriminante

L'*analyse discriminante* [31] consiste en la maximisation d'un critère lié à la dispersion inter-classe et la dispersion intra-classe. La dispersion inter-classe est la dispersion des centres des classes que l'on doit maximiser, alors que la dispersion intra-classe est la dispersion d'une même classe autour de son centre, qui doit être minimisée.

On définit donc deux matrices, la matrice de dispersion intra-classe :

$$S_w = \sum_i P(\omega_i) S_i \quad (3.6)$$

et la matrice de dispersion inter-classe :

$$S_b = \sum_i P(\omega_i) (m_i - m_0)(m_i - m_0)^t \quad (3.7)$$

avec $m_0 = \sum_i P(\omega_i) m_i$, et m_i , S_i et $P(\omega_i)$ respectivement la moyenne, la matrice de covariance et la probabilité *a priori* de la classe ω_i .

On définit $S_w^d = d^t S_w d$ et $S_b^d = d^t S_b d$, comme étant les projections des matrices définies plus haut par une matrice de projection orthogonale d permettant de passer de l'espace initial, au espace de dimension inférieure fixée.

On peut construire différents critères à partir de ces matrices, comme ceux recensés dans [31]. En pratique, on utilise le plus souvent le critère :

$$J = tr((S_w^d)^{-1} S_b^d) \quad (3.8)$$

qu'il faut maximiser par rapport à d .

L'optimisation conduit à un espace dont la dimension est limitée par le nombre de classes, car $S_w^{-1} S_b$ est de rang égal au nombre de classes moins 1.

D'autre part, si les classes ne sont pas assez uniformément réparties dans l'espace initial, les résultats de classification sont moins bons, notamment si deux classes ont des moyennes très proches. Malgré sa forte dépendance aux statistiques calculées dans l'espace initial, la méthode reste intéressante de par sa rapidité.

3.2.2 Poursuite de projection

Une autre méthode appelée poursuite de projection (*projection pursuit* en anglais) [30, 45, 52] permet de s'affranchir du problème de l'effet de Hughes. Son principe réside dans la recherche d'un sous-espace dans lequel un indice de projection défini par rapport à l'objectif du traitement, calculé à partir des statistiques des classes *dans ce sous-espace*, est maximisé.

L'intérêt de cette méthode est de calculer les statistiques dans l'espace projeté, ce qui rend plus robustes les estimations. Il est malheureusement irréalisable de mettre en pratique des algorithmes exacts pour trouver le sous-espace solution (temps de calcul prohibitif), mais il existe des algorithmes approchés rapides, que l'on expose par la suite. Un autre intérêt de cette méthode est la possibilité d'adapter l'indice de projection en fonction du traitement que l'on veut effectuer sur les données, comme la classification.

Nous discuterons plus en détails de cette méthode générale, de ses différentes mises en œuvre approchées ainsi que du choix des indices de projection, au chapitre 4, qui lui est consacré.

3.3 Méthode sémantique-statistique

Les images hyperspectrales, grâce à l'échantillonnage fin du spectre de rayonnement des sols imagés, permettent une distinction plus facile de la nature de ces sols, par comparaison, par exemple, aux images multispectrales. Pour cette raison, il est courant des les utiliser pour des applications comme la détection ou la quantification d'un ou plusieurs matériaux considérés comme des cibles. Dans ces cas, il est nécessaire de connaître certaines informations sur la (ou les) cible(s) et éventuellement sur la scène imagée. Ces informations peuvent être par exemple la réflectance du matériau, sa composition, ou les conditions géométriques ou météorologiques de l'acquisition. Cependant, la capacité de ces images à déterminer la présence ou la quantité de matériaux est intéressante, et l'on peut chercher à l'exploiter sans ces connaissances *a priori*, moyennant l'adaptation des algorithmes existants d'analyse de mélange. Dans ce paragraphe, on examine une solution permettant d'utiliser de telles méthodes pour effectuer une réduction de dimension.

3.3.1 Analyse de mélange

Une des applications possibles est l'estimation de mélanges. Ici, les spectres des matériaux purs sont supposés connus.

On considère le modèle de mélange linéaire suivant :

$$x_i = As_i + w_i \tag{3.9}$$

où x_i est le vecteur hyperspectral de dimension N mesuré au pixel i , A est la matrice $N \times M$ dont les colonnes a_i sont les spectres purs, s_i est un vecteur de dimension M dont la m -ième composante représente la proportion du spectre m dans le pixel i , et enfin w_i est un bruit, souvent considéré comme blanc, gaussien et centré. Les images hyperspectrales étant, en général, constituées de 20 à 200 bandes, le nombre de spectres purs M que l'on recherche est donc dans la majorité des cas inférieur à la dimension initiale N . De plus, par définition, les composantes de s_i prennent des valeurs positives et leur somme est égale à 1.

Cette analyse de mélange peut être considérée comme une réduction de dimension (de N à M), nécessitant une connaissance importante, celle des colonnes de la matrice A , c'est-à-dire des spectres purs. Ce modèle rappelle le modèle dit *Cocktail party model*, utilisé par Nuzillard et Bijaoui pour la séparation aveugle de sources [66], en imagerie multispectrale astronomique.

3.3.2 Résolution de l'analyse de mélange à A connue

Soit le problème d'analyse de mélange défini précédemment. Supposons les M colonnes de A connues, et linéairement indépendantes. Pour des raisons de clarté, on omet l'indice i correspondant au pixel examiné. Appelons y la mesure non bruitée $y = As$. La transformation permettant le passage de x à s n'est pas une simple projection orthogonale dans un espace vectoriel. En effet, d'une part les spectres purs n'ont pas de raison *a priori* d'être ni orthogonaux entre eux, ni normés. D'autre part, la contrainte imposée sur les coefficients fait que le vecteur y se situe dans un sous-ensemble de l'espace généré par ses vecteurs de base, les colonnes de A , comme le montre la figure 3.1.

Le domaine de y , sans la contrainte de positivité et de somme unitaire sur s , est l'espace \mathcal{E} engendré par les colonnes de A (ie. le plan dans la figure 3.1). Si l'on raisonne en termes d'espaces affines, ces deux contraintes sur s font que le domaine se réduit à la partie convexe du sous-espace affine \mathcal{A} défini comme l'espace affine naturel de \mathcal{E} . Ce domaine convexe est celui compris entre les extrémités des spectres purs (ie. le segment pointillé dans la figure 3.1).

Maintenant que le domaine D_y de y est défini, il reste à trouver la correspondance entre x et y , c'est à dire la projection dans le convexe défini précédemment. Si l'on suit le critère du maximum de vraisemblance et la règle de Bayes, on doit maximiser :

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \quad (3.10)$$

c'est-à-dire, si l'on considère un *a priori* uniforme :

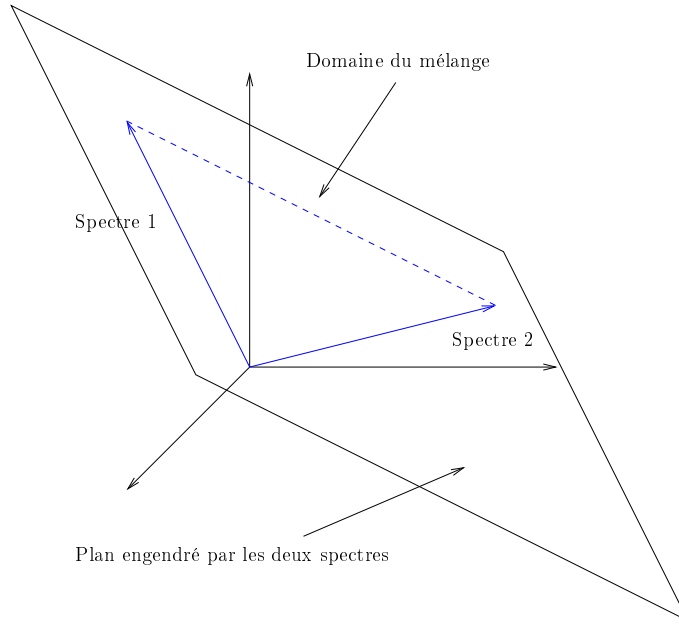


FIG. 3.1 – Exemple en 3 dimensions de 2 spectres purs (flèches bleues) définissant un domaine de mélange (pointillés bleus), sous ensemble de l'espace engendré par ceux-ci (le plan figuré par le parallélogramme en noir).

$$\hat{y} = \arg \max_{y \in D_y} p(x|y) \quad (3.11)$$

et $p(x|y)$ est une loi gaussienne centrée sur y , correspondant à la loi du bruit. Deux cas peuvent se présenter alors. Dans le cas où la loi est connue, on la sphérise en effectuant une rotation et une transformation d'échelle (ce qui revient à faire une analyse en composantes principales du bruit, suivie d'une normalisation des valeurs propres). Dans le cas contraire, on suppose que celle-ci est centrée, de matrice de covariance proportionnelle à l'identité.

Une fois la loi du bruit sphérisée, le problème de la détermination de \hat{y} revient alors à la détermination du point du domaine D_y le plus proche de x en norme euclidienne.

Enfin, il est possible d'obtenir le vecteur \hat{s} des proportions à partir du vecteur estimé \hat{y} , car celui-ci définit un, et un seul, vecteur \hat{s} .

3.3.3 Calcul de \hat{y}

Le calcul du point

$$\hat{y} = \arg \min_{y \in D_y} (\|y - x\|) \quad (3.12)$$

ou, de la même façon :

$$\hat{y} = \arg \min_{y \in D_y} (\|y - x\|^2) \quad (3.13)$$

n'est pas des plus simples car, comme mentionné précédemment, il s'agit d'une projection non dans un sous-espace entier, mais dans un domaine convexe fermé de ce sous-espace.

Soit \tilde{y} la projection orthogonale de x dans l'espace engendré par les spectres purs. On a :

$$\|y - x\|^2 = \|x - \tilde{y}\|^2 + \|\tilde{y} - y\|^2 \quad (3.14)$$

car $x - \tilde{y}$ et $\tilde{y} - y$ appartiennent à des espaces orthogonaux. Donc :

$$\hat{y} = \arg \min_{y \in D_y} \|\tilde{y} - y\|^2 \quad (3.15)$$

C'est-à-dire que le problème se résume à trouver le vecteur de D_y le plus proche de la projection de x dans l'espace \mathcal{E} .

Pour déterminer ce point, il est plus aisé de se placer à nouveau en géométrie affine. Plaçons-nous donc dans \mathcal{A} , espace affine naturel de \mathcal{E} centré sur l'origine. Soit $\tilde{\mathbf{Y}}$ le point tel que $\overrightarrow{O\tilde{\mathbf{Y}}} = \tilde{y}$. De la même façon, la notation en gras, lettres capitales, des autres variables de \mathcal{E} signifiera leur correspondant dans \mathcal{A} .

On note \mathcal{A}' l'hyperplan affine passant par les M points \mathbf{A}_i correspondant aux M spectres purs a_i , vecteurs colonnes de A . Le domaine \mathbf{D}_y est inclus dans cet hyperplan. De même que plus haut, il est aisé de montrer que le problème de la détermination de $\hat{\mathbf{Y}}$ revient à trouver le point de \mathbf{D}_y le plus proche de la projection $\tilde{\mathbf{Y}}'$ de \mathbf{X} sur \mathcal{A}' :

$$\hat{\mathbf{Y}} = \arg \min_{\mathbf{Y} \in \mathbf{D}_y} \|\tilde{\mathbf{Y}}' - \mathbf{Y}\|^2 \quad (3.16)$$

Soit un point quelconque $\tilde{\mathbf{Y}}'$ dans cet hyperplan. Il existe une famille de $\{\alpha_i\}$ unique telle que $\overrightarrow{A_0\tilde{\mathbf{Y}}'} = \sum_{i=1}^{M-1} \alpha_i \overrightarrow{A_0A_i}$. Ce qui signifie que $\tilde{\mathbf{Y}}'$ est un barycentre des M points A_i de poids α_i , avec $\alpha_0 = 1 - \sum_{i=1}^{M-1} \alpha_i$. La somme des α_i est égale à 1.

Pour connaître $\hat{\mathbf{Y}}$, calculons les $\{\alpha_i\}$. Deux cas peuvent se produire :

- ceux-ci sont tous positifs ou nuls,
- un ou plusieurs α_i sont négatifs.

Dans le premier cas, tout va bien car $\tilde{\mathbf{Y}}'$ est dans \mathbf{D}_y , donc $\tilde{\mathbf{Y}}' = \hat{\mathbf{Y}}$. Dans le deuxième cas, la projection de $\tilde{\mathbf{Y}}'$ sur le polytope est égale à sa projection sur la facette de celui-ci correspondant aux sommets pour lesquels les coefficients sont positifs.

On peut donc calculer $\hat{\mathbf{Y}}$ par un algorithme itératif, en calculant les coefficients barycentriques de la projection de \mathbf{X} dans l'espace affine généré par les points courants. Si ces coefficients sont tous positifs, $\hat{\mathbf{Y}}$ est le barycentre de ces points pondérés par les coefficients trouvés. Si un ou plusieurs coefficients sont strictement négatifs, on réduit l'ensemble de points courants pour ne garder que ceux pour lesquels les coefficients sont positifs ou nuls.

3.3.4 Utilisation de l'analyse de mélange pour la réduction d'espace

Le problème qui nous empêche d'utiliser directement cette approche est qu'elle nécessite la connaissance des colonnes de la matrice A , c'est-à-dire des matériaux composant la scène, ainsi que de leur réponse spectrale.

On va s'intéresser à la réduction de dimension dans le cadre de l'analyse de mélange sans connaissance *a priori* sur les spectres purs présents dans la scène.

En effet, l'objectif n'est pas exactement de reconnaître physiquement un type de sol, c'est-à-dire à travers le matériau dont il est constitué, mais de reconnaître une occupation *sémantique* du sol. Contrairement au cas des zones rurales, la différence entre ces deux approches n'est pas mince dans le cas des zones urbaines. En effet, dans les zones rurales, l'occupation d'un terrain est représentée par le matériau qui s'y trouve (cultures, forêt, roche...). Dans le cas des zones urbaines, la situation est différente, pour les deux raisons suivantes. Premièrement, dans le cas qui nous occupe, il est difficile de connaître *a priori* les cibles à chercher. On peut savoir à l'avance que certains matériaux seront présents de façon certaine, mais on ne peut pas tous les recenser. Deuxièmement, ces cibles ne sont en général ni caractéristiques des types d'occupation de sols urbains, ni nécessairement présentes dans toute occurrence du type d'occupation.

Pour ces raisons, il n'est pas nécessaire, de réaliser une correspondance directe entre un spectre pur et le matériau physique qu'il représente. Ce qui nous intéressera alors, ce sont les caractéristiques statistiques de ces proportions. Mais la modélisation par mélange linéaire n'en devient pas inintéressante pour autant, si elle est valide. On passe donc par la modélisation par mélange pour déterminer ces grandeurs de proportion. C'est l'obtention de ces proportions de spectres purs en faible nombre qui constitue la réduction de la dimension.

3.3.5 Recherche des spectres purs

Afin de trouver les spectres purs, on utilise une méthode qui s'apparente à la poursuite de projection (que l'on présente plus en détails au chapitre 4). Supposons que l'on connaisse le nombre de composantes pures M présentes dans la scène. La recherche de celles-ci va consister en la recherche d'une base, non obligatoirement orthonormée, dans laquelle une

combinaison de deux contraintes est satisfaite :

- Le modèle est le plus vraisemblable, c'est à dire que la probabilité conditionnelle de l'ensemble des vecteurs aléatoires $X = \{X_i\}_{i \in \Omega}$, $P(x|A,s)$, est maximisée. Considérant que les variables aléatoires X_i sont indépendantes :

$$\hat{A} = \arg \max_A (P(x|A,s)) = \arg \max_A \left(\prod_i P(x_i|s_i,A) \right) \quad (3.17)$$

$$\hat{A} = \arg \min_A \left(\sum_i \|w_i\|^2 \right) \quad (3.18)$$

où le bruit w_i a été défini par la formule 3.9

- Chaque composante pure possède au moins un représentant dans le nuage. C'est-à-dire :

$$\forall k, \exists x_i / (\hat{s}_i)_k = 1 \quad (3.19)$$

L'algorithme utilisé consistera à considérer chaque colonne de A séquentiellement, et de choisir pour la colonne a_m courante le vecteur \hat{a}_m qui minimise l'indice $I(A,x)$ défini de la façon suivante :

$$I(A,x) = \sum_i \|w_i\|^2 + \lambda \max_m \left(\min_i \|x_i - a_m\| \right) \quad (3.20)$$

Le premier terme concerne la vraisemblance du modèle (le premier point évoqué plus haut), et le deuxième terme concerne la présence dans l'image de représentants des spectres les plus purs possibles. La raison de cette dernière contrainte qui peut apparaître superflue, est que sans celle-ci, l'algorithme va avoir tendance à déterminer des spectres purs tels que le simplexe déterminé par leurs sommets sera beaucoup trop grand, de manière à englober tous les points du nuage.

L'optimisation de l'indice pour chaque a_m est effectuée par un algorithme numérique, la descente du simplexe, qui ne nécessite pas le calcul de dérivées d'aucun ordre. L'initialisation des spectres est discutée au paragraphe suivant.

3.3.6 Utilisation d'un espace privilégié

Considérons une image suivant le modèle linéaire de mélange. En chaque site i , la valeur x_i du vecteur aléatoire X_i est régie par l'équation :

$$x_i = As_i + w_i \quad (3.21)$$

Si l'on considère la matrice de variance-covariance de l'image, on a :

$$\Sigma_X = \frac{1}{N} \sum_i x_i x_i^t \quad (3.22)$$

or, x_i est la somme de y_i , le signal, et de w_i , le bruit. Ces deux vecteurs peuvent être considérés comme indépendants, donc :

$$\Sigma_X = \Sigma_Y + \Sigma_W \quad (3.23)$$

Dans le cas d'un rapport signal à bruit suffisant, la matrice de covariance des vecteur X_i est proche de celle des Y_i . Ajoutons que Σ_Y est une matrice de rang $M - 1$, si M est le nombre de spectres purs. En effet, les y_i sont tous situés dans un espace à $M - 1$ dimensions.

De plus, si le rapport S/B est suffisant, la matrice Σ_X sera de la forme :

$$\Sigma_X = P^t \Lambda_X P \quad (3.24)$$

où

$$\Lambda_X = \text{diag}(\lambda_0^Y, \dots, \lambda_{M-2}^Y, \lambda_{M-1}^W, \dots, \lambda_{N-1}^W) \quad (3.25)$$

Dans cette écriture, les λ_i^Y (resp. λ_i^W) ne sont pas les valeurs propres de Σ_Y (resp. Σ_W), mais bien les valeurs propres de Σ_X . L'exposant signifie que la valeur est principalement influencée par Y ou W .

Ceci a plusieurs conséquences :

- La première concerne le choix de M . En effet, *a priori* ce nombre est inconnu. On peut se fonder sur les valeurs propres de Σ_X pour déterminer M , puisque les λ_i^Y sont supposées avoir une valeur supérieure aux λ_i^W , qui sont, elles, homogènes et faibles. Il faut donc placer un seuil sur la somme des valeurs propres, et retenir le nombre de vecteurs M tel que la somme des M premières valeurs propres soit supérieure au seuil.
- Une autre conséquence concerne l'initialisation de l'algorithme de recherche des spectres purs. En effet, il paraît plus logique d'initialiser ceux-ci à l'intérieur du nuage, et dans les directions des axes principaux de celui-ci. Ceci permet de saisir dès l'initialisation la forme du nuage, tout en ayant des vecteurs linéairement indépendants.
- Enfin, on peut aussi exploiter la forme de la matrice Σ_X pour restreindre l'espace de recherche des spectres purs. On effectue la recherche des spectres purs dans l'espace engendré par les M premiers vecteurs propres. Cet espace est de dimension bien

inférieure à l'espace initial, ce qui permet une recherche plus rapide. Après convergence, il est aussi possible d'affiner la recherche en repassant dans l'espace initial. Cet affinement se fait en général, en très peu d'itérations.

Ceci est illustré par la figure 3.2. Ce tableau recense les valeurs propres de la matrice de covariance de deux images synthétiques de dimension 20 construites selon le modèle linéaire de mélange, ici avec 4 composantes pures et avec des rapport S/B de 5.35 dB et 8.72 dB. On constate que dans les deux cas, la chute des valeurs propres est très rapide, et que celles-ci se stabilisent une fois que le nombre $M - 1$ est dépassé. En revanche, il est difficile, pour l'image à faible rapport S/B, de se fonder sur le cumul des valeurs propres, puisque lorsque le nombre $M - 1$ de valeurs propres est atteint, le cumul des valeurs atteint seulement 80.7%, ce qui est un seuil assez bas. En revanche, si l'on se fonde sur la pente de la chute de valeurs, le choix du juste nombre M est évident.

3.3.7 Résultats

Images de synthèse

On teste la méthode de recherche des spectres purs sur des images de synthèse générées par le modèle de mélange de l'équation (3.9). Ces images sont construites de la façon suivante : on choisit la dimension N de l'espace initial, on tire au hasard un nombre donné M de vecteurs dans cet espace qui constituent les spectres purs, et enfin chaque pixel est le résultat d'un mélange de ces spectres pondérés par des coefficients aléatoires additionné d'un bruit gaussien de matrice de covariance proportionnelle à l'identité. La figure 3.3 montre des exemples de projections de ce type de nuage dans des espaces à 3 dimensions.

On réalise sur de telles images la recherche des composantes pures, dont on suppose le nombre connu. Les tableaux 3.1 et 3.2 donnent des exemples de résultats de recherche des spectres purs (SP) pour des mélanges de 3 et 4 spectres, pour différents rapports signal sur bruit (S/B). On remarque que, dans tous les cas, l'erreur d'estimation est faible (inférieure à 5% de la norme moyenne). D'autre part, quand le rapport signal sur bruit est plus faible, les spectres retrouvés présentent un indice bien inférieur aux spectres réels, car les hypothèses sont moins valides. On remarque aussi que l'obtention d'un minimum global de l'indice n'est pas garantie, puisque les spectres réels présentent, dans certains cas, des indices inférieurs aux spectres estimés.

Images réelles

On teste cet algorithme de recherche des spectres purs et d'analyse de mélange conjointes dans le cas de scènes réelles.

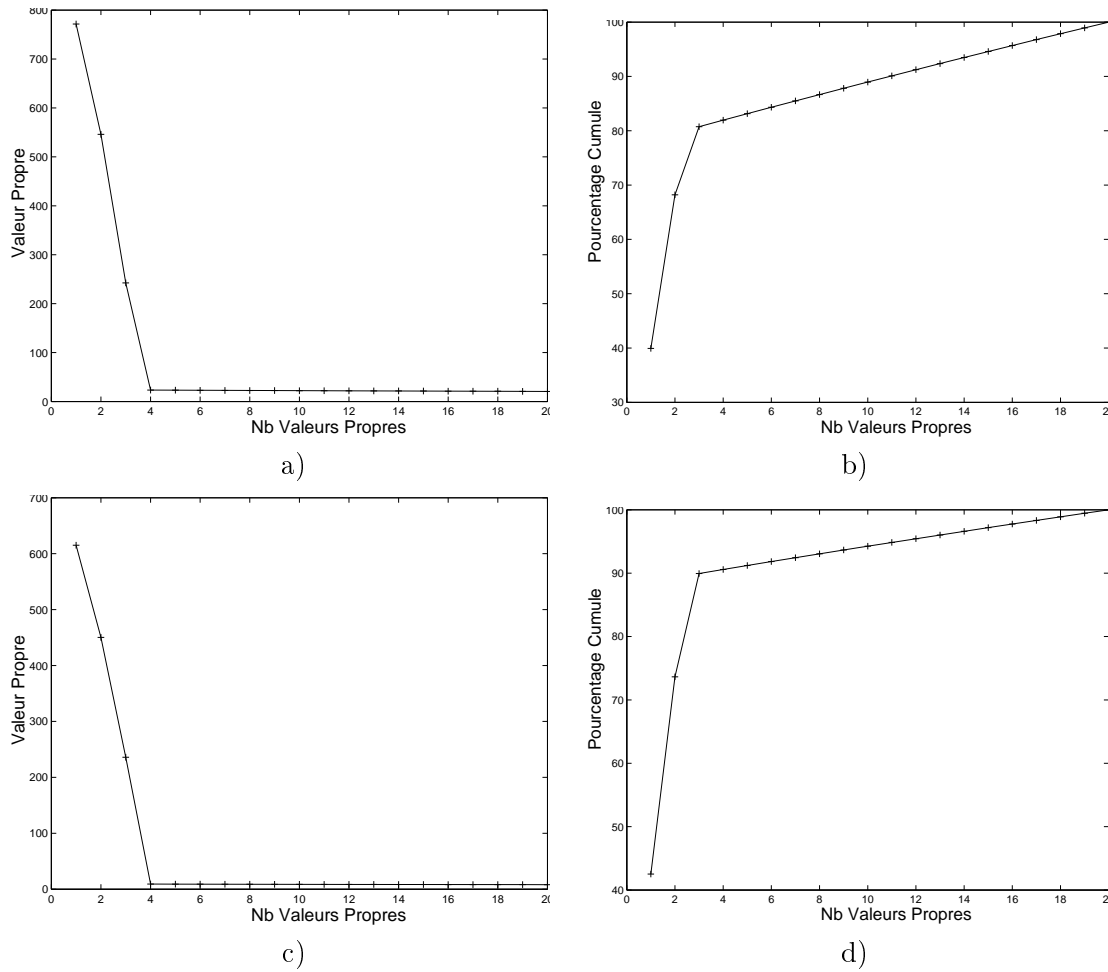


FIG. 3.2 – a) Evolution des valeurs propres pour une image de synthèse ($S/B=5.35$ dB), b) évolution du cumul de celles-ci pour la même image (en pourcentage du total), c) évolution des valeurs propres pour une image de synthèse ($S/B=8.72$ dB), d) évolution du cumul de celles-ci pour la même image (en pourcentage du total).

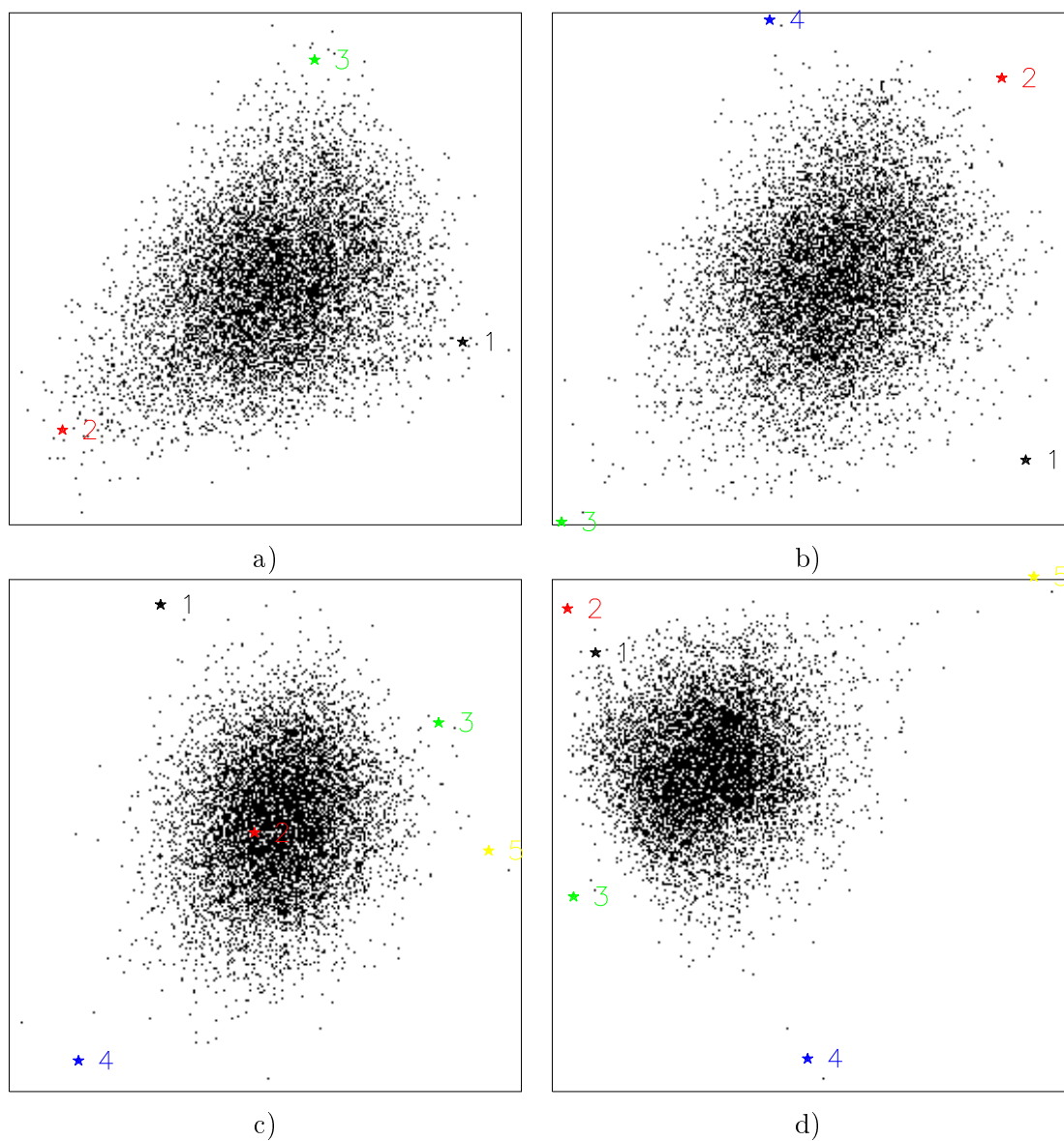


FIG. 3.3 – Exemple de nuages de points issus de mélanges : a) 3 composantes pures, b) 4 composantes pures, c) 5 composantes pures, d) 5 composantes pures avec des coefficients non-uniformément distribués.

Test (S/B en dB)	Indice Référence	Indice Estimé	Erreur d'estimation des SP (% de la norme moyenne)
1 (14.95)	3.02	3.37	1.52
2 (13.12)	6.48	6.99	2.03
3 (12.22)	4.58	4.37	1.07
4 (17.57)	2.53	2.34	1.00
5 (6.46)	12.76	12.40	2.08

TAB. 3.1 – *Résultat de l'estimation des spectres purs dans le cas d'un mélange à 3 composantes. 5 tests sont effectués à différents rapport S/B. Les deux premières colonnes de résultats donnent la valeur de l'indice pour les spectres purs réels et estimés, la dernière colonne donne l'erreur d'estimation des spectres purs, en pourcentage de la norme moyenne du signal.*

Test (S/B en dB)	Indice Référence	Indice Estimé	Erreur d'estimation des SP (% de la norme moyenne)
1 (12.04)	6.63	6.22	1.31
2 (14.48)	3.59	2.89	1.37
3 (9.73)	8.64	6.37	2.21
4 (16.05)	2.07	2.29	1.55
5 (7.70)	26.53	16.88	4.70

TAB. 3.2 – *Résultat de l'estimation des spectres purs dans le cas d'un mélange à 4 composantes. 5 tests sont effectués à différents rapport S/B. Les deux premières colonnes de résultats donnent la valeur de l'indice pour les spectres purs réels et estimés, la dernière colonne donne l'erreur d'estimation des spectres purs, en pourcentage de la norme moyenne du signal.*



FIG. 3.4 – Extrait d'une scène réelle en zone rurale (affichage de 3 bandes du visible).

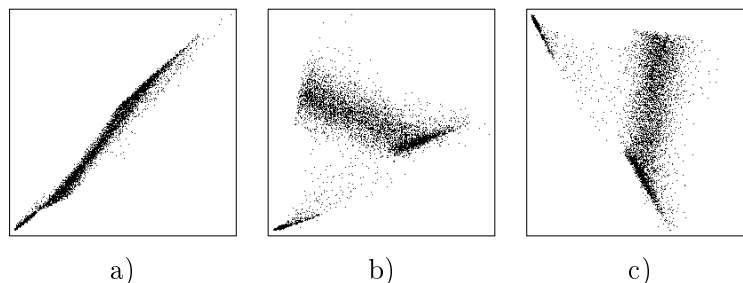


FIG. 3.5 – Projection du nuage de points correspondant à la figure 3.4 dans trois plans différents : a) plan défini par les bandes 10 et 15, b) plan défini par les bandes 15 et 35, c) plan défini par les bandes 20 et 30.

La figure 3.4 montre une partie d'une image AVIRIS représentant un cas simple du problème qui nous occupe (en zone rurale, néanmoins). L'image est composée de trois types d'utilisation du sol : l'eau (partie supérieure), la forêt (en vert), et le sol nu (reste). De plus, certains pixels sont des mélanges de sol nu et de forêt (au bord de celle-ci). La détermination du nombre de spectres purs à chercher (3) est conforme à l'analyse de la matrice de covariance de la scène. En effet, 99.7 % de l'énergie se trouve dans les 2 premières valeurs propres, ce qui reflète une structure à 2 dimension du nuage, liée à la présence de 3 spectres purs.

Trois projections du nuage dans des plans sont montrées sur la figure 3.5. La forme du nuage en triangle ou, plus précisément, en boomerang, y est clairement visible.

Les résultats de la figure 3.6 montrent les proportions trouvées des différentes composantes pures (au nombre de 3). Les résultats semblent en accord avec ce que l'on peut observer dans l'image, car les trois types de sol sont bien séparés.

Ceci montre un cas simple où l'on distingue bien la présence de différents matériaux, même "à l'œil". Dans le cas de zones urbaines, il est plus difficile, d'une part de connaître *a priori* le nombre de classes, mais également de valider les résultats. La figure 3.7 montre 3 bandes du spectre visible d'un extrait d'image AVIRIS acquise à Moffett Field, en Californie.

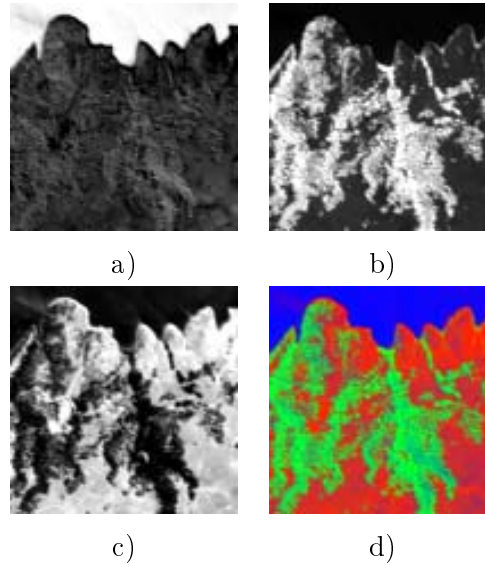


FIG. 3.6 – a) Proportion du spectre pur correspondant à l'eau, b) proportion du spectre pur correspondant à la forêt, c) proportion du spectre pur correspondant au sol nu, d) image en couleur des trois proportions (rouge : sol nu, vert : forêt, bleu : eau).

Cet extrait est principalement composé de zones urbaines, de différents types, mais on observe également la présence de forêt, de sol nu et d'eau, notamment dans la zone au milieu à droite.

Pour cette image, il est évident que le nombre de spectres purs est supérieur à trois. Dans ce cas, 99,7 % de l'énergie du signal se trouve dans les 3 premières valeurs propres, on cherche donc un mélange formé de 4 composantes.

Les spectres retrouvés sont visibles sur la figure 3.8 où ils sont représentés dans le nuage de points. On montre également dans la figure 3.9 comment le nuage est projeté dans le polyèdre convexe défini comme le domaine de s , par les extrémités des spectres purs estimés situés aux coins de l'image. On voit que d'une part, la densité de points est assez élevée et plutôt homogène dans tout le polyèdre, et d'autre part que peu de points se trouvent sur les arêtes où les faces de celui-ci, ce qui montre que peu de points avaient leur projection dans le sous-espace engendré par les spectres purs en dehors du polyèdre. La première remarque montre que le polyèdre est suffisamment petit pour qu'il n'apparaisse par trop de zones dépeuplées (même si leur apparition est inévitable), et la deuxième montre que le polyèdre est suffisamment grand pour que peu de projections dans le sous-espace se fassent en dehors de celui-ci. Il y a donc un bon compromis à ce niveau.

La figure 3.10 montre les proportions des 4 spectres estimés sur ces image. On peut



FIG. 3.7 – *Extrait d'une scène réelle en zone urbaine (affichage de 3 bandes du visible).*

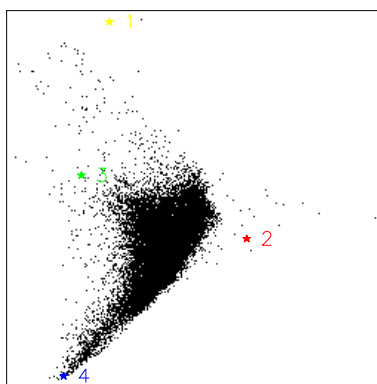


FIG. 3.8 – *Projection du nuage de points de l'image 3.7 et des 4 spectres purs, sur un plan de projection arbitraire.*

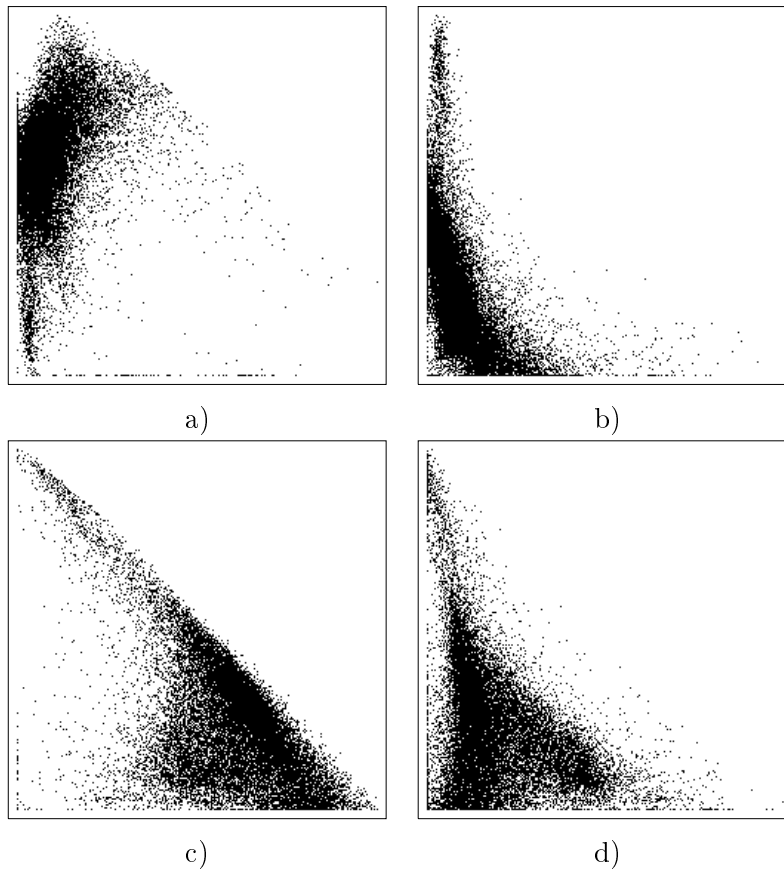


FIG. 3.9 – *Projection du nuage de points S_i : a) les proportions dans le plan 1-2, b) dans le plan 1-4, c) dans le plan 2-4, d) dans le plan 3-4.*

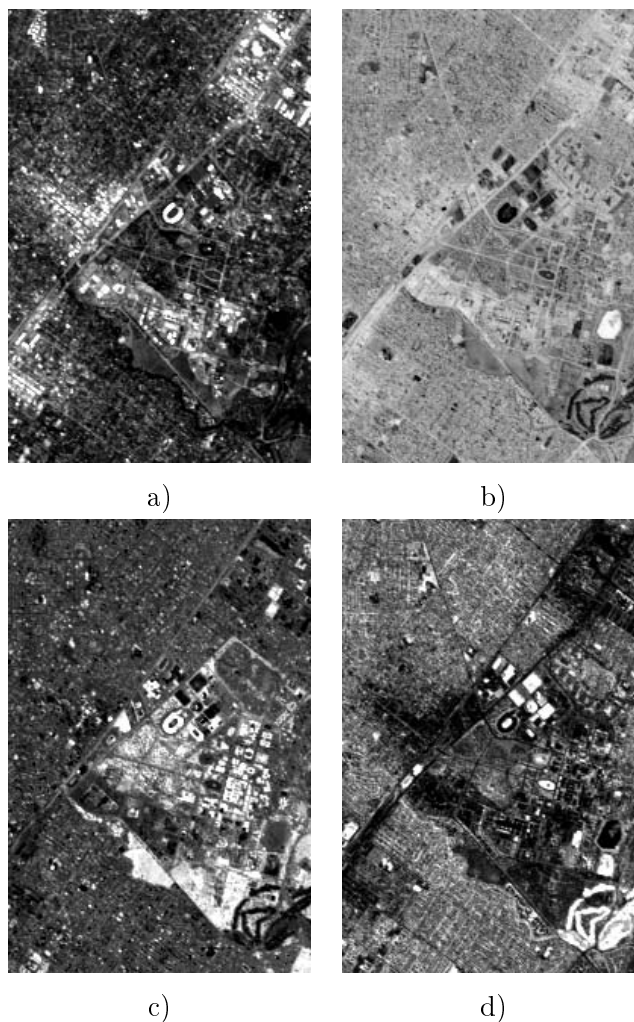


FIG. 3.10 – Proportions des 4 spectres purs estimés.

reconnaître par exemple en c) le sol nu et en d) l'herbe, tandis que l'on remarque que l'eau en b) recueille aussi les contributions des ombres, dans la plus grande partie de l'image.

3.4 Conclusion

On a recensé, dans ce chapitre, des méthodes de réduction d'espaces cherchant à préserver l'énergie du signal, et par conséquent applicables dans un éventail très large d'applications, ainsi que des méthodes spécialement mises au point pour effectuer une réduction d'espace mettant en valeur la discrimination de classes prédéfinies, dans l'optique de la classification. La méthode de réduction de dimension que l'on propose, en fin de chapitre,

se fonde sur une hypothèse de mélange linéaire de composantes spectrales pures pour la formation des images. Cette méthode a pour but de déterminer ces composantes pures par un algorithme dérivé de la poursuite de projection, et d'utiliser comme « image réduite » l'image obtenue des proportions de ces spectres purs. Ceci résulte en une réduction drastique de la dimension, puisque l'algorithme est capable de reconnaître un nombre réduit de spectres. Néanmoins, la perte d'information est minimale (légèrement plus grande que pour l'ACP), et le contenu sémantique des bandes extraites est plus grand, comme le montrent les exemples proposés.

Chapitre 4

Poursuite de projection

La poursuite de projection consiste en une famille de méthodes visant à déterminer de « bons » sous-espaces dans lesquels projeter des données. L'intérêt de ces méthodes est double : elles autorisent celui qui les utilise à choisir sa propre définition d'un bon sous-espace, et évitent certains problèmes dus à la grande dimensionnalité des données.

La première partie de ce chapitre rappelle deux algorithmes permettant de mettre en œuvre la poursuite de projection et d'obtenir une solution approchée du problème d'optimisation du sous-espace.

La deuxième partie traite du choix de la grandeur appelée *indice de projection* que l'on cherche à optimiser par la poursuite de projection. On rappelle le choix opéré par Jimenez et Landgrebe pour cet indice, et on adapte ce choix à la modélisation par champs de Markov gaussien que l'on a proposée.

4.1 Présentation générale

La poursuite de projection [45] est une méthode visant à rechercher un sous-espace dans lequel un indice de projection est maximisé.

Cette méthode paraît intéressante pour deux raisons :

- le phénomène de Hughes est atténué du fait que tous les calculs sont faits dans le sous-espace projeté,
- l'indice de projection peut être adapté selon le traitement visé.

Initialement [45], la poursuite de projection était utilisée dans des contextes non paramétriques, afin de trouver des angles de projection pour lesquels les données seraient « bien » représentées. Les indices de projection étaient alors des mesures de non gaussiannité liées à l'histogramme du nuage de points.

Plus récemment, cette méthode a été utilisée dans un contexte paramétrique dans le cadre de la classification d'image hyperspectrale [52]. Ayant choisi un modèle gaussien non-contextuel, les auteurs proposent d'appliquer une approximation de la poursuite de projection afin de trouver le sous-espace maximisant la distance de Bhattacharyya entre les classes définies au préalable à travers des échantillons d'apprentissage.

De façon analogue, on propose d'utiliser la distance de Bhattacharyya dans l'application considérée, car cette distance est reliée à une borne supérieure de la probabilité d'erreur [31, 59]. Néanmoins, cette distance sera calculée sur la probabilité conditionnelle définissant le modèle de texture considéré.

La poursuite de projection est une approche globale du problème de la détermination de sous-espace vectoriel, applicable dans différents domaines.

L'approche est très générale puisqu'il s'agit de trouver un sous-espace par projection linéaire dans lequel une fonction, appelée indice de projection, est maximisée ou minimisée. Le choix de l'indice de projection est bien entendu le point crucial de l'algorithme, puisqu'un

bon choix dépend presque intégralement de l'application. Par exemple, un choix d'indice pour la compression pourrait être la variance totale, ce qui revient à faire une analyse en composantes principales.

4.2 Algorithmes approchés

4.2.1 Poursuite de projection dite séquentielle

Dans [52], une mise en œuvre approximée de la poursuite de projection, dite séquentielle, est décrite. On la rappelle ci-dessous (algorithme 2). L'ensemble des bandes est divisé en K sous-groupes (K étant la dimension de l'espace de projection), de chacun desquels on extrait un axe de projection optimal estimé itérativement. Dans la mise en œuvre originale, les K groupes de bandes sont déterminés en divisant l'ensemble initial en groupes de mêmes nombres de bandes. On présente ci-dessous une variante de cet algorithme, dans laquelle les sous-groupes de bandes adjacentes n'ont pas forcément le même nombre de bandes, et sont choisis conformément à un critère que l'on décrit par la suite.

Algorithme 2 Poursuite de projection séquentielle

L'ensemble des bandes est divisé en K groupes de bandes adjacentes

$n \leftarrow 0$, $arret \leftarrow 0$

pour $k = 1$ à K **faire**

Initialiser arbitrairement $v_k = v_k^0$

fin pour

tant que $arret \neq 1$ **faire**

pour $k = 1$ à K **faire**

Soit $v_{k,n}(x) = (v_1^n, \dots, v_{k-1}^n, x, v_{k+1}^{n-1}, \dots, v_K^{n-1})$, avec x combinaison linéaire des vecteurs du k^{eme} groupe de vecteurs.

$v_k^n \leftarrow \arg \max_x I(v_{k,n}(x))$, I étant un indice de projection à définir

fin pour

$v^n = (v_1^n, \dots, v_P^n)$

si $\|v^n - v^{n-1}\| < \epsilon$ **alors**

$arret \leftarrow 1$

fin si

$n \leftarrow n + 1$

fin tant que

Poursuite de projection séquentielle avec groupes de bandes variables

On propose ici une variante de la poursuite de projection séquentielle, pour laquelle on cherche un découpage du spectre en groupes de bandes plus adapté que le découpage à pas constant.

Etant donné que l'on extrait un axe de projection de chacun des groupes, un critère pour le découpage peut être l'homogénéité de la répartition de l'indice de projection dans les différents groupes.

Soit $z = \{z_0, \dots, z_K\}$ les bornes définissant un découpage du spectre en K sous-ensembles de bandes, avec $z_0 = 0$ et $z_K = N_b$, N_b étant le nombre total de bandes. Un groupe de bande k , duquel on va extraire le vecteur v_k de la base d'arrivée, sera donc constitué des bandes de l'espace original comprises entre les borne z_k et z_{k+1} (z_k inclus).

On cherche à trouver le découpage \hat{z} tel que :

$$\hat{z} = \arg \min_z \sigma_I(z) \quad (4.1)$$

avec

$$\sigma_I(z) = \frac{1}{K} \sum_{k=0}^{K-1} (I(z_k, z_{k+1}) - m_I(z))^2 \quad (4.2)$$

où $I(z_k, z_{k+1})$ est la valeur de l'indice correspondant à une projection sur l'espace généré par le groupe des bandes comprises entre z_k et z_{k+1} (z_k inclus), et

$$m_I(z) = \frac{1}{K} \sum_{k=0}^{K-1} I(z_k, z_{k+1}) \quad (4.3)$$

Or

$$\sigma_I(z) = \frac{1}{K} \sum_{k=0}^{K-1} I^2(z_k, z_{k+1}) - m_I^2(z) \quad (4.4)$$

Pour minimiser σ_I , et trouver une estimation de \hat{z} , on examine les bornes z_k , $k = 1, \dots, k-1$ itérativement. La nouvelle valeur de z_k est la valeur qui minimise la fonction $\sigma_{I,k}(x)$, définie comme la valeur de l'indice de projection quand la k -ième borne est en position x , les autres bornes étant fixées. Celle-ci est une fonction convexe de x , x étant compris entre z_{k-1} et z_{k+1} . Ceci est illustré par la figure 4.1. L'algorithme est itéré jusqu'à la convergence, obtenue quand aucune borne ne bouge durant une itération.

Cet algorithme ne garantit pas de trouver le minimum global d'énergie, mais il est plus rapide que les algorithmes d'optimisation globale et aboutit à de bons résultats si

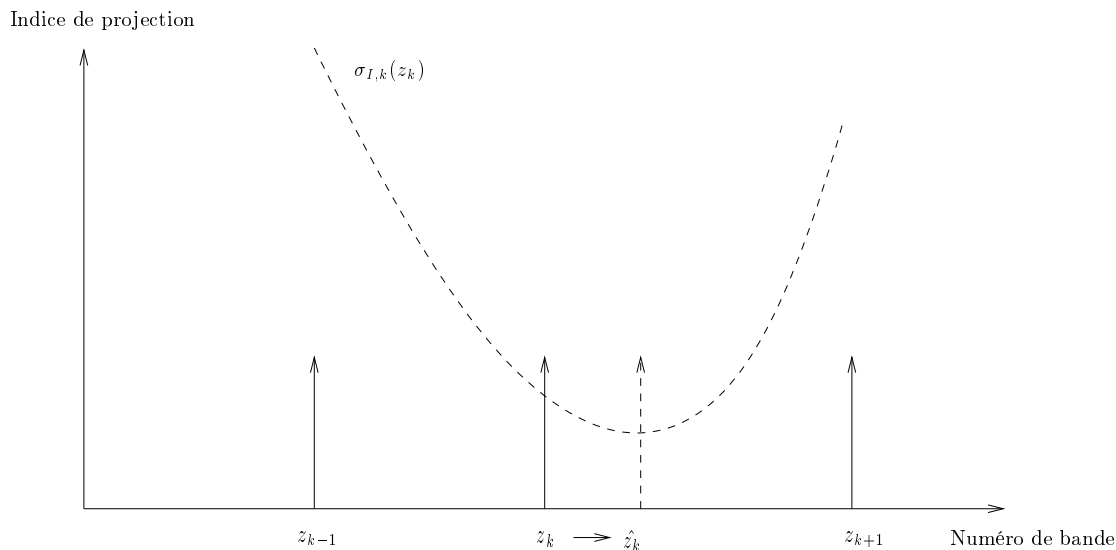


FIG. 4.1 – Illustration de l’algorithme d’ajustement des bornes : pour z_{k-1} et z_{k+1} fixés, la valeur optimale \hat{z}_k de z_k est celle minimisant la fonction $\sigma_{I,k}(z_k)$

l’initialisation est de bonne qualité. Pour ce faire, on choisit z tel que l’on ait un pas constant entre les z_k , c’est-à-dire que l’on considère le découpage choisi dans [52].

La recherche du vecteur optimal pour chaque groupe est effectuée à travers un algorithme d’optimisation numérique, comme le gradient conjugué si le calcul des dérivées est simple (ou tout simplement possible), ou la descente du simplexe dans le cas contraire.

La figure 4.2 montre les évolutions comparées de l’indice optimal¹ estimé en fonction du nombre de bandes dans le cas de deux classes présentes dans des données AVIRIS, pour les deux mises en œuvre de la poursuite de projection séquentielle. Il apparaît (dans ce cas représentatif) qu’il n’est pas toujours bénéfique d’opérer cet ajustement du découpage. En effet, l’ajustement des groupes de bandes est effectué avant la poursuite de projection, et ne se fonde pas sur l’indice de projection que l’on obtiendra en fin de poursuite de projection mais seulement sur l’indice de projection des groupes de bandes. Il n’y a donc pas un lien direct entre le critère utilisé et la distance finale de la base de projection. Ici, quand la dimension finale est supérieur à un seuil de l’ordre de 20 bandes, le nombre de bandes dans les groupes est faible, et peu d’ajustements des groupes sont opérés. En revanche, quand on considère des dimensions finales inférieures à 20, l’indice de projection optimal est supérieur si l’on opère un ajustement des groupes de bandes.

1. Le choix de l’indice est expliqué par la suite : ce graphe ne sert qu’à une évaluation qualitative.

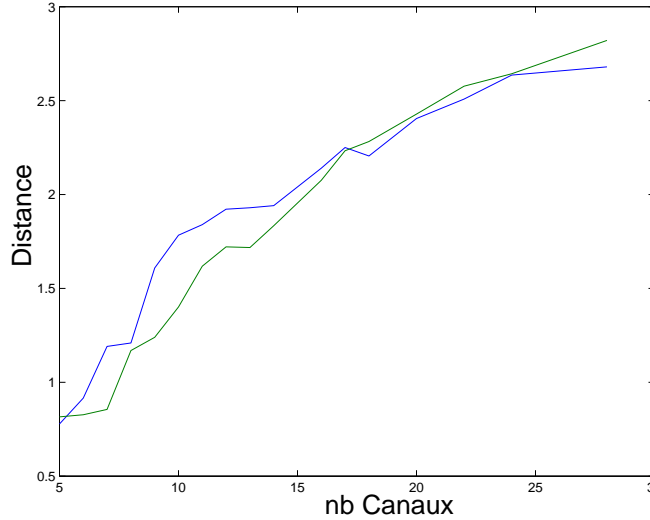


FIG. 4.2 – Distance optimale estimée en fonction du nombre de bandes pour les deux mises en œuvre de la poursuite de projection séquentielle, dans le cas de deux textures d’une même scène AVIRIS. En bleu : mise en œuvre avec groupes variables, en vert : mise en œuvre avec groupes fixes.

4.2.2 Recherche par croissance d’espace

Une autre méthode approchée consiste en la recherche d’un sous-espace par croissance d’espace. Son principe, extrêmement simple, est le suivant :

- Le premier espace courant \mathcal{E}^0 est égal à l’espace initial des données,
- pour $k = 0$ à $K - 1$, K dimension finale :
 - la recherche du k -ième vecteur e_k de la base de projection est effectuée dans l’espace courant \mathcal{E}^k ,
 - une fois ce vecteur trouvé, le nouvel espace courant \mathcal{E}^{k+1} est réduit à l’orthogonal du vecteur e_k dans l’espace courant précédent \mathcal{E}^k :

$$\mathcal{E}^{k+1} = (e_k)_{\mathcal{E}^k}^\perp \quad (4.5)$$

La recherche du vecteur e_k est effectuée les autres $\{e_i\}_{i=1\dots k-1}$ étant connus, c’est-à-dire que ce vecteur est tel que :

$$e_k = \arg \max_{x \in \mathcal{E}^k} I(e_1, \dots, e_{k-1}, x) \quad (4.6)$$

où $I(B)$ est l’indice de projection de l’espace généré par la base B .

Cet algorithme fait donc grossir l'espace en ajoutant à chaque itération le « meilleur » vecteur dans l'orthogonal du sous-espace que l'on a à l'étape courante.

La solution trouvée par cet algorithme est une solution approchée. Cependant, cette solution est de meilleure qualité que celle trouvée par l'algorithme précédent, car les contraintes sont bien moindres. La contrepartie est que l'algorithme est plus lent, car même si la dimension de l'espace de recherche se réduit de 1 à chaque étape, la dimension initiale est en général relativement grande (224 pour les images AVIRIS).

4.2.3 Comparaison des deux algorithmes

Les deux algorithmes de poursuite de projection présentés précédemment sont des algorithmes approchés. Cependant leurs atouts et leurs faiblesses sont bien différents.

L'algorithme de poursuite de projection séquentielle possède le désavantage sérieux de proposer un découpage figé et arbitraire du spectre en groupes de bandes adjacentes. Mais, ce découpage permet à l'algorithme d'être plus efficace en terme de rapidité. En outre, ce découpage n'est pas absurde dans la mesure où il est plus logique de voir les bandes proches présenter de la redondance que les bandes lointaines.

L'algorithme par croissance d'espace donne lui aussi une solution approchée, mais moins contrainte. En revanche, comme on l'a souligné ci-dessus, les recherches se font dans des espaces à très grande dimension. Ceci peut être gênant pour deux raisons. La première est que cela rend la recherche très lente, la deuxième est que les algorithmes d'optimisation numériques utilisés ne garantissent pas de trouver un optimum global de l'indice de projection à chaque itération, et que la possibilité de trouver un minimum local dans des espaces de grande dimension est accrue.

La figure 4.3 montre les évolutions comparées des valeurs de l'indice de projection dans l'espace déterminé. L'image sur laquelle on a fait le test est la restriction à 40 bandes du spectre visible d'une image AVIRIS. L'indice choisi pour cette expérience est celui utilisé par Jimenez et Landgrebe dans [52]. Ce tableau illustre bien la première des remarques précédentes. En effet, on remarque que l'algorithme par croissance d'espace permet d'obtenir de meilleures valeurs optimales de l'indice de projection, car l'ensemble des espaces parcouru dans la poursuite de projection séquentielle, du fait du découpage du spectre, est plus réduit. En revanche, la différence entre les deux algorithmes est assez faible (la différence est en moyenne de 13% de la plus grande distance). On remarque également que l'indice optimal atteint par l'algorithme séquentiel, à la différence de l'autre, ne croît pas strictement avec le nombre de bandes. Cela vient du fait qu'avec l'ajout de bandes, on change parfois radicalement le découpage du spectre, ce qui peut être préjudiciable.

La figure 4.4 illustre bien la remarque faite ci-dessus sur les temps de calcul respectifs de

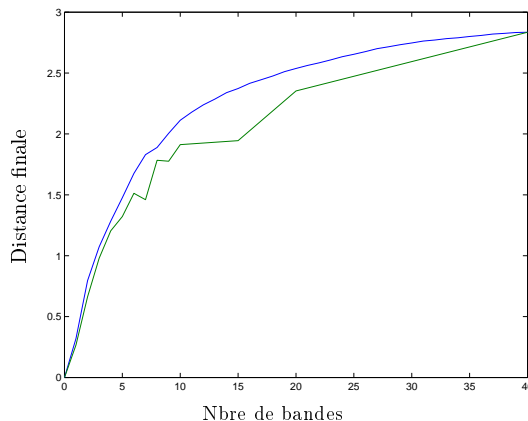


FIG. 4.3 – *Evolution de la distance optimale trouvée par les deux algorithmes de poursuite, avec le même indice de projection. En bleu, la courbe correspondant à l'algorithme par croissance d'espace, et en vert, l'algorithme de poursuite de projection séquentielle.*

ces deux algorithmes. En effet, l'algorithme par croissance d'espace est beaucoup plus lent que l'autre car il fait sa recherche dans l'espace entier (privé de l'espace engendré par les vecteurs déjà trouvés). Dans ce graphique, on voit également l'effet dû au changement du nombre de vecteurs dans chaque groupe, car la courbe verte n'est pas croissante partout. D'autre part, la courbe s'arrête à un espace final de dimension vingt, car au-delà de ce nombre, le découpage en groupes de bandes fait que certains groupes ne comportent qu'une bande (on est au delà de la moitié du nombre de bandes total), ce qui fausse la durée du calcul.

Le problème de la détermination d'un bon indice de projection reste ouvert. On discute de ce point au paragraphe suivant.

4.3 Choix de l'indice de projection : modélisation gaussienne non contextuelle

A l'origine [30, 45], la poursuite de projection a été introduite non pas pour extraire des caractéristiques propices à une bonne classification, mais des caractéristiques permettant d'exhiber une information contenue dans les données de grande dimension. Cette information était, pour les besoins de l'application, une déviation des données par rapport à la gaussiannité. L'indice de projection était donc une mesure de la gaussiannité des données. Il fallait alors extraire les bandes où cette mesure était la plus faible.

Dans notre application, on ne cherche pas à bien décrire une information singulière

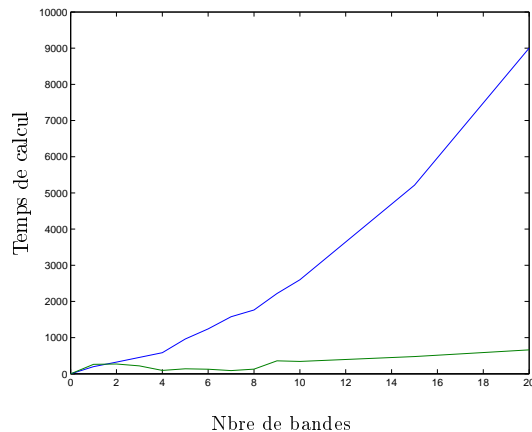


FIG. 4.4 – Evolution du temps de calcul pour les deux algorithmes de poursuite de projection. En bleu, la courbe correspondant à l’algorithme par croissance d’espace, et en vert, l’algorithme de poursuite de projection séquentielle.

contenue dans le signal, mais à opérer une bonne classification.

Jimenez et Landgrebe ont proposé dans [52] d’utiliser une distance entre les classes prédéfinies comme indice de projection. Ce choix est détaillé ci-dessous.

Mise en relation de la règle de classification et de l’indice de projection

Si l’on considère la classification en deux classes par maximum de vraisemblance, la règle de décision qui permet d’assigner une étiquette à un pixel est bien connue :

$$\lambda_s = \arg \max_{l \in \{0,1\}} P_l(x_s) \quad (4.7)$$

où $P_l(x_s)$ est la probabilité de x_s conditionnellement à la classe l .

Dans le cas de la classification en deux classes équiprobables, on peut écrire la probabilité d’erreur de ce classifieur comme la somme sur tout l’espace de la probabilité minimale entre les deux :

$$P_e = \int \min(P_0(x), P_1(x)) dx \quad (4.8)$$

Cette probabilité d’erreur peut être majorée par une famille de bornes, appelées bornes de Chernoff, dont un cas particulier est la borne de Bhattacharyya. La démonstration est rappelée dans [31]. Elle se fonde sur le fait que le min dans l’intégrale est majoré :

$$\min(P_0(x), P_1(x)) \leq (P_0(x))^s (P_1(x))^{1-s} \quad (4.9)$$

avec $s \in [0,1]$.

La probabilité d'erreur est donc majorée par la borne de Chernoff [16], dont l'expression est la suivante :

$$P_e \leq \int (P_0(x))^s (P_1(x))^{1-s} dx \quad (4.10)$$

Il est possible de chercher le paramètre s qui minimise cette borne, mais une borne moins compliquée peut être trouvée pour un s particulier. Pour $s = \frac{1}{2}$, on parle de borne de Bhattacharyya [8]. Alors, on a :

$$P_e \leq \int \sqrt{P_0(x)P_1(x)} dx \quad (4.11)$$

Cette borne peut aussi être exprimée comme l'exponentielle d'une distance :

$$\int \sqrt{P_0(x)P_1(x)} dx = e^{-d_B} \quad (4.12)$$

Cette distance d_B est appelée distance de Bhattacharyya, et est reliée à un majorant de la probabilité d'erreur d'un classificateur par maximum de vraisemblance.

Pour des distributions normales, cette distance s'écrit de la façon suivante [53] :

$$d_B = \frac{1}{8} \|\mu_0 - \mu_1\|_{\Sigma_{01}} + \frac{1}{2} \ln \left(\frac{|\Sigma_{01}|}{\sqrt{|\Sigma_0||\Sigma_1|}} \right) \quad (4.13)$$

où μ_0 et μ_1 sont les moyennes des deux distributions, Σ_0 et Σ_1 , les deux matrices de covariance, et $\Sigma_{01} = \frac{\Sigma_0 + \Sigma_1}{2}$.

Ceci a amené Jimenez et Landgrebe à proposer dans [52] un indice de projection fondé sur les distances de Bhattacharyya entre les distributions des échantillons d'apprentissage des différentes paires de classes.

4.4 Choix de l'indice de projection : modélisation markovienne gaussienne

4.4.1 Mise en relation de la règle de classification et de l'indice de projection

Soit un problème de classification à deux classes, C_0 et C_1 . La règle de décision correspondant au maximum de vraisemblance (MV) est dans le cas de la modélisation présentée :

$$\lambda_s = \arg \max_{l \in \{0,1\}} P_l(x_s | \{m_s^k\}) \quad (4.14)$$

où les $\{m_s^k\}$ sont les moyennes des voisins de type k du pixel s .

Dans la modélisation que l'on propose, les probabilités conditionnelles sont gaussiennes. Or, pour ces distributions, la distance de Bhattacharyya a une expression exacte simple. On peut donc définir une distance de Bhattacharyya conditionnellement aux $\{m_s^k\}$. Pour simplifier l'écriture, on considère un modèle à un seul type de voisinage, par exemple un voisinage 4-connexe isotrope. On a donc :

$$x_s \sim \mathcal{N}(a'_i m_s + b_i, \Sigma_i) \quad (4.15)$$

si x_s appartient à la classe C_i , m_s étant la moyenne des 4 voisins de s .

L'apparition du vecteur $b_i = (1 - a'_i)\mu_i$, où μ_i est la moyenne de la texture i , vient du fait que l'on ne peut plus considérer les données comme centrées, car cela reviendrait à se restreindre au cas très particulier où toutes les textures ont la même moyenne. On revient donc à la formule pour les données non centrées. Cela ne change en rien le processus d'estimation. En effet, on peut calculer la moyenne de la classe considérée et ainsi se ramener au cas de données centrées sur celle-ci.

La distance de Bhattacharyya conditionnelle entre les classes C_0 et C_1 s'écrit donc :

$$d_B(m_s) = \frac{1}{8} \|\mu_0(m_s) - \mu_1(m_s)\|_{\Sigma_{01}} + \frac{1}{2} \ln \left(\frac{|\Sigma_{01}|}{\sqrt{|\Sigma_0||\Sigma_1|}} \right) \quad (4.16)$$

avec $\mu_i(m_s) = a'_i m_s + b_i$ et $\Sigma_{01} = \frac{\Sigma_0 + \Sigma_1}{2}$.

Etant donné la règle de décision (4.14), la probabilité d'erreur connaissant m_s est liée à $d_B(m_s)$. En maximisant $d_B(m_s)$, on minimise cette probabilité d'erreur. On peut construire deux indices de projection reliés à cette distance de Bhattacharyya. La mise en œuvre de la poursuite de projection consistera à rechercher la projection dans laquelle l'un ou l'autre de ces indices sera maximisé.

Indice 1

$$I_1 = E\{b(m_s)\} = \int d_B(m_s) f(m_s) dm_s \quad (4.17)$$

Indice 2

$$I_2 = \min_{m_s} d_B(m_s) \quad (4.18)$$

Le problème de l'indice I_1 est que l'on n'a accès à la loi $f(x)$ qu'à travers l'histogramme de l'image, ce qui peut augmenter le temps nécessaire à l'intégration. Par ailleurs, l'indice I_2 a l'avantage de rester relié à une borne supérieure de la probabilité d'erreur ce qui lui donne un avantage en plus du temps de calcul. Mais, l'indice I_1 , qui moyenne les contributions,

évitte de prendre en compte les contributions de m_s peu représentatives. Le choix entre les deux indices n'est donc pas trivial. Toutefois, de façon empirique, on remarque que les résultats de classification obtenus lorsque l'on utilise l'indice I_1 pour la réduction d'espace sont en général meilleurs que lorsque l'on utilise l'indice I_2 . Les résultats présentés par la suite sont donc obtenus avec le premier indice.

4.4.2 Problème à plusieurs classes

Le cas où le nombre de classes est supérieur à 2 est beaucoup plus vraisemblable dans le cadre de la télédétection. Or, la distance de Bhattacharyya ne peut se calculer qu'entre 2 distributions. Une méthode pour construire l'indice correspondant à un problème de ce type consiste à prendre la plus petite des distances entre classes :

$$I = \min_{i,j} I_{i,j} \quad (4.19)$$

$I_{i,j}$ représentant un des deux indices proposés au paragraphe 4.4.1, correspondant au problème de classification entre i et j .

On peut même aller plus loin, puisque si l'on est intéressé par la classification des zones urbaines, il suffit de ne chercher ce minimum que dans l'espace des couples de classes dont l'une d'elles au moins est une classe urbaine. C'est-à-dire que les distances entre classes non urbaines ne nous intéressent pas, car on ne cherche pas à les classifier. Par conséquent, il n'est pas nécessaire de faire en sorte que la distance entre ces classes soit grande.

4.4.3 Calcul de l'indice de projection

Etant donné un sous-espace $E' \subset E$, le calcul de l'indice n'est pas immédiat. Il faut, en effet, procéder en deux étapes. La première consiste à ré-estimer les paramètres du modèle, car un des intérêts de la méthode est d'effectuer les calculs dans le sous-espace. La deuxième est la projection des réponses et des matrices dans le sous-espace et le calcul de l'indice.

L'estimation des $\{a'_k\}$ se fait par l'équation (2.45), qui, si l'on approxime la matrice de covariance conditionnelle par la matrice identité, peut se récrire comme suit :

$$\begin{pmatrix} \text{tr} \langle m_s^0 (m_s^0)^t \rangle & \dots & \text{tr} \langle m_s^K (m_s^0)^t \rangle \\ \vdots & & \vdots \\ \text{tr} \langle m_s^0 (m_s^K)^t \rangle & \dots & \text{tr} \langle m_s^K (m_s^K)^t \rangle \end{pmatrix} \begin{pmatrix} a'_0 \\ \vdots \\ a'_K \end{pmatrix} = \begin{pmatrix} \text{tr} \langle m_s^0 x_s^t \rangle \\ \vdots \\ \text{tr} \langle m_s^K x_s^t \rangle \end{pmatrix} \quad (4.20)$$

Soit P une matrice de projection, l'estimation des $\{a'_k\}$ dans l'espace image du projecteur correspond à la résolution de l'équation suivante :

$$\begin{pmatrix} \text{tr}(P\Sigma_{00}P^t) & \dots & \text{tr}(P\Sigma_{0K}P^t) \\ \vdots & & \vdots \\ \text{tr}(P\Sigma_{0K}P^t) & \dots & \text{tr}(P\Sigma_{KK}P^t) \end{pmatrix} \begin{pmatrix} a'_0 \\ \vdots \\ a'_K \end{pmatrix} = \begin{pmatrix} \text{tr}(P\Sigma_{x,0}P^t) \\ \vdots \\ \text{tr}(P\Sigma_{x,K}P^t) \end{pmatrix} \quad (4.21)$$

avec $\Sigma_{ij} = \langle m_s^i (m_s^j)^t \rangle$ et $\Sigma_{x,i} = \langle m_s^i x_s^t \rangle$.

A la suite du calcul des $\{a'_k\}$, on peut ré-estimer Σ . On a alors calculé tous les paramètres du modèle nécessaires au calcul de la distance de Bhattacharyya.

La deuxième étape du calcul de l'indice consiste à calculer les valeurs de la distance de Bhattacharyya, connaissant les paramètres, correspondant aux différentes valeurs de m_s , pour évaluer l'indice de projection.

4.4.4 Indices de projection globaux

L'indice de projection I_1 (qui donne de meilleurs résultats que I_2) est fondé sur une somme de contributions locales à la distance de Bhattacharyya. On peut également exprimer une distance globale entre les différents champs, et chercher à maximiser cette distance entre les champs.

Or il se trouve que les champs ont des lois gaussiennes. La loi globale d'un champ correspondant à une classe i est la suivante :

$$P_i(x) = \frac{1}{Z(S_i)} \exp\left(-\frac{1}{2}(x - m_i)^t (I_{N_h} \otimes I_{N_v} \otimes \Sigma_i^{-1}) A_i(x - m_i)\right) \quad (4.22)$$

où m_i est un vecteur colonne formé de N fois la moyenne μ_i du champ, N étant le nombre de points du champ. La distance de Bhattacharyya entre les lois de deux champs i et j peut donc facilement être exprimée comme :

$$d_B = \frac{1}{8} \|m_i - m_j\|_{S_{ij}} + \frac{1}{2} \ln\left(\frac{|S_{ij}|}{\sqrt{|S_i||S_j|}}\right) \quad (4.23)$$

Dans cette expression, $S_i = A_i^{-1} (I_{N_h} \otimes I_{N_v} \otimes \Sigma_i)$.

Le problème dans cette expression vient du premier terme. En effet, pour calculer $\|m_i - m_j\|_{S_{ij}}$, il est nécessaire de connaître S_{ij}^{-1} .

Le calcul de S_i^{-1} et S_j^{-1} est possible, en utilisant la transformation de Fourier, de même que pour calculer leur déterminant. Malheureusement, l'inverse de S_{ij} , qui est la somme des matrices S_i^{-1} et S_j^{-1} , ne peut pas se calculer aisément, du fait que cette matrice ne garde pas la même structure comportant le produit de Kronecker et la matrice convolution.

Une alternative est le choix d'un autre type de distance, par exemple la divergence de Kullback-Leibler, définie comme suit, pour deux classes i et j ayant comme distribution respectives f_i et f_j :

$$H_{kb}(i||j) = \int f_i(x) \ln \left(\frac{f_i(x)}{f_j(x)} \right) dx \quad (4.24)$$

Cette divergence constitue une mesure de différence entre les deux distributions de i et j . Il est à noter que cette divergence n'est pas symétrique, donc que $H_{kb}(i||j) \neq H_{kb}(j||i)$. Soit :

$$D_{kb}(i,j) = H_{kb}(i||j) + H_{kb}(j||i) \quad (4.25)$$

Cette valeur, elle, est symétrique.

En présence de deux distributions gaussiennes, on calcule cette divergence [4] par :

$$2D_{kb}(i,j) = (\mu_i - \mu_j)^t (S_i^{-1} + S_j^{-1}) (\mu_i - \mu_j) + \text{tr} (S_i^{-1}S_j + S_j^{-1}S_i - 2I) \quad (4.26)$$

Le calcul de cette "distance" ne nécessite pas le calcul de l'inverse d'une somme des matrices Σ_i et Σ_j . Or on sait que la modélisation implique pour chaque classe :

$$S = A^{-1} (I_{N_h N_v} \otimes \Sigma) \quad (4.27)$$

$$S^{-1} = (I_{N_h N_v} \otimes \Sigma^{-1}) A \quad (4.28)$$

Il est donc possible de la calculer plus rapidement que la distance de Bhattacharyya, ce qui est important dans un schéma itératif assez lourd comme les algorithmes de poursuite de projection.

4.5 Conclusion

Ce chapitre a été consacré à la présentation du cadre général d'analyse que constitue la poursuite de projection pour la recherche de sous-espaces ayant de bonnes propriétés. Dans un premier temps, les principes généraux de la poursuite de projection ont été énoncés, et la notion de l'indice de projection à optimiser a été introduite. On a détaillé ensuite deux algorithmes approchés permettant de mener à bien la poursuite de projection, la poursuite de projection séquentielle et la poursuite de projection par croissance d'espace, ainsi qu'une variante du premier avec groupes de bandes variables, que l'on propose.

Le choix de la distance de Bhattacharyya comme indice de projection dans le cadre de la classification, qui a déjà été utilisée dans un cadre non-contextuel, a été justifié, et on a explicité plus précisément cet indice dans le cadre de la modélisation proposée au chapitre précédent.

Chapitre 5

Application

Dans ce chapitre, on met en application le modèle de texture par champ de Markov gaussien développé et les méthodes de réduction de dimension. Ceux-ci sont testés à travers deux algorithmes globaux de classification d'images hyperspectrales que l'on propose. Ces deux algorithmes découlent de deux approches différentes pour la classification : l'une par la mise en place d'une règle de décision fondée sur la vraisemblance des pixels supposant la texture générée par le modèle, l'autre par l'ajout aux images de radiométrie de nouvelles bandes représentant les paramètres de texture, suivi d'une classification non-contextuelle. Les deux algorithmes passent chacun par le biais de la poursuite de projection pour la réduction de dimension.

Le deuxième algorithme s'avère plus efficace en terme de taux de bonne classification, en comparaison avec une méthode totalement non-contextuelle et avec une méthode de classification de texture fondée sur l'analyse des matrices de cooccurrence.

5.1 Classification de texture

In fine, le but de cette analyse de texture est la classification, en particulier celle de zones urbaines qui sont texturées (par opposition aux zones rurales, qui sont plus homogènes à la résolution considérée).

Pour ce faire, on propose deux types d'algorithmes fondés sur les modèles et méthodes décrits aux chapitres précédents. On détaille ces algorithmes dans les paragraphes suivants. Ceux-ci sont comparés à des algorithmes utilisant d'autres outils d'analyse de texture, et à des algorithmes de classification non-contextuels.

Tous les algorithmes que l'on présente sont constitués d'une première phase durant laquelle les paramètres des différentes classes sont estimés grâce aux échantillons d'apprentissage. Ensuite, la classification est faite selon différentes méthodes, mais toujours en comparant d'une manière ou d'une autre des mesures faites en chaque pixel (ou dans un voisinage local) avec celles de référence données par les échantillons d'apprentissage.

5.2 Classification : Maximum de vraisemblance

La première famille d'algorithmes [70, 72] de classification supervisée de texture s'appuie sur le maximum de vraisemblance du modèle markovien gaussien. Le critère de décision est directement calculé selon la concordance entre les données et les modèles estimés pour chaque classe.

5.2.1 Méthode locale

Pour mettre au point cette méthode locale, on revient à l'équation (4.14), qui nous a aidé à justifier l'indice de projection pour la poursuite de projection dans le cadre du modèle markovien gaussien. En effet, il s'agit de la règle de décision correspondant au maximum de vraisemblance local du modèle proposé.

Pour un problème de classification à L classes C_l , $l = 0 \dots L - 1$, la règle de décision correspondant au maximum de vraisemblance (MV) est, dans le cas de la modélisation présentée au chapitre 2, d'affecter l'étiquette $\lambda_s \in \{0 \dots L - 1\}$ au pixel s de la façon suivante :

$$\lambda_s = \arg \max_{l \in \{0 \dots L - 1\}} P_l(x_s | \{x_t\}, t \in V_s) \quad (5.1)$$

où $P_l(x_s | \{x_t\}, t \in V_s)$ est la loi conditionnelle estimée à partir des échantillons d'apprentissage de la classe l .

Cette loi est, comme on l'a vu précédemment, une loi gaussienne centrée sur la somme d'une combinaison linéaire des voisins et d'une fonction de la moyenne.

$$x_s \sim \mathcal{N} \left(\sum_{t \in V_s} a_t x_t + \nu_i, \Sigma_i \right) \quad (5.2)$$

où $\nu_i = (1 - \sum_{t \in V_s} a_t) \mu_i$ se déduit de l'équation 2.5 au chapitre 2.

On choisit donc, pour le pixel s , l'étiquette λ_s tel que la probabilité conditionnelle soit la plus grande. Cette méthode locale est très sensible aux anomalies des champs peu réguliers que l'on rencontre dans les images hyperspectrales réelles, et donne des résultats de classification très irréguliers.

Pour pallier ce problème, on opère une régularisation de la classification par l'ajout d'une loi *a priori* sur les étiquettes, sous forme d'un modèle de Potts [83], modèle fréquemment utilisé pour la régularisation des champs d'étiquettes en classification. Ce modèle *a priori* pénalise les pixels isolés grâce à des potentiels de la forme suivante :

$$V_{(l_s, l_t)}^{\text{Potts}} = -\beta \delta_{l_s = l_t} \quad (5.3)$$

où l_s et l_t sont les étiquettes des pixels voisins s et t . Les paires de pixels voisins d'étiquettes différentes sont pénalisés par β .

La classification régularisée est obtenue grâce à l'algorithme du recuit simulé [32, 33].

En outre, le terme ν_i prenant en compte la moyenne de la texture présente un problème dans le sens où il disparaît progressivement quand la somme des coefficients d'interaction tend vers 1, la limite du domaine de définition. Ceci a pour conséquence de faire disparaître des équations un terme qui peut avoir un rôle important pour discriminer certaines textures, ou pour atténuer les conséquences d'un voisin aberrant ou d'une anomalie dans la texture. En effet, c'est dans ces cas-là que la moyenne pourrait avoir une importance.

5.2.2 Méthode par blocs

La méthode précédente produit des classifications très irrégulières car elle s'appuie sur des mesures très locales, fortement sujettes au bruit de modèle et aux irrégularités de champs dans les images réelles.

C'est pour cela que l'on préfère utiliser un algorithme de classification similaire au précédent, mais plus robuste vis-à-vis de ces irrégularités, et qui par conséquent, en s'appuyant sur une prise en compte plus étendue du voisinage, résulte en des classifications plus régulières.

Cette méthode est très similaire à la précédente, puisque comme celle-ci, on va tester l'adéquation des données avec les lois calculées pour les différentes classes, et retenir la plus vraisemblable. La différence est que cette fois-ci, on ne teste pas seulement en chaque pixel la vraisemblance de chaque classe se fondant uniquement sur la loi conditionnelle du pixel connaissant ses voisins, mais l'on va chercher un voisinage plus étendu. Soit \tilde{X}_s une fenêtre carrée, centrée sur X_s , de rayon r . Connaissant la loi globale d'un champ markovien gaussien, on peut déterminer les probabilités $P_l(\tilde{X}_s = \tilde{x}_s)$ conditionnelles aux classes. Pour ces lois, on peut considérer un raccordement torique de la fenêtre d'analyse, ou les conditions réelles aux bords.

La classification se fait tout de même pixel par pixel, et la règle de décision pour cet algorithme est d'assigner au pixel s l'étiquette λ_s telle que :

$$\lambda_s = \arg \max_{l \in \{0 \dots L-1\}} P_l(\tilde{x}_s) \quad (5.4)$$

On se sert donc du contexte local d'un pixel pour en déterminer la classe supposée. Ceci permet d'exploiter le fait que le concept de champ de Markov, même s'il s'appuie explicitement uniquement sur le voisinage immédiat, sert à exprimer des corrélations qui vont bien plus loin que ce voisinage dans un contexte plus étendu.

La taille de la fenêtre résulte d'un compromis entre d'une part la précision spatiale de la classification (qui réclame une taille de fenêtre faible), et d'autre part la régularité de la solution trouvée (qui augmente avec la taille de la fenêtre). On obtient des solutions assez régulières avec des fenêtres d'environ 7 ou 9 pixels de côté.

La décision de classification étant prise en considérant un voisinage plus étendu du pixel considéré, celle-ci est de ce fait plus régulière. En effet, la classification de deux pixels voisins sera effectuée en se fondant sur deux ensembles de pixels ayant une grande intersection. Il n'y a donc pas besoin, dans ce cas, de passer par un modèle *a priori* de Potts pour la régularisation.

5.2.3 Réduction de dimension

C'est pour ces méthodes de classification que l'adaptation de l'indice de projection tel que cela a été proposé au chapitre 4 (paragraphe 4.4), est intéressante, car la distance que l'on maximise dans ces cas-là est une fonction d'une borne supérieure du taux d'erreur théorique de classification par le critère du maximum de vraisemblance.

On réalise donc, en début de traitement pour ces deux algorithmes, une réduction de dimension par l'algorithme de poursuite de projection avec l'indice de projection adapté au paragraphe 4.4. Le nombre de bandes sera variable dans les tests de classification que nous réaliserons.

5.3 Classification : Ajout de bandes de textures

5.3.1 Description de l'algorithme

En matière de classification, une méthode d'analyse de texture envisageable consiste à faire l'hypothèse d'un modèle probabiliste, et de tester la vraisemblance de ce modèle sur les données. C'est la méthode que l'on propose au paragraphe précédent. Cette méthode apporte une amélioration par rapport à des méthodes non contextuelles dans le cas où la moyenne de la texture est peu significative (peu discriminante). En effet, celle-ci rentre parfois (selon la texture considérée) pour une faible part dans le calcul de la vraisemblance du modèle, notamment quand les interactions spatiales sont fortes.

Dans un cas plus général, on souhaite avoir une méthode où les influences de la radiométrie et de la texture sont équivalentes. Pour ce faire, il paraît plus judicieux de passer par des méthodes où l'on extrait des images de texture que l'on ajoute aux images initiales de radiométrie (cf. figure 5.1).

L'algorithme de classification proposé [71] est divisé en deux phases. La première phase est un pré-traitement qui consiste en une réduction de la dimension des données d'une part, et en une extraction des paramètres de texture d'autre part. La deuxième phase est la classification elle-même.

5.3.2 Paramètres envisageables

Les outils d'analyse permettant d'obtenir les images de texture peuvent être, par exemple, les ondelettes (ou les paquets d'ondelettes), les filtres de Gabor ou les matrices de cooccurrence. Ces outils fournissent les grandeurs qui forment les images de texture. Dans ce rapport, on a supposé des textures générées par des MGMRF, il paraît donc logique d'utiliser les paramètres associés à ce champ comme caractéristiques des textures analysées. Il

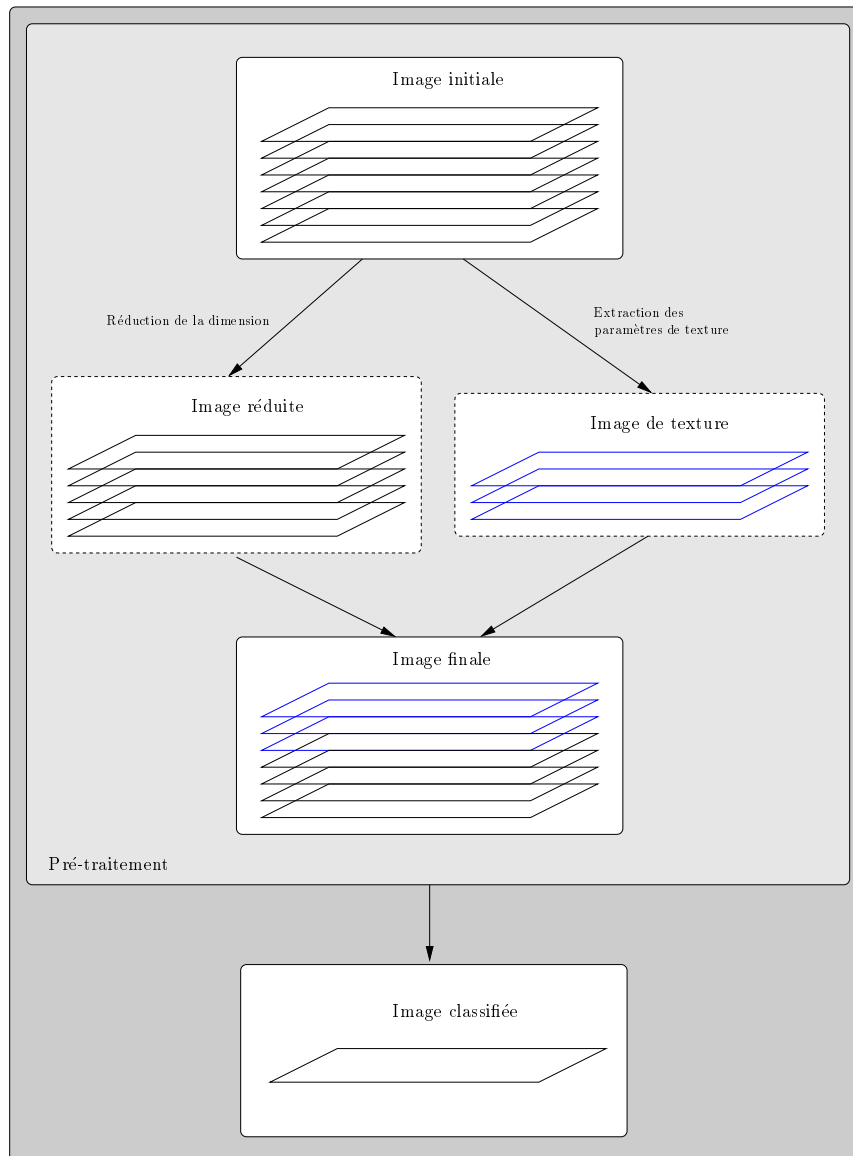


FIG. 5.1 – *Algorithme de classification avec ajout de bandes.*

s'agit donc de paramètres de deux types : la matrice de covariance conditionnelle Σ , et des paramètres d'interaction qui, dans le cas particulier du MGMRF que l'on a décrit, sont réduits à un vecteur, le vecteur a , où sont regroupés tous les paramètres d'interaction.

5.3.3 Calcul des paramètres

Dans un premier temps, il est nécessaire d'obtenir les images de paramètres de texture.

On veut avoir en chaque point une estimation des paramètres de texture. Or, la texture n'est pas une notion pixélique, mais se situe au niveau d'un voisinage local. On va donc calculer en chaque point les paramètres de texture correspondant à une fenêtre centrée sur ce point. L'estimation se fait à partir de l'un des algorithmes décrits au paragraphe 2.3.7. Dans les applications, nous utilisons l'estimateur MMSE, puisqu'il est beaucoup plus rapide que l'estimateur MV et que le nombre d'estimations à effectuer, égal au nombre de pixels, peut être très grand.

Cette méthode suppose que, dans une fenêtre centrée sur un point, seul un type de texture est représenté. Ceci peut être supposé vrai si la fenêtre est assez petite, mais il est préférable de traiter des tailles de fenêtre suffisamment grandes. La raison en est que si l'on travaille avec des fenêtres trop petites, l'estimation est très peu fiable (d'une part, du fait que l'on peut avoir des configurations très particulières ou peu caractéristiques du champ à certains endroits, et d'autre part du fait du phénomène de Hughes).

Malheureusement, en augmentant la taille de la fenêtre, on augmente la possibilité d'y trouver plusieurs textures. Pour remédier à ce problème, on passe par une euristique destinée à détecter la présence de deux textures dans une même fenêtre. Prenons le cas d'une fenêtre dans laquelle on aurait deux classes représentées, C_0 et C_1 , de moyennes μ_0 et μ_1 et de matrices de covariance Γ_0 et Γ_1 , à raison de proportions n_0 et n_1 , telles que $n_0 + n_1 = 1$.

Le calcul de la matrice de covariance dans cette fenêtre donne :

$$\hat{\Gamma} = \frac{1}{N-1} \sum_s (x_s - \hat{\mu})(x_s - \hat{\mu})^t \quad (5.5)$$

Soit Ω_0 (resp. Ω_1), la zone correspondant à la classe C_0 (resp. C_1).

$$\hat{\Gamma} = \frac{1}{N-1} \left(\sum_{s \in \Omega_0} (x_s - \hat{\mu})(x_s - \hat{\mu})^t + \sum_{s \in \Omega_1} (x_s - \hat{\mu})(x_s - \hat{\mu})^t \right) \quad (5.6)$$

Or, $\hat{\mu} = \hat{\mu}_0 + n_1(\hat{\mu}_1 - \hat{\mu}_0) = \hat{\mu}_1 + n_0(\hat{\mu}_0 - \hat{\mu}_1)$. Donc :

$$\begin{aligned}
(N-1)\hat{\Gamma} &= \sum_{s \in \Omega_0} (x_s - \hat{\mu}_0 + n_1(\hat{\mu}_1 - \hat{\mu}_0))(x_s - \hat{\mu}_0 + n_1(\hat{\mu}_1 - \hat{\mu}_0))^t \\
&+ \sum_{s \in \Omega_1} (x_s - \hat{\mu}_1 + n_0(\hat{\mu}_0 - \hat{\mu}_1))(x_s - \hat{\mu}_1 + n_0(\hat{\mu}_0 - \hat{\mu}_1))^t
\end{aligned} \tag{5.7}$$

Soit :

$$\begin{aligned}
(N-1)\hat{\Gamma} &= \sum_{s \in \Omega_0} (x_s - \hat{\mu}_0)(x_s - \hat{\mu}_0)^t \\
&+ \sum_{s \in \Omega_1} (x_s - \hat{\mu}_1)(x_s - \hat{\mu}_1)^t \\
&+ n_1 n_0 (n_1 - n_0) (\hat{\mu}_1 - \hat{\mu}_0) (\hat{\mu}_1 - \hat{\mu}_0)^t
\end{aligned} \tag{5.8}$$

Si l'on note $M_{12} = (\hat{\mu}_1 - \hat{\mu}_0)(\hat{\mu}_1 - \hat{\mu}_0)^t$, on obtient :

$$\hat{\Gamma} = n_0 \hat{\Gamma}_0 + n_1 \hat{\Gamma}_1 + n_1 n_0 (n_1 - n_0) M_{12} \tag{5.9}$$

Lorsqu'une fenêtre de calcul traverse une transition entre deux textures, on passe progressivement de $n_0 = 1$ à $n_1 = 1$. Dans le premier cas, la matrice de covariance empirique est celle d'un ensemble d'échantillons de la texture C_0 , dans l'autre de C_1 .

Mais la transition n'est pas forcément douce. En effet, celle-ci dépend de μ_0 et μ_1 . Si les moyennes sont éloignées, la transition sera fortement perturbée, alors que si celles-ci sont proches, on a quasiment un passage direct, par le plus court chemin entre Γ_0 et Γ_1 .

Ceci est valable pour les matrices de covariance, et le comportement des matrices de corrélation est similaire. Donc pour la matrice de covariance conditionnelle, qui est égale à la matrice de covariance pondérée par un coefficient dépendant du vecteur a , on a également une trajectoire non directe quand les moyennes sont distantes, du fait du cumul de deux perturbations : l'une sur Γ , et l'autre sur a .

L'heuristique (cf. fig 5.2) proposée ici est fondée sur le fait que le problème de la présence de deux textures dans une même fenêtre apparaît quand les moyennes des textures sont distantes. Le principe est simple, il consiste à vérifier avant le calcul des paramètres de texture que l'on n'est pas en présence de deux zones de moyennes éloignées, en déterminant d'abord deux nuages de points au moyen de l'algorithme *k-means*, puis de tester la distance entre ces deux nuages. Si la distance entre les centres de ces nuages est supérieur à un seuil dépendant de la variance des nuages, alors on suppose que l'on est en présence de deux zones distinctes, et le calcul des paramètres de texture se fait sur le nuage ayant le plus grand nombre de points. On ne prend en compte que les cas où l'on a deux textures, au plus, dans une fenêtre.

La figure 5.3 illustre le besoin de passer par cette heuristique pour éviter les problèmes de transition entre textures. Il est net que les frontières entre textures sont alors beaucoup

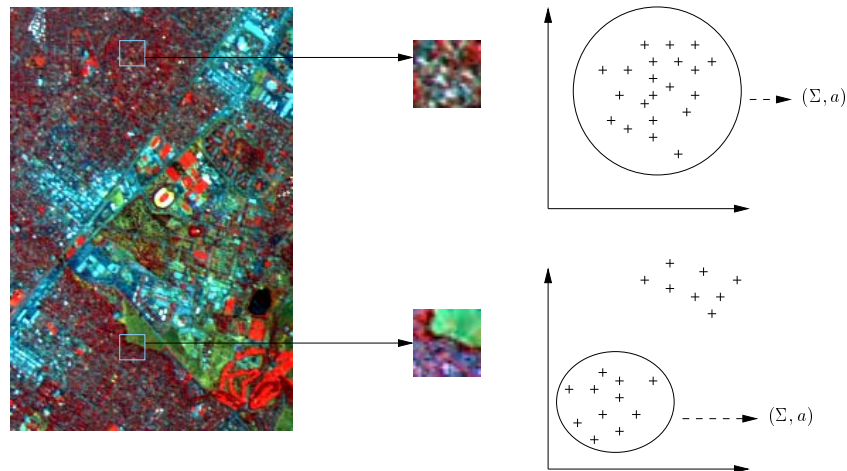


FIG. 5.2 – *Illustration de l'heuristique : dans le cas du haut, une seule texture est présente, on calcule les paramètres sur tous les échantillons, dans le cas du bas, il y a deux classes, on calcule les paramètres sur la classe la plus présente.*

plus franches. En revanche, on n'élimine pas les problèmes quand plus de deux textures sont présentes dans la fenêtre (à la jonction des 4 textures, par exemple, au centre de l'image). Il serait immédiat d'adapter la méthode à trois textures ou plus, mais la complexité algorithmique, déjà fortement augmentée avec l'heuristique à deux textures, deviendrait alors prohibitive. On se restreint donc au cas le plus simple.

Il reste à opérer le choix des paramètres pertinents pour la classification de texture : on garde la matrice Σ et on rejette le vecteur a . Pour justifier ce choix, la figure 5.4 montre le résultat de l'extraction de Σ et a dans le cas d'une dimension finale égale à 1 concernant la texture et d'une modélisation isotrope. Dans ce cas, Σ est réduite à un scalaire σ^2 , de même que a . On se rend bien compte du pouvoir bien plus discriminant de la variance conditionnelle par rapport au paramètre d'interaction. De plus, en raison de la grande variabilité intra-classe de a par rapport à la taille de l'espace dans lequel il évolue, ce paramètre est difficilement exploitable.

5.3.4 Retour sur le calcul de l'indice de projection pour la poursuite de projection

L'attribut de texture que l'on garde est donc la matrice de covariance conditionnelle Σ estimée, c'est-à-dire que l'on a $\frac{N(N-1)}{2}$ paramètres distincts. Comme pour les bandes de radiométrie, il est nécessaire d'effectuer une réduction de dimension pour les paramètres de texture. Cette réduction est effectuée par un algorithme de poursuite de projection, dont

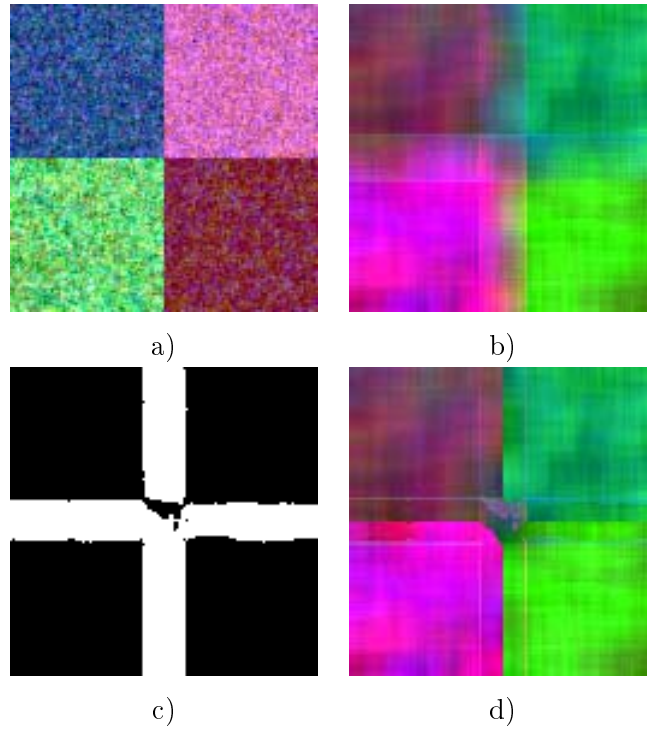


FIG. 5.3 – a) Collage de textures à 3 bandes spectrales simulées par MGMRF, b) image en fausses couleurs des éléments diagonaux de $\hat{\Sigma}$ sans euristique, c) transitions détectées par l'euristique, d) éléments diagonaux de $\hat{\Sigma}$ avec l'euristique.

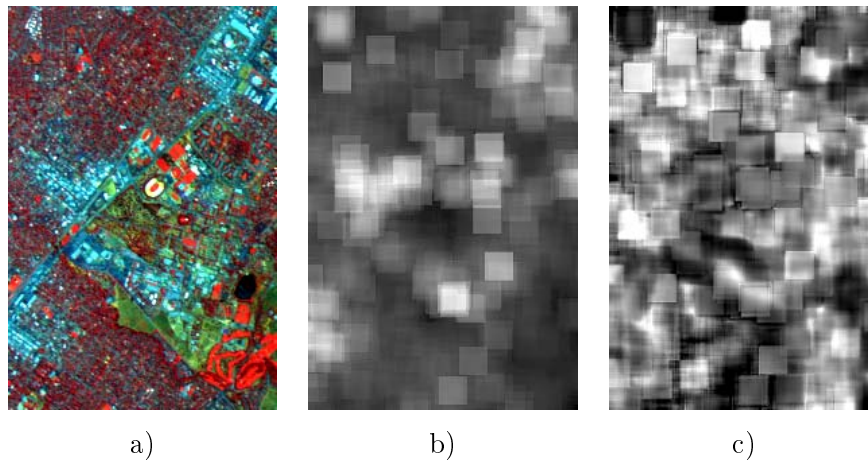


FIG. 5.4 – a) Extrait de 3 bandes visibles de l'image AVIRIS de Moffett Field, b) variance conditionnelle σ^2 extraite d'une bande de cette image, c) paramètres d'interaction a extrait d'une bande de cette image.

on décrit dans ce chapitre l'indice de projection.

Lorsque l'on a traité le cas des bandes de radiométrie, l'on a choisi la distance de Bhattacharyya entre les classes prédéfinies comme indice de projection à maximiser, comme proposé dans [52]. Cette distance est facilement calculable dans le cas de données gaussiennes, et l'on va chercher à la calculer pour les paramètres dont on dispose.

Approximation de la loi de probabilité du paramètre de texture

Dans le cas des paramètres de texture que nous considérons, il est clair que la modélisation gaussienne n'est pas adaptée. La figure 5.5 montre des exemples d'histogrammes du paramètre σ^2 calculés sur une bande du spectre visible de 2 zones d'une image AVIRIS, ceux-ci n'ont pas la forme de gaussiennes.

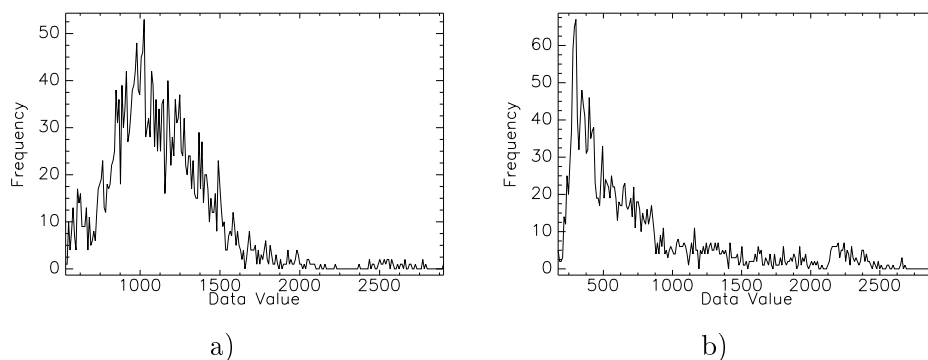


FIG. 5.5 – 2 histogrammes du paramètre σ^2 calculés sur une bande du spectre visible de 2 zones de l'image AVIRIS de Moffett Field.

La matrice $\hat{\Sigma}$ est définie, en supposant que la moyenne de la texture est nulle, par :

$$\hat{\Sigma} = \frac{1}{N} \sum_s u_s u_s^t \quad (5.10)$$

Et l'on note $S_u = N\hat{\Sigma} = \sum_s u_s u_s^t$

Dans le cas où les u_s sont indépendants, la loi de S_u est une loi de Wishart à N degrés de liberté (N étant le nombre de pixels servant à l'estimation) de paramètre Σ (cf. [1]). Cette loi est le pendant multidimensionnel de la loi χ^2 , et sa densité s'écrit de la façon suivante :

$$p(S_u) = \frac{|S_u|^{\frac{1}{2}(N-p-1)} \exp\left(-\frac{1}{2}\text{tr}\Sigma^{-1}S_u\right)}{K |\Sigma|^{\frac{1}{2}N}} \quad (5.11)$$

où p est la dimension des données, et K est une constante de normalisation, dépendant de N et p . Une variable ayant cette loi de probabilité a une moyenne égale au nombre de degrés de liberté que multiplie la matrice Σ .

$$E\{S_{ij}\} = N\Sigma_{ij} \quad (5.12)$$

C'est-à-dire que la moyenne de $\hat{\Sigma}$ est Σ .

En ce qui concerne les moments d'ordre deux, il vient :

$$E\{(S_{ij} - N\Sigma_{ij})(S_{kl} - N\Sigma_{kl})\} = N(\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \quad (5.13)$$

De même, on en déduit pour $\hat{\Sigma}$:

$$E\{(\hat{\Sigma}_{ij} - \Sigma_{ij})(\hat{\Sigma}_{kl} - \Sigma_{kl})\} = \frac{1}{N}(\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \quad (5.14)$$

La distribution de $\hat{\Sigma}$ est de plus en plus "piquée" autour de Σ quand N augmente. Intuitivement, cela semble normal puisque plus on utilise de données pour l'estimation, plus l'estimée est précise.

On va maintenant calculer l'approximation de la loi réelle de $\hat{\Sigma}$, puisqu'en réalité, dans notre cas, les erreurs d'estimation u_s sont corrélées comme mentionné ci-dessus. On s'intéresse ici à la loi de $\hat{\Sigma}$ conditionnellement à a . Cette loi est notée $p_a(\hat{\Sigma})$.

On peut montrer que, dès l'instant où les variables u_s sont corrélées, la densité de S_u n'est pas celle d'une loi de Wishart. Néanmoins, on approche la densité de $\hat{\Sigma}$ par une telle loi pour plusieurs raisons. La première est que ce modèle de distribution est assez proche de ce que l'on remarque empiriquement sur des données hyperspectrales. La deuxième raison est que les deux grandeurs que sont la covariance et la covariance conditionnelle sont obtenues de façon assez analogue.

Pour approcher la loi de $\hat{\Sigma}$, on va rechercher la loi de Wishart dont les deux premiers moments coïncident avec la loi donnée. Calculons les moments d'ordre 1 et 2 de $S_u = \sum_s u_s u_s^t$ et de $\hat{\Sigma} = \frac{1}{N}S_u$. On note $S_x = \sum_s x_s x_s^t$. On a $S_x = AS_u A^t$. Et enfin, soit y la transformée de Fourier de x et $S_y = \sum_s y_s y_s^t$. On a $S_y = S_x$, et :

$$E\{(S_y)_{ij}\} = E\left\{\sum_s y_{s,i} y_{s,j}\right\} \quad (5.15)$$

Or, du fait de la structure circulante par blocs de la matrice de covariance S du MGMRF, la transformée de Fourier de ce champ a ses sites non corrélés (cf. annexe A). Donc :

$$E\{(S_y)_{ij}\} = \sum_s E\{y_{s,i} y_{s,j}\} \quad (5.16)$$

$$E\{(S_y)_{ij}\} = \sum_s E\{\lambda_s z_{s,i} z_{s,j}\} \quad (5.17)$$

où les z_s sont des variables décorréelées de matrice de covariance Σ , et les λ_s sont les valeurs propres de la matrice A .

$$E\{(S_y)_{ij}\} = \sum_s \lambda_s \Sigma_{ij} = \Sigma_{ij} \sum_s \lambda_s \quad (5.18)$$

Or $\sum_s \lambda_s = N$ (pour les calculs concernant les λ_s , se reporter à l'annexe A).

Donc :

$$E\{(S_y)\} = N\Sigma \quad (5.19)$$

Et, par conséquent, la moyenne de $\hat{\Sigma}$ est égale à Σ .

Calculons maintenant les moments d'ordre deux :

$$E\{(S_y)_{ij}(S_y)_{kl}\} = E\left\{\sum_{s,t} y_{s,i} y_{s,j} y_{t,k} y_{t,l}\right\} \quad (5.20)$$

$$E\{(S_y)_{ij}(S_y)_{kl}\} = E\left\{\sum_s y_{s,i} y_{s,j} y_{s,k} y_{s,l}\right\} + E\left\{\sum_{s \neq t} y_{s,i} y_{s,j} y_{t,k} y_{t,l}\right\} \quad (5.21)$$

Et comme les variables de la transformée de Fourier sont décorréelées :

$$E\{(S_y)_{ij}(S_y)_{kl}\} = \sum_s E\{y_{s,i} y_{s,j} y_{s,k} y_{s,l}\} + \sum_{s \neq t} E\{y_{s,i} y_{s,j} y_{t,k} y_{t,l}\} \quad (5.22)$$

Pour le premier terme, on se réfère à la formule des moments d'ordre quatre d'une variable gaussienne [1] :

$$E\{z_{s,i} z_{s,j} z_{s,k} z_{s,l}\} = \Sigma_{ij} \Sigma_{kl} + \Sigma_{ik} \Sigma_{jl} + \Sigma_{il} \Sigma_{jk} \quad (5.23)$$

avec $E\{zz^t\} = \Sigma$. On peut en déduire que :

$$\sum_s E\{y_{s,i} y_{s,j} y_{s,k} y_{s,l}\} = \sum_s \lambda_s^2 (\Sigma_{ij} \Sigma_{kl} + \Sigma_{ik} \Sigma_{jl} + \Sigma_{il} \Sigma_{jk}) \quad (5.24)$$

Quant au deuxième terme, puisque l'on sait que y_s et y_t sont décorréelées, on peut écrire :

$$E\{y_{s,i} y_{s,j} y_{t,k} y_{t,l}\} = E\{y_{s,i} y_{s,j}\} E\{y_{t,k} y_{t,l}\} \quad (5.25)$$

$$\sum_{s \neq t} E \{y_{s,i} y_{s,j} y_{t,k} y_{t,l}\} = \sum_{s \neq t} \lambda_s \lambda_t \Sigma_{ij} \Sigma_{kl} = \Sigma_{ij} \Sigma_{kl} \sum_{s \neq t} \lambda_s \lambda_t \quad (5.26)$$

Par conséquent :

$$E\{(S_y)_{ij}(S_y)_{kl}\} = (\Sigma_{ij}\Sigma_{kl} + \Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \sum_s \lambda_s^2 + \Sigma_{ij}\Sigma_{kl} \sum_{s \neq t} \lambda_s \lambda_t \quad (5.27)$$

$$E\{((S_y)_{ij} - N\Sigma_{ij})((S_y)_{kl} - N\Sigma_{kl})\} = (\Sigma_{ij}\Sigma_{kl} + \Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \sum_s \lambda_s^2 + \Sigma_{ij}\Sigma_{kl} \sum_{s \neq t} \lambda_s \lambda_t - N^2 \Sigma_{ij}\Sigma_{kl} \quad (5.28)$$

$$E\{((S_y)_{ij} - N\Sigma_{ij})((S_y)_{kl} - N\Sigma_{kl})\} = (\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \sum_s \lambda_s^2 \quad (5.29)$$

L'expression des moments d'ordre deux de la matrice $\hat{\Sigma}$ est donc la suivante :

$$E\{(\hat{\Sigma}_{ij} - \Sigma_{ij})(\hat{\Sigma}_{kl} - \Sigma_{kl})\} = (\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \frac{1}{N^2} \sum_s \lambda_s^2 \quad (5.30)$$

On approxime donc $\hat{\Sigma}$ par une variable Σ_w telle que $\frac{N^2}{\sum_s \lambda_s^2} \Sigma_w$ suive une loi de Wishart à $\frac{N^2}{\sum_s \lambda_s^2}$ degrés de liberté.

Considérons, par exemple, un modèle isotrope en 4-connexité. Dans ce cas, on a :

$$\lambda_{ij} = 1 - 2a \cos\left(\frac{2\pi i}{N_r}\right) - 2a \cos\left(\frac{2\pi j}{N_c}\right) \quad (5.31)$$

Donc :

$$\sum_{i,j} \lambda_{ij}^2 = \sum_{i,j} \left(1 - 2a \cos\left(\frac{2\pi i}{N_r}\right) - 2a \cos\left(\frac{2\pi j}{N_c}\right)\right)^2 \quad (5.32)$$

$$\sum_{i,j} \lambda_{ij}^2 = N^2 + 4a^2 \sum_{i,j} \cos^2\left(\frac{2\pi i}{N_r}\right) + 4a^2 \sum_{i,j} \cos^2\left(\frac{j}{N_c}\right) \quad (5.33)$$

Or :

$$\sum_{i=0}^{N_r-1} \cos^2\left(\frac{2\pi i}{N_r}\right) = \frac{N_r}{2} \quad (5.34)$$

Donc :

$$\sum_{i,j} \lambda_{ij}^2 = N + 4a^2 N = N(1 + 4a^2) \quad (5.35)$$

Dans ce cas, on approche la loi de $\hat{\Sigma}$ par la loi d'une variable Σ_w telle que $\frac{N}{(1+4a^2)}\Sigma_w$ suive une loi de Wishart à $\frac{N}{(1+4a^2)}$ degrés de liberté.

On peut noter que la loi approchée de $\hat{\Sigma}$ est centrée en Σ , et que la précision est d'autant plus grande que le nombre de pixels servant à l'estimation est grand et que la corrélation est faible.

Distance

On va maintenant chercher à déterminer une expression de la distance de Bhattacharyya entre deux distributions du type décrit plus haut. La distance de Bhattacharyya entre deux distributions de $\hat{\Sigma}_1$ et $\hat{\Sigma}_2$, telles que $n_1\hat{\Sigma}_1$ suive une distribution de Wishart de paramètres (Σ_1, n_1) et que $n_2\hat{\Sigma}_2$ suive une distribution de Wishart de paramètres (Σ_2, n_2) , peut se calculer ainsi :

$$B_{12} = \int_S \sqrt{p_1(S)p_2(S)} dS \quad (5.36)$$

où p_1 est la densité de probabilité de la première variable, et p_2 de la deuxième variable. On peut montrer qu'une expression de la distance B_{12} est donnée par :

$$B_{12} = \frac{n_1^{\frac{pn_1}{4}} n_2^{\frac{pn_2}{4}} |\Sigma_{12}|^{\frac{n_1+n_2}{4}}}{|\Sigma_1|^{\frac{n_1}{4}} |\Sigma_2|^{\frac{n_2}{4}}} \quad (5.37)$$

où $\Sigma_{12}^{-1} = \frac{1}{2} (n_1\Sigma_1^{-1} + n_2\Sigma_2^{-1})$.

Dans le cas où $n_1 = n_2$, l'expression devient très simple :

$$B_{12} = \frac{|\tilde{\Sigma}_{12}|^{\frac{n}{2}}}{|\Sigma_1|^{\frac{n}{4}} |\Sigma_2|^{\frac{n}{4}}} \quad (5.38)$$

où $\tilde{\Sigma}_{12}^{-1} = \frac{1}{2} (\Sigma_1^{-1} + \Sigma_2^{-1})$.

On a donc, grâce à l'équation (5.37), une expression simple de la distance de Bhattacharyya entre deux distributions de $\hat{\Sigma}$ pour deux classes différentes.

Poursuite de projection sur les paramètres de texture

L'algorithme de réduction de dimension des paramètres de texture est une poursuite de projection qui s'appuie sur la distance de Bhattacharyya décrite ci-dessus. De même que

pour la réduction de dimension des bandes de radiométrie, on va chercher à maximiser la distance entre les distributions des paramètres des échantillons d'apprentissage.

L'algorithme consiste en une technique de poursuite de projection par croissance d'espace, telle que décrite au paragraphe ?? . A partir d'un espace vide, on augmente cet espace itérativement d'une dimension, jusqu'à la dimension désirée. A chaque itération, on ajoute à la base courante un vecteur directeur de la droite qui, dans l'orthogonal de la base courante, maximise la distance de Bhattacharyya entre les distributions des paramètres de texture des classes prédéfinies.

5.3.5 Règle de classification

Soit $\mathbf{X} = \{\mathbf{x}_s\}_{s \in S}$ l'image finale dont on se sert pour la classification. Cette image est constituée d'un ensemble de pixels \mathbf{x}_s qui sont eux-mêmes constitués de deux parties :

$$\mathbf{x}_s = \begin{pmatrix} x_s^{(r)} \\ x_s^{(t)} \end{pmatrix} \quad (5.39)$$

où $x_s^{(r)}$ est la partie « radiométrie » de \mathbf{x}_s , et $x_s^{(t)}$ sa partie « texture ». $x_s^{(r)}$ est le résultat de la projection des données sous le sous-espace \mathcal{E}_1 maximisant l'indice de projection lié à la distance de Bhattacharyya entre les classes prédéfinies, ayant supposé une modélisation gaussienne. $x_s^{(t)}$ est une matrice de covariance conditionnelle calculée dans un sous-espace \mathcal{E}_2 . Ce dernier est l'espace dans lequel est maximisé l'indice de projection fondé sur la distance de Bhattacharyya entre l'approximation des lois des matrices de covariances conditionnelles des classes prédéfinies.

La règle de classification est la suivante :

$$\begin{aligned} \lambda_s &= \arg \max_{l \in \{0 \dots L-1\}} P_l(\mathbf{x}_s) \\ &= \arg \max_{l \in \{0 \dots L-1\}} P_l\left(x_s^{(r)}, x_s^{(t)}\right) \end{aligned} \quad (5.40)$$

On suppose que les deux mesures (radiométrie et texture) sont indépendantes, ce qui simplifie la règle de décision. On a donc :

$$\lambda_s = \arg \max_{l \in \{0 \dots L-1\}} P_l\left(x_s^{(r)}\right) P_l\left(x_s^{(t)}\right) \quad (5.41)$$

où $P_l\left(x_s^{(r)}\right)$ est modélisé par une gaussienne, et $P_l\left(x_s^{(t)}\right)$ par une loi de Wishart, et leurs paramètres sont calculés pour chaque classe sur les échantillons d'apprentissage.

5.4 Résultats et comparaisons

Dans ce paragraphe, on analyse les résultats obtenus par les différents algorithmes présentés, on montre l'apport de l'analyse de texture dans les résultats de classification, et l'on compare les résultats obtenus avec ceux que fournissent d'autres méthodes plus classiques d'analyse de texture.

5.4.1 Analyse de texture grâce aux matrices de cooccurrence

Au premier chapitre, on a évoqué les matrices de cooccurrences, comme étant un moyen populaire d'analyser les textures. Il existe divers paramètres que l'on peut tirer de ces matrices, comme par exemple l'énergie, l'inertie, l'entropie, la dissimilarité, le contraste, les moments [38]...

La figure 5.6 présente des images de paramètres issus des matrices de cooccurrence, que le logiciel ENVI [28] propose pour l'analyse de texture, calculés sur une bande d'une image extraite d'une image AVIRIS. On remarque que certains de ces paramètres ne sont pas adaptés à nos besoins. En effet, l'entropie et le second moment renvoient des images quasiment constantes, et la corrélation ou l'homogénéité des images peu informatives. En réalité, après examen de ces images, seules les images de moyenne et de dissimilarité semblent intéressantes dans l'analyse qui nous intéresse. De plus, le paramètre moyenne n'est pas un paramètre de cooccurrence, puis qu'il s'agit de la moyenne locale des pixels dans la fenêtre. Quant au paramètre de dissimilarité, il se calcule ainsi :

$$D = \sum_{i,j=0}^{N_q-1} P_{ij} |i - j| \quad (5.42)$$

où N_q est le nombre de niveaux de gris quantifiés.

Ce sont donc ces paramètres que nous utiliserons pour faire des comparaisons avec les méthodes par champs de Markov gaussiens. Les images à dimension réduite par la méthode choisie seront augmentées des images de paramètres de moyenne et dissimilarité calculés pour chaque bande et pour des vecteurs de déplacement $d_1 = (0,1)$ et $d_2 = (1,0)$ en ce qui concerne la dissimilarité, grâce à une fenêtre de 9×9 . Une image réduite à n bandes de radiométrie sera augmentée de $3n$ bandes de texture, pour former une image de $4n$ bandes.

La modélisation des paramètres de texture est une modélisation gaussienne non contextuelle.

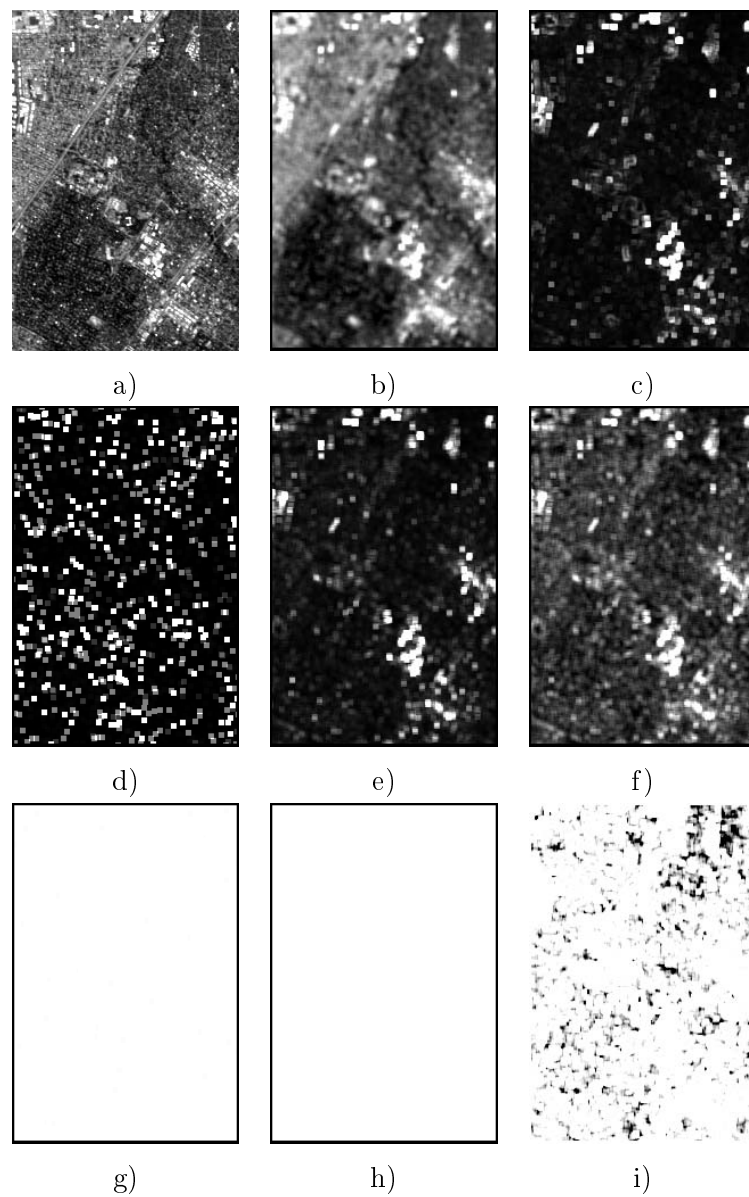


FIG. 5.6 – a) Une bande de l'image analysée par une matrice de cooccurrence, b) Moyenne, c) Variance, d) Homogénéité, e) Contraste, f) Dissimilarité, g) Entropie, h) Second Moment, i) Corrélation.

5.4.2 Récapitulatif des algorithmes testés

Plusieurs algorithmes de réduction de données et de classification déjà existants ont été présentés, et nous avons proposé des méthodes fondées sur les champs de Markov dans ces deux buts également. Il convient maintenant, avant de montrer les résultats obtenus en classification, de récapituler les algorithmes que l'on va tester.

En matière de réduction d'espace, on teste deux algorithmes que l'on considère comme des références : l'analyse en composantes principales (ACP), et la poursuite de projection séquentielle (PP) s'appuyant sur l'indice de Jimenez et Landgrebe, la distance de Bhattacharyya du modèle gaussien non-contextuel. On compare donc les deux méthodes que l'on a proposées à ces méthodes de référence. Les méthodes proposées sont la poursuite de projection utilisant comme indice la distance de Bhattacharyya adaptée au modèle markovien gaussien (PPmg), et la méthode d'extraction des spectres purs (SP).

En matière de classification, on teste des méthodes exploitant la notion de texture, ainsi que la méthode la plus simple : le maximum de vraisemblance du modèle gaussien ponctuel (MV). Les algorithmes de classification de texture sont l'algorithme de maximum de vraisemblance du modèle markovien gaussien par blocs (MVmg), présenté au paragraphe 5.2.2, et l'algorithme avec ajout de bande (ABmg) proposé au paragraphe 5.3. Enfin, une comparaison est faite avec un algorithme du même type que l'*ABmg* : l'ajout aux images de radiométrie des images des paramètres extraits des matrices de cooccurrence que sont la dissimilarité et la moyenne (Cooc).

5.4.3 Images de synthèse

Nous avons d'abord testé les algorithmes d'analyse et classification sur des images de synthèse représentant des collages de quatre images, étant elles-mêmes des simulations de champs de Markov gaussiens multivariés. Le modèle dont sont issues ces images est un modèle isotrope en 4-connexité, et les images sont générées grâce à l'algorithme de Metropolis [64, 78]. On teste les algorithmes de classification sur des images ayant différents rapports S/B.

Les images, sur lesquelles on teste les algorithmes, sont les images de texture du paragraphe 2.3.8 : *colSynth1*, *colSynth2*, *colSynth5* et *colSynth10* de la figure 2.2 (dans l'ordre décroissant de S/B¹). Chaque quart de l'image est constitué d'un carré de texture à 3 bandes spectrales possédant ses propres paramètres Σ et a (a est un scalaire). Ces images ne nécessitent pas de réduction de dimension, puisque celle-ci est déjà faible en comparaison du nombre d'échantillons d'apprentissage. On effectue ici uniquement une comparaison

1. Dans notre définition du rapport S/B, nous considérons le signal comme étant la moyenne de la texture caractérisable par la radiométrie seule, et le bruit comme la texture de moyenne nulle.

entre les différents algorithmes de classification.

Les zones d'apprentissage correspondent à un carré de 19 pixels de côté au centre de chaque classe, c'est-à-dire à 361 pixels. Bien entendu, il est nécessaire de ne pas prendre des pixels isolés, mais bien des zones connexes, car l'on cherche à saisir l'information d'interaction entre pixels voisins.

Résultats de ABmg

Les résultats sont présentés en figure 5.7. On constate que l'on n'obtient des bords francs que si les textures sont très facilement séparables dans le domaine spectral. En l'occurrence, seule l'image *colSynth1* permet d'obtenir un résultat satisfaisant concernant les contours de texture. La raison en est simple : la figure 5.8 représente la projection dans un plan du nuage de points correspondant à une transition entre deux textures dans les images *colSynth1* et *colSynth2*. Les plans de projection sont différents, on les a choisis afin de bien distinguer les deux nuages de points correspondant à chacune des textures. On constate tout de même que les nuages sont mélangés dans le deuxième cas, ce qui rend la distinction des deux nuages par l'algorithme de *k-means* beaucoup plus délicate. Pour cette raison, les contours des textures sont mal détectés, et l'heuristique est inefficace dans le cas de l'image *colSynth2*.

Résultats de MVmg

Les résultats de l'algorithme *MVmg* sur les mêmes images simulées que précédemment sont présentés en figure 5.9. Le premier constat est que la première image, *colSynth1*, est très mal classifiée bien que ses textures soient très facilement reconnaissables. Ceci vient du fait que l'algorithme ne laisse que très peu d'importance à la radiométrie pour n'utiliser que l'information de texture. Les autres images, bien que moins bien classifiées que par l'algorithme *ABmg*, sont assez bien reconnues, surtout quand l'information principale est texturale. Les deux images aux plus faibles rapport S/B sont les mieux classifiées.

Comparaisons

Les résultats de classification sont recensés dans le tableau 5.1. Pour chaque classification, on présente dans le tableau, à la fois le taux de bonne classification et l'indice κ de celle-ci. Le taux de bonne classification est le pourcentage de pixels classifiés dans leur classe réelle. L'indice κ est un indice souvent utilisé pour mesurer la qualité d'une classification [60]. S'appuyant sur la matrice de confusion, dont chaque coefficient m_{ij} est égal au nombre de pixels de la classe i classifiés dans la classe j , il donne une vision synthétique de celle-ci. En pratique, ce coefficient se calcule comme un quotient où le numérateur est égal

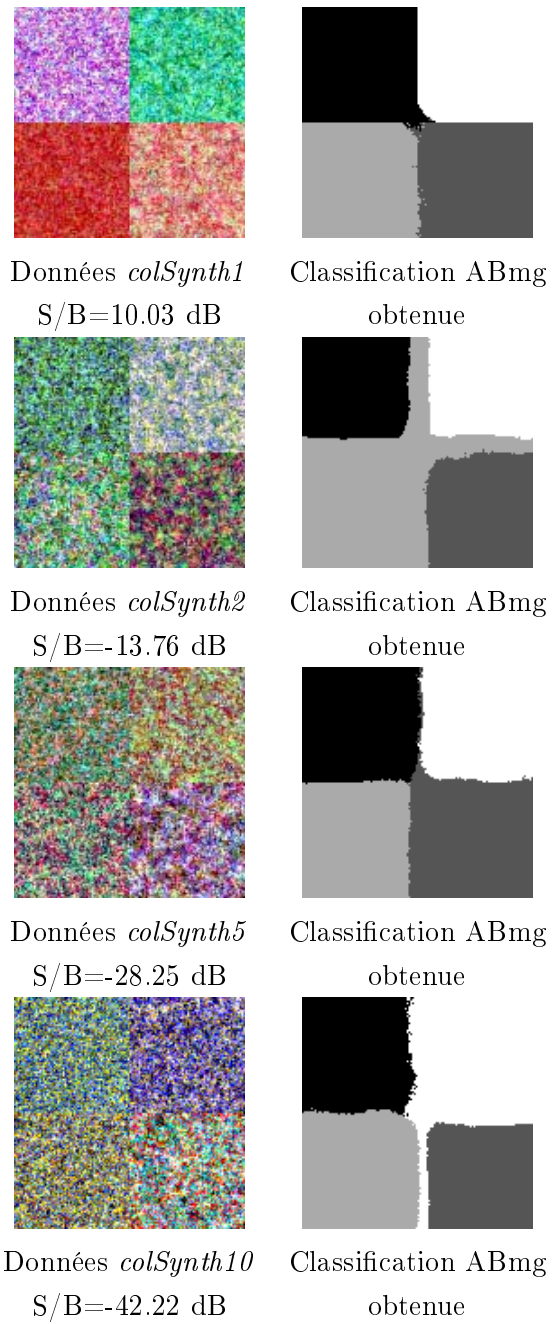


FIG. 5.7 – Données simulées à différents rapports S/B (colonne de gauche), classification obtenue par la méthode ABmg.

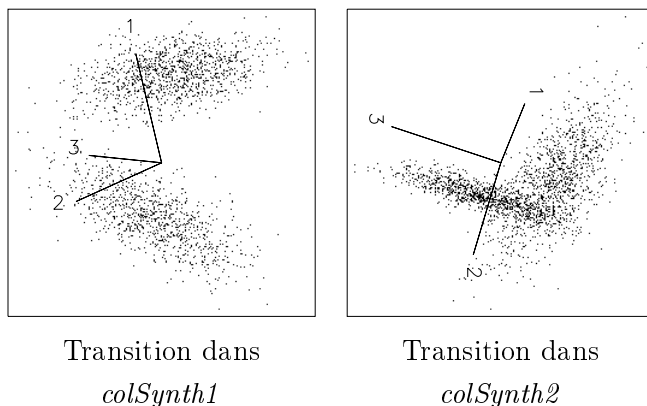


FIG. 5.8 – *Projection du nuage de points correspondant à une zone de transition entre deux textures : a) les deux zones sont distinctes, b) les deux zones sont entremêlées. A noter que les points de vue, différents, ont été choisis parce qu'ils présentaient les nuages de la façon la plus séparée possible.*

au taux de bonne classification réel moins le taux de classification que l'on aurait obtenu par une classification complètement au hasard, et le dénominateur est égal à 1 moins le taux de classification au hasard. Cet indice est, en général, compris entre 0 et 1, 0 correspondant à une classification faite au hasard et 1 à une classification parfaite. Un indice κ inférieur à zéro signifie que la classification est pire qu'une classification au hasard, c'est-à-dire que l'on a des confusions entre classes récurrentes.

On constate, avant tout, que les classificateurs ont des comportements différents vis-à-vis de l'évolution du rapport signal sur bruit (S/B). L'algorithme par ajout de bandes $ABmg$ est celui qui semble le mieux fonctionner. En effet, dans les forts rapports S/B, comme $colSynth1$, l'algorithme détecte de fortes différences de moyennes entre classes, et ainsi l'heuristique est utilisée dans l'algorithme d'estimation. Ceci a pour effet d'améliorer la classification pour cette image. En revanche, l'image $colSynth2$ n'est pas reconnue comme présentant de grandes différences entre classes, et par conséquent l'heuristique n'est pas utilisée et les transitions entre textures sont très mal classifiées. Pour les images $colSynth5$ et $colSynth10$, la classification est bonne, car les textures sont différenciables essentiellement par leurs caractéristiques spatiales.

Les algorithmes $MVmg$ et MV ont un comportement inverse l'un vis-à-vis de l'autre. En effet, alors que le premier fonctionne mal pour les images à fort rapport S/B et bien pour les images à faible S/B, le deuxième a un comportement inverse. Il est facile de se rendre compte que les images $colSynth5$ et $colSynth10$, avec leurs textures aux moyennes proches, sont difficilement classifiables en ne se fondant que sur la radiométrie. De même,

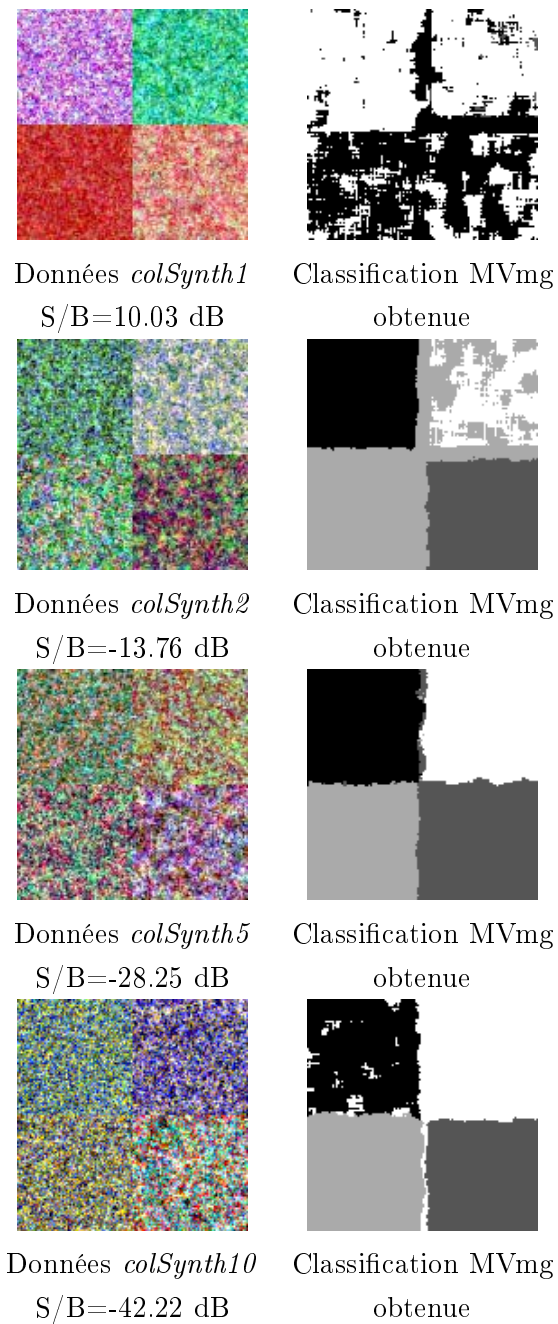


FIG. 5.9 – Données simulées à différents rapports S/B (colonne de gauche), classification obtenue par la méthode MVmg.

Image	ABmg	MVmg régularisé	MV régularisé	Cooc régularisé
colSynth1	97.6 (0.97)	20.9 (-0.05)	98.2 (0.98)	45.5 (0.28)
colSynth2	89.8 (0.86)	82.8 (0.77)	95.3 (0.93)	83.6 (0.78)
colSynth5	96.9 (0.96)	96.9 (0.96)	87.4 (0.82)	88.3 (0.84)
colSynth10	95.1 (0.93)	94.9 (0.93)	78.8 (0.72)	44.2 (0.26)

TAB. 5.1 – Taux de bonne classification et indice κ pour les classifications obtenues par 4 méthodes différentes à partir des images de synthèse à différents rapports S/B .

les images *colSynth1* et *colSynth2* sont porteuses d’une très grande information au niveau de leur moyenne, ce qui rend l’algorithme *MVmg* peu (voire nullement) efficace pour ces images. Ceci est illustré par le cas extrême de la classification de l’image *colSynth1* pour laquelle l’indice κ est inférieur à zéro, c’est-à-dire inférieure à un assignement des étiquettes au hasard. Cela vient du fait que deux classes ne contiennent aucuns pixels.

L’algorithme *Cooc* prenant en compte les mesures de cooccurrence, est globalement peu efficace du fait que les deux paramètres utilisés renvoient des mesures contradictoires pour les deux images extrêmes (*colSynth1* et *colSynth10*). En ce qui concerne *colSynth1*, la moyenne est très discriminante alors que la dissimilarité l’est moins, et c’est le contraire pour l’image *colSynth10* (cf. figure 5.10), où la moyenne introduit d’importantes confusions. Pour les deux autres images, les deux paramètres sont moyennement discriminants ce qui donne des classifications relativement bonnes, mais moins bonnes qu’avec les algorithmes précédents.

5.4.4 Collage de textures extraites d’images AVIRIS

Il reste donc à tester les algorithmes de classification proposés sur des données hyperspectrales réelles. Les images sur lesquelles on travaille sont des extraits d’une image du capteur AVIRIS prise au dessus de la région de Moffett Field, en Californie. Ici, on effectue des comparaisons concernant à la fois les algorithmes de classification et de réduction de la dimension.

La figure 5.11 représente une mosaïque de trois types différents de texture urbaine, plus une partie d’un plan d’eau. On teste les différents algorithmes sur cette image, dans un

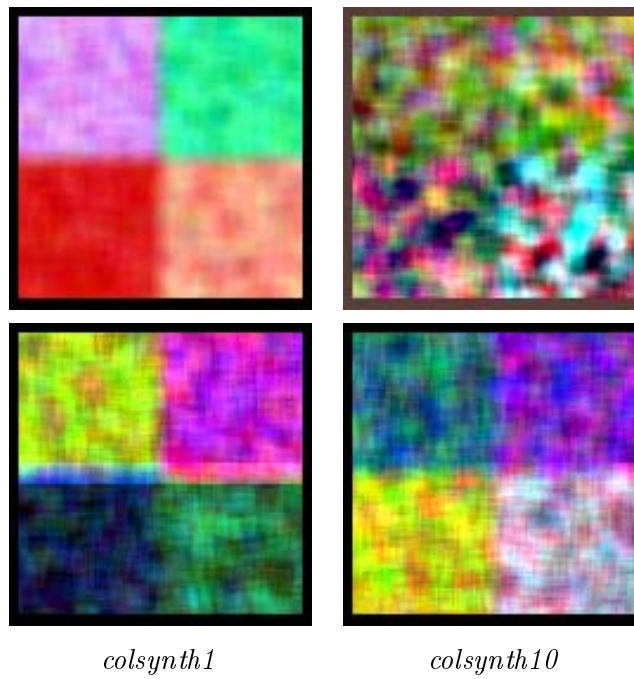


FIG. 5.10 – Les 3 bandes en fausses couleurs de paramètres de moyenne (haut) et dissimilarité (bas) pour les images colSynth1 et colSynth10, $d = (0,1)$.

premier temps, pour avoir une idée de la qualité des résultats que l'on peut obtenir, même si les transitions entre textures introduites artificiellement sont bien plus abruptes que celles qui existent naturellement dans les images réelles. Donc, sur ce type d'images, les méthodes contextuelles auront un léger désavantage *a priori*, du fait de l'utilisation de fenêtre d'analyse, sur les transitions. Néanmoins, l'intérêt des tests sur ce type d'images artificielles n'est pas négligeable, puisque l'on a des textures homogènes, que l'on sait délimiter, avec une vérité de terrain assez précise (hormis quelques particularités locales).

Sur ce genre de configurations, il est probable que de fortes régularisations résultent en de meilleures classifications, puisque l'on a quatre carrés de texture qui sont bien séparés, aucunes classes imbriquées dans d'autres, ou dispersées dans l'image en petites zones.

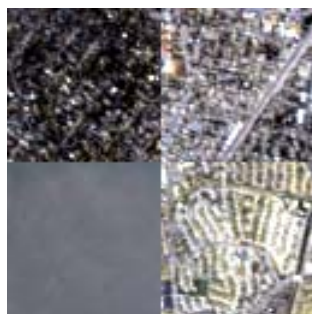


FIG. 5.11 – 3 bandes extraites du spectre visible d'un collage de 4 textures AVIRIS.

On réalise les tests sur un sous-ensemble de 40 bandes spectrales parmi les 224 disponibles, prises dans la même zone (visible). On compare les différents algorithmes pour plusieurs valeurs de n , la dimension finale pour la radiométrie. La valeur de la dimension finale pour la texture est choisie empiriquement égale à n dans le cas de l'algorithme *ABmg*, car celle-ci donne les meilleurs résultats. Les échantillons d'apprentissage sont au nombre de 361 par classe, il s'agit du carré de 19 pixels de côté au centre de chaque texture. Tous les pixels de l'image sont utilisés pour la validation.

Les quatre méthodes comparées sont les suivantes :

- réduction de dimension par poursuite de projection avec comme indice de projection la distance de Bhattacharyya d'un modèle pixélique, classification MV sur le même modèle, notée *PP/MV*, figure 5.12.a,
- réduction de dimension par poursuite de projection avec comme indice de projection la distance de Bhattacharyya du modèle markovien gaussien, classification MV sur le même modèle, notée *PPMG/MVmg*, figure 5.12.b,
- réduction de dimension par poursuite de projection avec comme indice de projection

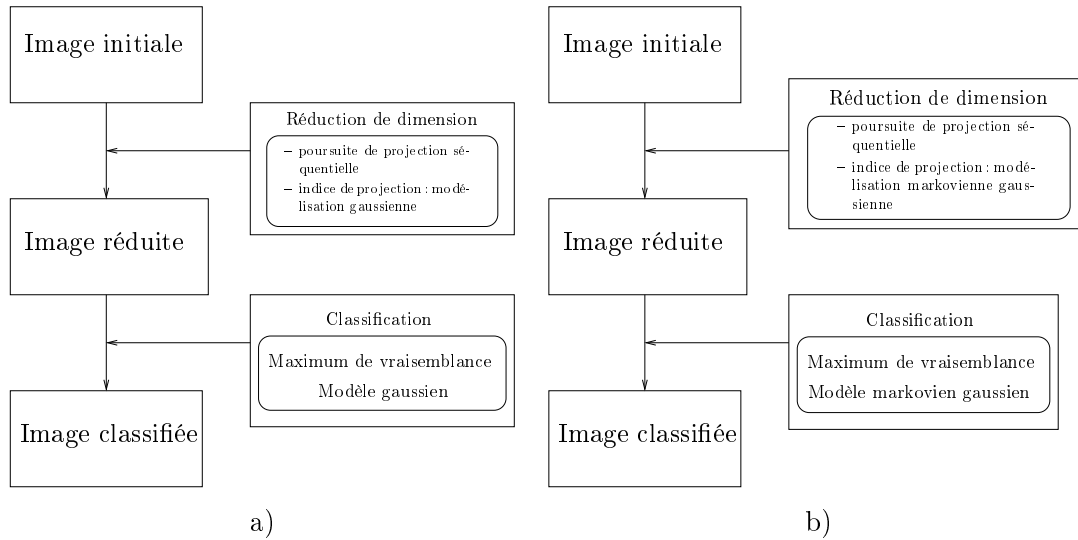


FIG. 5.12 – a) *Algorithme PP/MV*, b) *Algorithme PP/MVmg*.

la distance de Bhattacharyya d'un modèle pixélique, classification MV avec ajout de bande, notée *PP/ABmg*, figure 5.12,

- réduction de dimension par analyse en composantes principales, extraction des paramètres *moyenne* et *dissimilarité* pour des matrices de cooccurrence correspondant à des vecteurs de déplacement égaux à (1,0) et (0,1), et classification par MV sur un modèle gaussien de l'image augmentée des paramètres, notée *ACP/Cooc*, figure 5.14.

La méthode exploitant les paramètres de cooccurrence est précédée d'une ACP afin de décorréler les bandes, de façon à ce que l'utilisation des paramètres de texture extraits bande par bande soient eux aussi le plus décorrélés possible.

Enfin, les résultats donnés par les quatre algorithmes sont régularisés par l'introduction d'un modèle *a priori* de Potts, la régularisation étant faite par l'algorithme ICM [7].

On constate plusieurs choses sur la figure 5.15, où sont donnés les résultats de classification sur la mosaïque de textures AVIRIS (à travers le taux de bonne classification). La première constatation est que les classifications sont globalement de meilleure qualité avec l'augmentation du nombre de bandes. Toutefois, les approximations par découpage du spectre de la poursuite de projection font qu'il y a parfois une diminution du taux de bonne classification avec l'ajout d'une bande. De plus, la classification « sature » quand on dépasse un nombre de bandes minimum, qu'on appellera la *dimension intrinsèque* des données. Cette dimension se situe un peu au-delà de 10 dans le cas de cette mosaïque.

La deuxième constatation que l'on peut faire au vu des résultats est que la première méthode qui s'appuie sur la vraisemblance du modèle défini est peu efficace. En effet, on

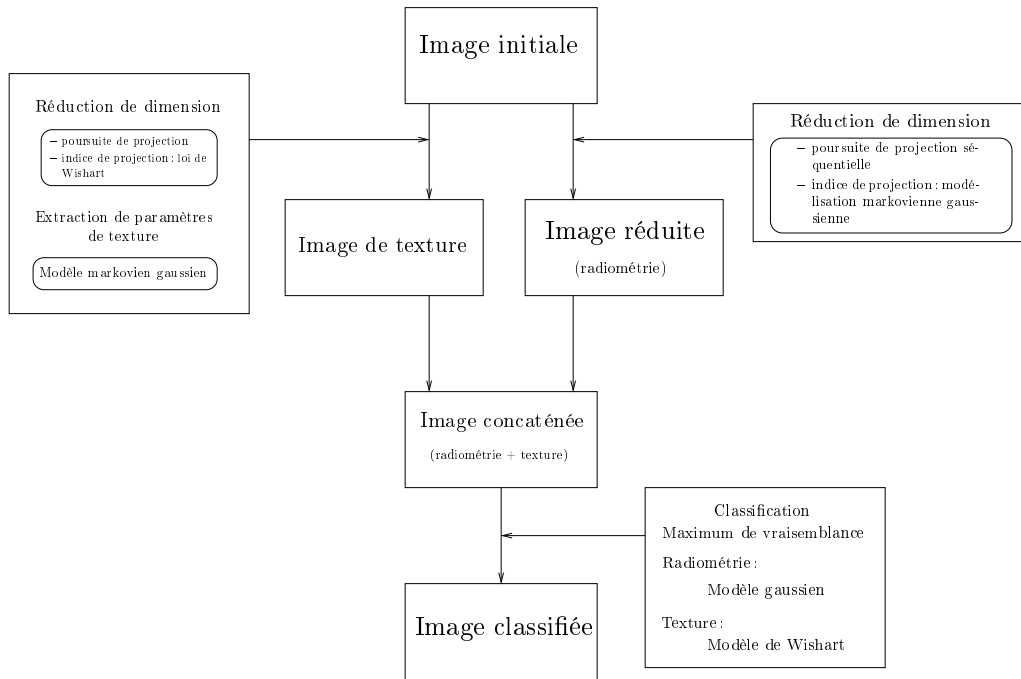


FIG. 5.13 – *Algorithme PP/ABmg*

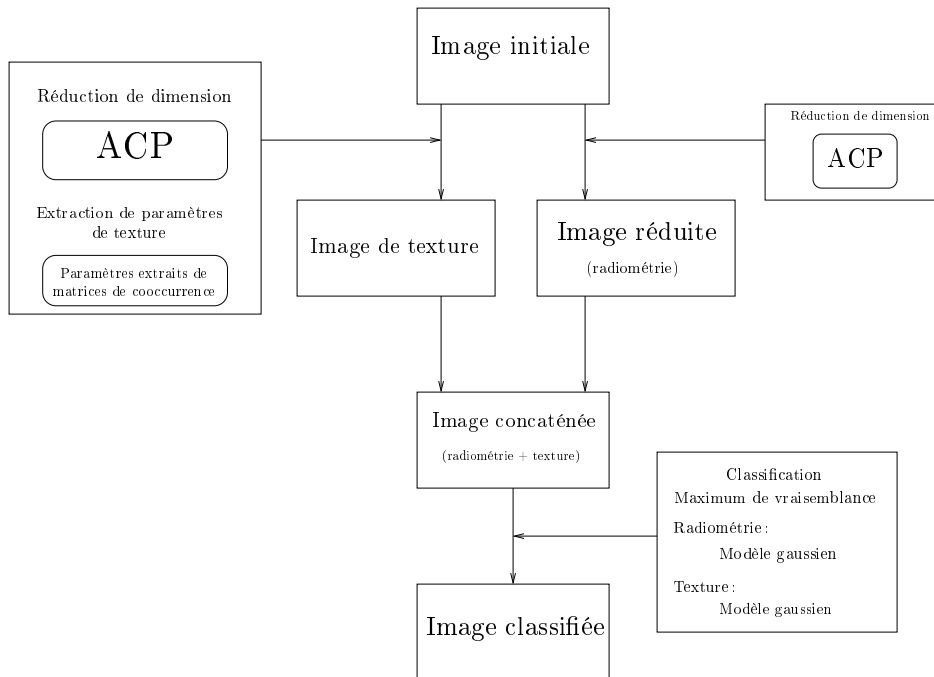


FIG. 5.14 – *Algorithme ACP/Cooc*

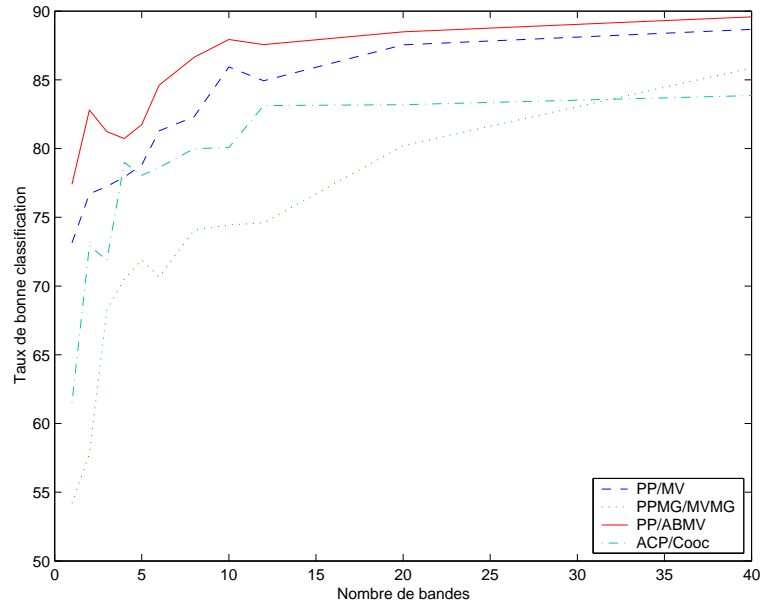


FIG. 5.15 – Evolution du taux de bonne classification pour les 4 méthodes comparées sur la mosaïque de la figure 5.11.

est dans un cas où les moyennes des textures sont aussi discriminantes que les variations spatiales, et cet algorithme ne prend pas en compte de façon équivalente ces deux aspects.

Par ailleurs, la méthode *ACP/Cooc* a tendance à saturer assez rapidement en terme de bonne classification, puisque au-delà de douze bandes la classification ne bouge que très peu. Pourtant, les résultats obtenus par celle-ci sont bons, bien que la méthode soit très simple, de même que ceux obtenus avec une modélisation gaussienne non contextuelle *PP/MV*, qui sont pratiquement aussi bons que ceux obtenus par la méthode *PPMG/ABmg*, qui donne les meilleurs taux de bonne classification.

Enfin, on remarque que les résultats obtenus par les méthodes *ACP/Cooc* et *PPMG/ABmg* reflètent le problème de l'estimation des paramètres dans une fenêtre glissante, qui crée des erreurs de classification aux transitions entre textures.

5.4.5 Images réelles AVIRIS

Un autre test est mené cette fois sur une vraie sous-image de la même image AVIRIS. L'image de test est l'image de la figure 5.4. De la même façon que pour le test précédent, on utilise un sous-ensemble des bandes disponibles (ici 40 bandes prises dans le domaine visible). Sur cette image de test, on a défini deux sortes de tissus urbains différents et trois classes non-urbaines. On opère la classification sur ces cinq classes et le résultat est validé

pour les deux classes de zones urbaines qui nous intéressent, sur des zones de validation différentes de celles d'apprentissage. Les zones d'apprentissage et de validation ont été déterminées grâce à des cartes de l'USGS de la région. La taille des ensembles d'apprentissage est de 361 échantillons pour les zones urbaines, et oscille entre 182 et 559 échantillons pour les classes non-urbaines. Les courbes de la figure 5.16 donnent l'évolution du taux de bonne classification des zones urbaines par les méthodes PP/MV , $PP/ABmg$ et $ACP/Cooc$. En ce qui concerne la modélisation proposée, on garde uniquement la méthode $PP/ABmg$ (et non $PP/MVmg$) car c'est celle qui donne les meilleurs résultats sur les images réelles.

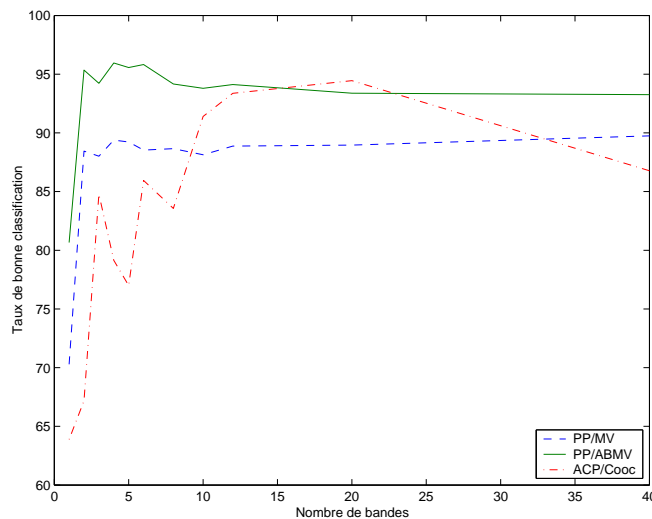


FIG. 5.16 – Evolution du taux de bonne classification pour les 4 méthodes comparées sur l'extrait d'image AVIRIS de Moffett Field.

Le premier constat qu'on peut faire au vu de la courbe de bonne classification (cf. figure 5.16) est que le nombre de bande nécessaires pour obtenir une bonne classification est très faible, puisqu'avec la méthode $PP/ABmg$ il suffit de deux bandes de texture et du même nombre de bandes de radiométrie pour obtenir un taux de 95% de pixels bien classifiés.

Le deuxième constat est que les courbes montrent une supériorité du taux de bonne classification des algorithmes avec ajout de paramètres de texture : $ACP/Cooc$ et $PP/ABmg$, pour cette image. Les meilleurs résultats sont obtenus, en général, avec $PP/ABmg$. La baisse du taux obtenu avec $ACP/Cooc$ est sans doute une manifestation du phénomène de Hughes, car avec l'augmentation du nombre de bandes, on augmente d'autant plus le nombre de bandes de texture. Une classification sur 40 bandes spectrales par cette méthode se fait donc en réalité au total sur 160 bandes (40 bandes spectrales, 40 bandes correspondant au paramètre moyenne, et 80 bandes correspondant aux dissimilarités pour les deux vecteurs

de déplacements $d_1 = (0.1)$ et $d_2 = (1.0)$.

Les résultats numériques sont toutefois légèrement trompeurs en ce qui concerne la classification par la méthode *ACP/Cooc*, car malgré des taux de bonne classification très acceptables pour des nombres de bandes élevés, il s'avère que visuellement la classification n'est pas satisfaisante. C'est ce que l'on constate sur la figure 5.17. La segmentation de la zone grise est beaucoup moins précise au niveau des contours pour la méthode *ACP/Cooc*, et en ce qui concerne la zone de grands bâtiments (en noir), elle déborde sur la zone résidentielle, la zone grise. Ces différences, entre résultat numérique et visuel, viennent du fait que les problèmes de cette classification viennent principalement des bords des zones, et que les échantillons utilisés pour la validation sont pour la plupart situés nettement à l'intérieur des zones urbaines.

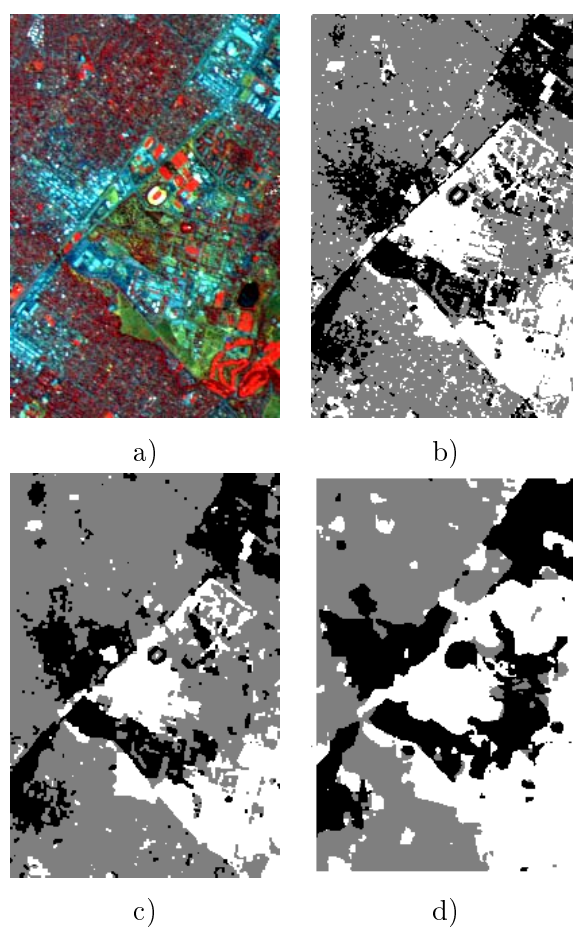


FIG. 5.17 – Classification par 3 méthodes de l'image AVIRIS, en gardant 10 bandes de radiométrie : a) 3 bandes de l'image originale, b) PP/MV, c) PP/ABmg (5 bandes de texture), d) ACP/Cooc.

Pour ce qui est de la classification par l'algorithme *PP/ABmg*, il faut prendre certaines précautions sans lesquelles le résultat de la classification est médiocre. Celle-ci est de ne pas utiliser, pour la partie texture dans la formule de la vraisemblance (5.41), une loi de Wishart dont le nombre de degrés de liberté est trop élevé. En effet, quand celui-ci augmente, cela signifie que la loi est beaucoup plus « piquée » autour de la moyenne, et que l'on a une grande confiance dans les données. Or les différentes irrégularités des textures font que l'on peut avoir des valeurs souvent très variables en ce qui concerne la matrice de covariance conditionnelle. En choisissant un nombre de degrés de liberté plus faible que le degré théorique, on élargit le pic de la loi de Wishart, et par conséquent, on accepte mieux les déviations par rapport à la moyenne de la loi.

D'autre part, il n'est pas nécessaire d'utiliser un grand nombre de bandes de texture. Car les bandes de texture sont parfois très corrélées et certaines bandes expriment quasiment la même information. Ceci permet d'économiser à la fois du temps et de la mémoire, pour une perte de classification très faible, voire nulle.

Ce phénomène est illustré par la figure 5.18 La classification de référence est la figure a), où le taux de classification est plus élevé. On se rend compte sur la classification où le nombre de degrés de liberté (b) est élevé, que la classification comporte de gros carrés. Ceci traduit le fait que les irrégularités du champ ont donné des valeurs très singulières pour la covariance conditionnelle, et que celles-ci ont été prises en compte dans l'estimation des paramètres de classe. Dans ce cas, on a trop confiance dans l'estimation de nos paramètres, ce qui dans ce cas est préjudiciable, puisque des valeurs aberrantes y ont participé. La figure c) illustre le fait que le nombre de bandes de texture ne doit pas forcément être très élevé, puisque la classification avec trois bandes de texture a) est proche de celle avec vingt bandes c). Cette dernière, est tout de même plus régulière, car la texture revêt plus d'importance, mais la zone de grands bâtiments n'est pas bien détectée. Dans tous les cas, le nombre de bandes de radiométrie est égal à quarante.

On teste les mêmes algorithmes sur une scène entière de l'image AVIRIS de Moffett Field (qui contient 4 scènes de 512 lignes de 614 pixels). Dans cette scène, on a défini deux types de zones urbaines, les grands bâtiments et les zones résidentielles, et cinq classes non-urbaines (eau, forêt...). La première image (a) de la figure 5.19 est un affichage de trois des bandes de l'image sur laquelle on travaille. Une nouvelle fois, on a fait le test sur un sous-ensemble de 40 bandes visibles de l'image originale, et la dimension finale en ce qui concerne la radiométrie est égale à 10. Comme auparavant, la classification est faite sur toutes les classes (ici au nombre de 7), puis les résultats sont validés sur les 2 classes de zones urbaines.

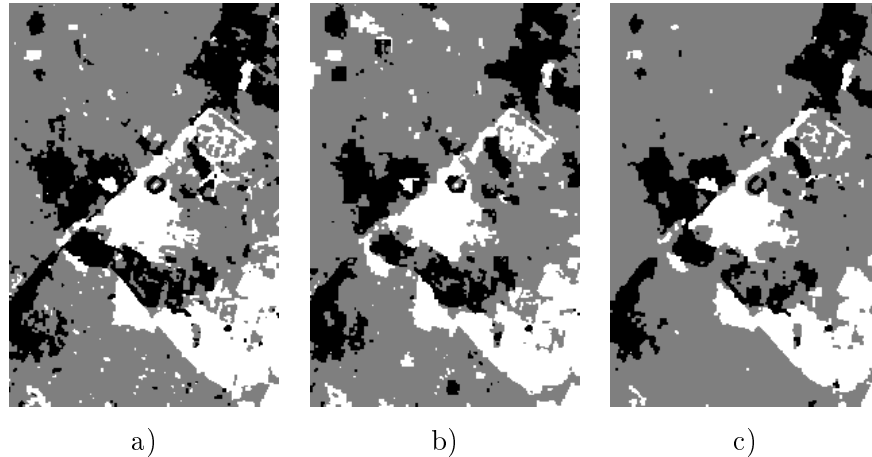


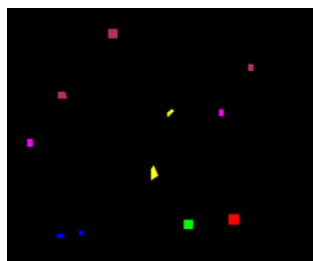
FIG. 5.18 – Influence du nombre de bandes de texture et du nombre de degrés de liberté dans l'approximation de la loi du paramètre de texture. a) Classification convenable (40 bandes de radiométrie, 3 bandes de texture), b) classification avec un nombre de degrés de liberté élevé, c) classification avec un nombre de bandes de texture élevé (20 bandes de texture).

La figure 5.19 montre l'image considérée, ainsi que les zones d'apprentissage et de validation. Les zones d'apprentissage comportent de 207 à 441 pixels, alors que les zones de validation comportent 9979 échantillons pour les grands bâtiments, et 4869 pour les zones résidentielles.

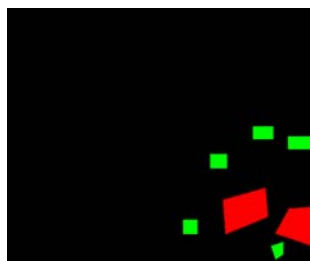
L'algorithme *ACP/Cooc* est appliqué en déterminant les 10 bandes de radiométrie par *ACP*, puis en extrayant pour chaque bande les 3 paramètres de texture. Pour l'algorithme *PP/ABmg*, on extrait 10 bandes de radiométrie, ainsi que les paramètres de texture calculés sur 2 bandes. Les images classifiées sont représentées sur la figure 5.20. On voit que la classification uniquement fondée sur la radiométrie (*PP/MV*) est de moins bonne qualité, notamment du fait que beaucoup de pixels isolés des zones résidentielles sont classifiés comme grands bâtiments, car elles sont constituées de maisons aux toits plus réfléchissants, quasiment comme les immeubles. Ceci est confirmé par le taux de bonne classification, qui est égal à 84.5% sur les zones de validation que l'on a données, avec 66% de bonne classification des zones résidentielles. Le deuxième constat est que parmi les 2 méthodes prenant en compte la texture, la méthode *ACP/Cooc* donne des résultats très décevants, malgré la qualité visuelle des paramètres de texture que sont la moyenne et la dissimilarité. Cette méthode manque vraisemblablement d'une bonne modélisation de la probabilité de ces paramètres, qui sont par défaut considérés comme gaussiens. Le taux de classification de cette méthode est donc assez faible (égal à 73.3%) surtout à cause d'un grand nombre de fausses alarmes de la classe noire (immeubles) dans la classe grise (zones résidentielles). La méthode



a)



b)



c)

FIG. 5.19 – a) affichage en fausse couleur des canaux $\{30,20,10\}$ de la scène AVIRIS, b) zones d'apprentissage, c) zones de validation.

qui donne les meilleurs résultats reste $PP/ABmg$ avec un taux de bonne classification de 95,5%, mais avec tout de même quelques fausses alarmes, détection de zones urbaines dans des zones en réalité rurales. Le bord du plan d'eau en haut à gauche de l'image est également détecté comme de la zone urbaine. Malgré cela, la segmentation intra-urbaine est globalement meilleure qu'avec les deux autres algorithmes.

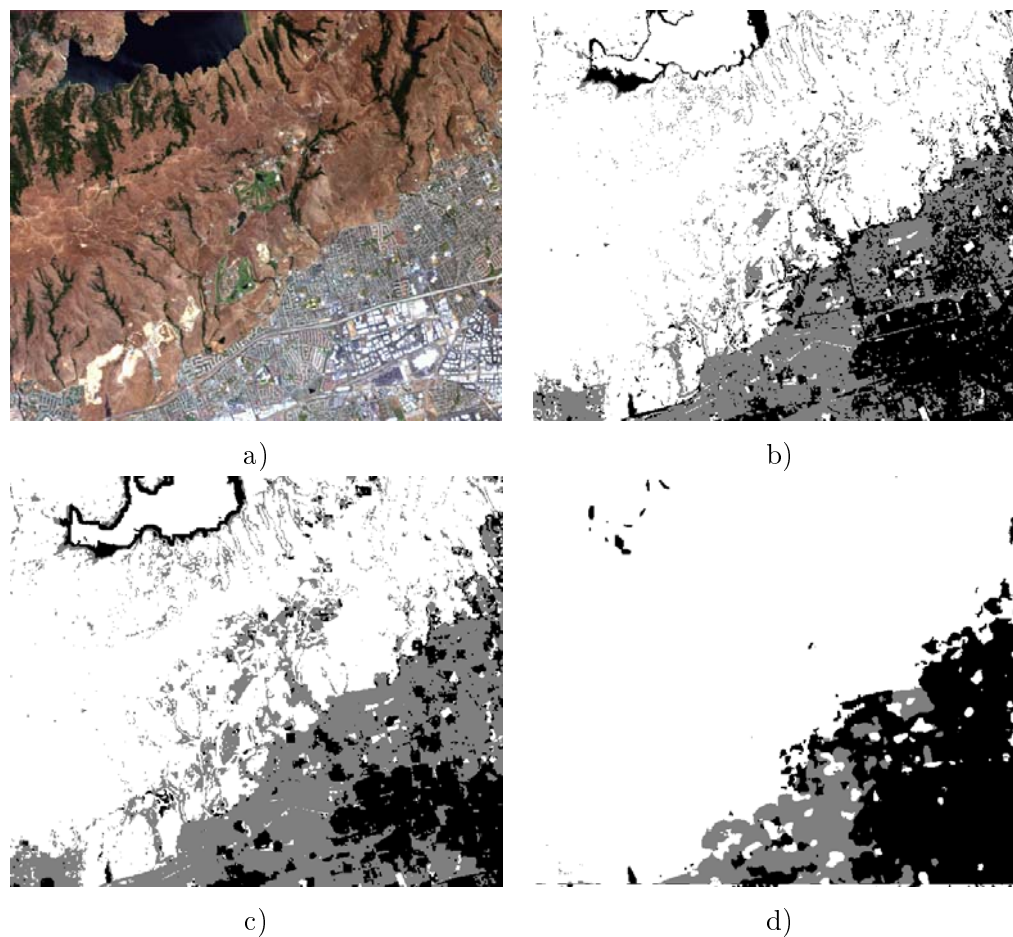


FIG. 5.20 – a) affichage en fausse couleur des canaux $\{30,20,10\}$ de la scène AVIRIS, b) résultat obtenu par PP/MV, c) PP/ABmg (2 bandes de texture), d) ACP/Cooc.

Influence de l'heuristique dans l'estimation de paramètres

Dans cette image AVIRIS, il existe des différences assez notables de moyennes entre les classes, notamment entre l'eau et les autres classes. Pour cette raison, il peut être intéressant de passer par l'heuristique de détection des contours dans la phase d'estimation des paramètres dans la fenêtre glissante, en ce qui concerne l'algorithme $PP/ABmg$. L'utilisation de

cette euristique n'apporte pas de bénéfice à la classification des zones urbaines réelles, mais permet d'éviter un certain nombre de fausses alarmes. En effet, la présence d'une frontière entre deux textures ayant des moyennes éloignées fait que les paramètres de texture sont très élevés, et peuvent dans certains cas être confondus avec ceux de zones urbaines. Ceci est illustré dans la figure 5.21, où l'on donne la différence entre le résultat de l'algorithme *PP/ABmg* avec et sans l'euristique, pour la partie de l'image précédente où les moyennes des classes sont éloignées. Mais de façon générale, le gain en classification est assez faible comparativement à l'augmentation du temps de calcul.



FIG. 5.21 – a) Zoom sur une partie de la classification par *PP/ABmg* sans euristique b) *PP/ABmg* avec euristique.

5.5 Conclusion

Ce chapitre est consacré aux applications des modèles et algorithmes présentés aux chapitres précédents. On présente deux algorithmes globaux de classification de texture d'images hyperspectrales. Le premier consiste, dans un premier temps, à réduire la dimension de l'espace d'observation, puis à effectuer la classification, les deux phases étant fondées sur la modélisation par MGMRF décrite au chapitre 2. En effet, l'indice de projection à maximiser est la distance de Bhattacharyya entre les classes prédéfinies calculée sur le modèle, et la classification est effectuée directement en comparant la vraisemblance des classes, avec le modèle markovien gaussien comme hypothèse. Le deuxième algorithme permet une prise en compte équitable des caractéristiques radiométriques et texturales des images considérées, car il consiste, dans un premier temps, en l'extraction de caractéristiques radiométriques par projection des données dans un sous-espace, en parallèle de l'extraction de paramètres de texture calculés en faisant l'hypothèse d'un modèle MGMRF. Une fois ces deux images extraites, on effectue la classification par maximum de vraisemblance, grâce à une approximation de la loi conditionnelle des paramètres de texture. Il s'avère que ce deuxième algorithme donne de meilleurs résultats de classification sur des images réelles, il

laisse une importance plus équilibrée aux notions de texture et de radiométrie.

C'est ce que soulignent les tests de classification sur des images AVIRIS dont on rend compte dans la dernière partie de ce chapitre. Ces tests montrent d'une part, l'utilité de la prise en compte de la texture dans des algorithmes qui traitent d'images urbaines et d'autre part, la compétitivité de notre approche par champ de Markov par rapport à une méthode classique, telle que celle fondée sur les matrices de cooccurrence. En effet, les différences de classification entre ces deux derniers algorithmes sont, dans les cas étudiés, soit assez grandes (pour les images réelles), soit très faibles (pour les mosaïques de textures).

Conclusion et perspectives

Dans ce manuscrit, on a présenté des outils contribuant à l'analyse conjoint spatiale et spectrale des textures appliquée aux images composées d'un grand nombre de canaux, dites images hyperspectrales. Les deux principales contributions de ce travail concernent la modélisation de la texture, et la réduction d'espace. Des tests ont été réalisés sur des images AVIRIS dans le cadre d'une application de classification supervisée pour l'analyse de l'urbain.

Modélisation

Le premier point présenté dans ce manuscrit est la modélisation des textures fines par champ de Markov gaussien multivarié. Ce modèle est la version vectorielle du même champ scalaire, pour des données multispectrales. On a adapté ce modèle aux images hyperspectrales en le simplifiant, de manière à éviter les problèmes que procurent les espaces de grande dimension en termes d'estimation statistique. Cette modélisation prend en compte les relations et interactions au niveau spectral, à travers la *matrice de covariance*, et au niveau spatial, à travers les *paramètres d'interaction*. Les estimateurs qui sont associés au modèle modifié sont plus simples, notamment quand il faut passer par de l'optimisation numérique, car le nombre de variables est diminué de façon drastique.

Après avoir précisé dans quelles conditions les hypothèses simplificatrices sont réalistes, on propose un algorithme de partition du spectre en groupes de bandes homogènes, permettant de se trouver dans ces conditions.

Les simulations menées d'après ce modèle montrent que celui-ci est capable de saisir l'organisation de textures fines, en revanche les macro-textures sont moins bien modélisées, car aucune notion structurelle n'est prise en compte dans le modèle. Considérer des voisinages plus étendus pourrait être une solution, mais un obstacle devra d'abord être franchi : la faible robustesse des estimateurs quand le nombre de paramètres augmente. En effet, l'augmentation du voisinage rend les domaines de définition des paramètres beaucoup plus

complexes, et par conséquent l'optimisation numérique également. Quant aux estimateurs linéaires, ils reposent sur des mesures de corrélation qui sont de moins en moins fiables, à mesure que le voisinage s'étend. Pourtant, les paramètres d'interaction sont directement liés aux corrélations du champ, ce qui justifierait peut-être l'emploi de tables de correspondance entre corrélation et interaction (ceci réduirait le temps de calcul, mais augmenterait la capacité de mémoire nécessaire).

Réduction d'espace

La réduction de dimension est un autre aspect du problème de traitement d'images hyperspectrales que l'on a abordé. Le besoin de réduction d'espace provient principalement de ce que l'estimation de paramètres est parfois problématique dans les espaces de grande dimension.

Une méthode fondée sur l'analyse de mélange linéaire a été proposée pour la réduction de dimension. Celle-ci permet une réduction de dimension non par projection linéaire dans un sous-espace, mais par projection sur un simplexe de l'espace. Les sommets de ce simplexe sont les composantes spectrales du mélange, et sont ajustés itérativement selon la projection des données sur le simplexe. Cet algorithme permet d'avoir une réduction d'espace proche de l'analyse en composantes principales, excepté que les coordonnées dans l'espace final sont les proportions d'un mélange (de somme unité), et que les composantes spectrales ont un sens physique.

L'autre partie du travail de recherche concernant la réduction d'espace consiste en une adaptation de l'algorithme séquentiel de poursuite de projection proposé par Jimenez et Landgrebe. Celui-ci vise à déterminer un sous-espace dans lequel un indice de projection est optimisé. Dans la mise en œuvre approchée, l'espace des bandes spectrales est divisé en groupes de bandes desquels on extrait un vecteur de la base finale. L'adaptation proposée diffère sur deux plans : le découpage en groupes de bandes n'est plus effectué de façon uniforme (même nombre de bandes dans chaque groupe), mais adapté pour mieux répartir l'indice de projection sur les groupes, et l'indice de projection découle de la modélisation de texture.

Cette mise en œuvre reste donc approchée, puisque l'espace de recherche de sous-espace est fortement réduit par le découpage. L'algorithme par croissance d'espace permet d'obtenir un meilleur sous-espace, mais le temps de calcul augmente très sérieusement avec le nombre de bandes initial.

Applications

Cette approche d'analyse de texture a été testée dans le cadre applicatif de la classification supervisée d'images AVIRIS en milieu urbain. Pour cela, deux chaînes de classification ont été mises au point. La première est constituée d'une phase de réduction de dimension par l'algorithme de poursuite de projection proposé, puis d'une classification par le critère du maximum de vraisemblance calculé selon le modèle de texture. Cette classification est régularisée, soit par une prise en compte d'une fenêtre autour du pixel pour la classification de celui-ci, soit par l'introduction d'un modèle *a priori* de Potts. Le deuxième algorithme consiste en l'extraction de bandes radiométriques par la poursuite de projection de Jimenez et Landgrebe, et de bandes dites *de texture* représentant le paramètre de covariance calculé sur une fenêtre glissante. Ces deux types de bandes sont concaténés pour former une nouvelle image classifiée par maximum de vraisemblance, grâce à une approximation de la loi des paramètres de texture.

Les résultats de classification par ces deux algorithmes sont comparés aux résultats donnés par des méthodes classiques, l'une prenant en compte la texture, l'autre non. On constate que les algorithmes contextuels apportent une aide à la classification, en particulier en présence de classes dont les moyennes sont proches. Le premier algorithme ne s'avère efficace que lorsque les données suivent d'assez près l'hypothèse markovienne gaussienne, sans irrégularités. Le calcul du critère de vraisemblance pour la classification sur une fenêtre glissante permet d'intégrer des informations sur un voisinage plus étendu, ce qui améliore un peu la classification. Le problème de cet algorithme est que très peu d'importance est laissée au potentiel discriminant de la radiométrie (par les moyennes du champ). Or dans des scènes mélangeant zones urbaines et péri-urbaines, ou rurales, il est fréquent que certaines zones se caractérisent simplement par leur spectre de réflectance (zones de cultures, eau. . .).

De façon générale, le deuxième algorithme fournit les meilleurs résultats. Sur les images réelles AVIRIS, notamment, on constate que l'ajout de quelques bandes de texture apporte rapidement une meilleure qualité de classification. Cette méthode paraît donc prometteuse, mais quelques améliorations pourraient y être apportées. D'abord, on a vu que la présence de transitions abruptes à l'intérieur de la fenêtre glissante perturbait sensiblement l'estimation des paramètres. L'heuristique proposée apporte un plus à ce niveau en proposant de partitionner le nuage de points en deux classes avant estimation, cependant elle est lourde à mettre en œuvre et moyennement efficace. Pour améliorer cette heuristique, il faudrait intégrer des considérations spatiales dans la partition, ce qui améliorerait vraisemblablement les résultats, mais aboutirait à des temps de calcul encore plus élevés.

Un autre problème de la méthode proposée est l'absence d'une estimation de certains

des paramètres entrant en compte dans la classification : nombre de bandes de radiométrie et de texture à garder, nombre de degrés de liberté dans l'approximation de la loi de la covariance conditionnelle. Si pour ce dernier paramètre, les méthodes d'estimation statistiques classiques peuvent s'appliquer (ie. MV), le choix des premiers est délicat. Une possibilité pourrait être d'étudier l'évolution d'un critère (par exemple de distance, ou d'énergie), en fonction du nombre de bandes final.

Les résultats fournis sur les mosaïques de textures par la méthode des paramètres de co-occurrence étant bons, il semble que cette méthode, malgré son échec sur les images réelles, ait un potentiel intéressant. Une combinaison du deuxième algorithme et des matrices de cooccurrence, associée à une étude plus précise des paramètres extraits de ces matrices, pourrait apporter une solution intéressante au problème, en combinant les apports de ces deux sources différentes.

Enfin, ces applications permettent de tirer des enseignements sur l'utilisation des données hyperspectrales pour améliorer l'analyse de la texture. L'analyse de texture appliquée aux zones urbaines sur ces données est de bonne qualité, confirmant l'utilité de travailler sur un nombre de bandes supérieur aux nombres habituels des images multispectrales. Néanmoins, il semble que l'utilisation de quelques centaines de bandes soit une perte de temps de calcul, en ce qui concerne les problèmes traités. En effet, les courbes de classification montrent toutes un palier au-delà duquel le taux de bonne classification reste quasiment constant. Ce palier correspond à une dimension (entre 10 et 20 bandes) permettant de bien caractériser les données, que l'on pourra appeler dimension intrinsèque, et celle-ci semble suffisante pour les applications que l'on vise. On peut même penser que cette dimension, ainsi que les bandes obtenues par la réduction de dimension, pourraient être utiles lors de la conception de nouveaux capteurs superspectraux ou hyperspectraux.

Annexe A

Propriétés de la matrice A

A.1 Valeurs propres et déterminant

Pour calculer les valeurs propres et le déterminant de la matrice A du MGMRF, on peut par exemple exploiter sa structure de matrice circulante (donc de convolution).

On commence par le cas où la dimension spectrale est égale à un, puis on étend le résultat aux dimensions supérieures.

A.1.1 Cas monospectral

Prenons comme exemple un modèle anisotrope en 4-connexité. On construit A comme suit :

$$B = \text{circulant}(1, -a_h, 0, \dots, 0, -a_h) \quad (\text{A.1})$$

$$A = \text{circulant}(B, -a_v I_{N_h}, 0, \dots, 0, -a_v I_{N_h}) \quad (\text{A.2})$$

Cette matrice correspond à un masque de convolution \mathcal{A} de la forme :

$$\begin{array}{|c|c|c|} \hline & -a_v & \\ \hline -a_h & 1 & -a_h \\ \hline & -a_v & \\ \hline \end{array}$$

Ce masque de convolution correspond à un voisinage en 4-connexité anisotrope. Pour tout autre système de voisinage, il est aisé de trouver le masque de convolution \mathcal{A} correspondant.

Soit \mathcal{X} une image de taille $N_v \times N_h$, rangée en un vecteur colonne X , les lettres calligraphiées représentant des images sous forme de matrices (N_h colonnes, N_v lignes) et les autres

leur pendant vectoriel ($N_h N_v$ lignes, 1 colonnes). On note \mathcal{V} l'opérateur transformant une image en un vecteur, en rangeant les pixels de la première dans l'ordre lexicographique, et \mathcal{V}^{-1} l'opérateur inverse.

$$AX = \mathcal{V}(\mathcal{A} * \mathcal{X}) \quad (\text{A.3})$$

Si l'on calcule la transformée de Fourier de chaque membre :

$$FAX = \mathcal{V}(\tilde{\mathcal{A}} \times \tilde{\mathcal{X}}) \quad (\text{A.4})$$

où F désigne l'opérateur matriciel de la transformation de Fourier, $\tilde{\mathcal{X}}$ la transformée de Fourier discrète 2D de la grandeur X , et le signe \times entre deux matrices désigne un produit point à point.

Puisque l'on a un produit point à point, on peut dire que :

$$\mathcal{V}(\tilde{\mathcal{A}} \times \tilde{\mathcal{X}}) = \Delta \tilde{\mathcal{X}} = \Delta FX \quad (\text{A.5})$$

où Δ est une matrice diagonale. Par conséquent :

$$FAX = \Delta FX \quad (\text{A.6})$$

Et ceci quel que soit X , donc $FA = \Delta F$, et $|A| = |\Delta|$. Les éléments de Δ sont donc les valeurs propres de A .

$$\lambda_{s_1, s_2} = \sum_{m=0, n=0}^{N_h, N_v} A_{m, n} \exp\left(2\pi i \left(\frac{s_1 m}{N_h} + \frac{s_2 n}{N_v}\right)\right) \quad (\text{A.7})$$

où les $A_{m, n}$ sont les coefficients du masque de convolution \mathcal{A} . Pour connaître $|A|$, il suffit donc de faire le produit des termes de la transformée de Fourier du masque \mathcal{A} :

$$|A| = \prod_{x=0, y=0}^{N_h, N_v} \sum_{m=0, n=0}^{N_h, N_v} A_{m, n} \exp\left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v}\right)\right) \quad (\text{A.8})$$

A.1.2 Cas multispectral

Le cas multispectral, s'agissant du modèle proposé, est très peu différent du cas monospectral. En effet, dans le cas où la dimension spectrale est supérieure à 1, et que les paramètres d'interaction sont de la forme $\theta = aI_{N_b}$, la matrice A est de la forme suivante :

$$A = A' \otimes I_{N_b} \quad (\text{A.9})$$

Où \otimes désigne le produit de Kronecker et A' est exactement de la même forme que la matrice A du paragraphe précédent.

Les valeurs propres viennent alors immédiatement, puisque ce sont les mêmes que précédemment, mais ce sont toutes des valeurs propres d'ordre N_b , et le déterminant est :

$$|A| = |A'|^{N_b} \quad (\text{A.10})$$

A.2 Dérivée

L'expression du logarithme du déterminant de A est donc, dans le cas monospectral :

$$\ln |A| = \sum_{x=0, y=0}^{N_h, N_v} \ln \left(\sum_{m=0, n=0}^{N_h, N_v} A_{m,n} \exp \left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v} \right) \right) \right) \quad (\text{A.11})$$

Or $A_{m,n} = -a_{m,n}$. De plus, on sait que $a_{m,n}$ et $a_{-m,-n}$ sont égaux, et que $a_{0,0} = 1$. Donc, on peut écrire :

$$\ln |A| = \sum_{x=0, y=0}^{N_h, N_v} \ln \left(1 - \sum_{m=1, n=1}^{\frac{N_h}{2}, \frac{N_v}{2}} 2a_{m,n} \cos \left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v} \right) \right) \right) \quad (\text{A.12})$$

Par conséquent, la dérivée peut s'écrire :

$$\frac{\partial}{\partial a_{m,n}} \ln |A| = \sum_{x=0, y=0}^{N_h, N_v} \frac{-2 \cos \left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v} \right) \right)}{1 - \sum_{m=1, n=1}^{\frac{N_h}{2}, \frac{N_v}{2}} 2a_{m,n} \cos \left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v} \right) \right)} \quad (\text{A.13})$$

C'est-à-dire :

$$\frac{\partial}{\partial a_{m,n}} \ln |A| = -C_{m,n} \quad (\text{A.14})$$

où $C_{m,n}$ est le coefficient de corrélation théorique liée au modèle, car il s'agit de $(A^{-1})_{m,n}$. En effet, les $(A^{-1})_{m,n}$ forment la transformée de Fourier inverse des $\frac{1}{\lambda_{s_1, s_2}}$, c'est-à-dire l'inverse de A .

Dans le cas multispectral, puisque $|A| = |A'|^{N_b}$, alors $\ln |A| = N_b \ln |A'|$, et sa dérivée est égale à N_b fois celle de son équivalent monospectral.

$$\frac{\partial}{\partial a_{m,n}} \ln |A| = -N_b C_{m,n} \quad (\text{A.15})$$

A.3 Matrice B du champ 3D

En ce qui concerne la matrice B du champ de Markov gaussien 3D, la démarche pour en trouver les valeurs propres et le déterminant est similaire que pour le MGMRF. En effet, il s'agit du même type de matrice que la matrice A .

Pour un modèle à 6 voisins, la matrice B correspond à un masque de convolution \mathcal{B} qui est l'extension 3D de celui décrit en A.1.1.

On peut donc, de même que précédemment, calculer les valeurs propres de cette matrice en passant par la transformée de Fourier en 3 dimensions.

$$\lambda_{s_1, s_2, s_3} = \sum_{m=0, n=0, p=0}^{N_h, N_v, N_b} B_{m, n, p} \exp \left(2\pi i \left(\frac{s_1 m}{N_h} + \frac{s_2 n}{N_v} + \frac{s_3 p}{N_b} \right) \right) \quad (\text{A.16})$$

où les $B_{m, n, p}$ sont les coefficients du masque de convolution \mathcal{B} . Le déterminant s'obtient en faisant le produit des valeurs propres.

Par un calcul similaire à celui fait au paragraphe précédent, on peut également déterminer la dérivée du logarithme du déterminant de B :

$$\frac{\partial}{\partial b_{m, n, p}} \ln |B| = \sum_{x=0, y=0, z=0}^{N_h, N_v, N_b} \frac{-2 \cos \left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v} + \frac{zp}{N_b} \right) \right)}{1 - \sum_{m=1, n=1, p=1}^{\frac{N_h}{2}, \frac{N_v}{2}, \frac{N_b}{2}} 2b_{m, n, p} \cos \left(2\pi i \left(\frac{xm}{N_h} + \frac{yn}{N_v} + \frac{zp}{N_b} \right) \right)} \quad (\text{A.17})$$

C'est-à-dire :

$$\frac{\partial}{\partial b_{m, n, p}} \ln |B| = -C_{m, n, p} \quad (\text{A.18})$$

où $C_{m, n, p}$ est le coefficient de corrélation théorique liée au modèle, c'est-à-dire $(B^{-1})_{m, n, p}$.

Bibliographie

- [1] T.W Anderson. *Introduction to multivariate statistical analysis*. J. Wiley and Sons, 1958.
- [2] S.K. Babey et C.D. Anger. A Compact Airborne Spectrographic Imager (CASI). In *Proceedings of IGARSS*, volume 2, pages 1028–1031, Vancouver, 1989.
- [3] N. Balram et J. M. F. Moura. Noncausal gauss-markov random fields: Parameter structure and estimation. *IEEE Transactions on Information Theory*, 39:1333–1355, Juillet 1993.
- [4] M. Basseville. Distance measures for signal processing and pattern recognition. *Signal Processing*, 18(4):349–369, Décembre 1989.
- [5] J. Bennett et A. Khotanzad. Maximum likelihood estimation methods for multispectral random field image models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(6):537–543, 1999.
- [6] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc. B*, 36:192–236, 1974.
- [7] J. Besag. On the statistical analysis of dirty pictures. *J. Roy Stat. Soc. B*, 48(3):259–302, 1986.
- [8] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–110, 1943.
- [9] A. Bovik, M. Clark et W. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(1):55–73, 1990.
- [10] F. Campbell et J. Robson. Application to Fourier analysis to the visibility of gratings. *J. Physiology*, 197(551-566), 1968.
- [11] C-I. Chang, H. Ren et S-S. Chiang. Real-time processing algorithms for target detection and classification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 39(4):760–768, Avril 2001.
- [12] C-I. Chang, X-L. Zhao et M.L.G. Althouse. Least square subspace projection ap-

- proach to mixed pixel classification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 36(3):898–912, Mars 1998.
- [13] T. Chang et C.C.J Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing (IP)*, 2(4):1376–1387, Octobre 1993.
- [14] B.B. Chaudhuri et N. Sarkar. Texture segmentation using fractal dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(1):72–77, Janvier 1995.
- [15] R. Chellappa, S. Chatterjee et R. Bagdazian. Texture synthesis and compression using Gauss-Markov random fields. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(2):298–303, Mars-Avril 1985.
- [16] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.*, 23:493–507, 1952.
- [17] D.A. Clausi et M.E. Jernigan. Designing Gabor filters for optimal texture separability. *Pattern Recognition*, 33(11):1835–1849, Novembre 2000.
- [18] R.W. Connors, M.M. Trivedi et C.A. Harlow. Segmentation of a high-resolution urban scene using texture operators. *Computer Vision Graphics Image Processing*, 25(3):273–310, Mars 1984.
- [19] T.M. Cover et P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, Janvier 1967.
- [20] G.R. Cross et A.K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(1):25–39, Janvier 1983.
- [21] M. Dekking, J. Levy Vehel, E. Lutton et C. Tricot. *Fractals : theory and applications in engineering*. Springer, 1999.
- [22] H. Derin et H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 9:721–741, 1987.
- [23] X. Descombes. *Champs markoviens en analyse d'images*. Thèse de Doctorat, ENST Paris, 1993.
- [24] X. Descombes, R. Morris et J. Zerubia. Quelques améliorations à la segmentation d'image bayésienne. *Traitement du Signal*, 14(4):373–382, 1997.
- [25] X. Descombes, M. Sigelle et F. Prêteux. Estimating Gaussian Markov random field parameters in a nonstationary framework : application to remote sensing imaging. *IEEE Transactions on Image Processing (IP)*, 8(4):490–503, 1999.
- [26] D. Dubuc, S.W. Zucker, C. Tricott, J.F. Quiniou et D. Wehbi. Evaluating the fractal dimension of surfaces. *Proc. Roy. Soc. London A*, 425:113–127, 1989.

-
- [27] G. H. Dunteman. *Principal Components Analysis*. Sage Inc., 1989.
- [28] *ENVI, User's guide*.
- [29] J.H. Friedman. Regularized discriminant analysis. *J. Am. Statistical Assoc.*, 84:165–175, Mars 1989.
- [30] J.H. Friedman et J.W Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 23:881–889, Septembre 1974.
- [31] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, 2eme edition, 1990.
- [32] S. Geman et D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(6):721–741, Novembre 1984.
- [33] S. Geman et D. Geman. A stochastic algorithm for global optimisation with constraints. Rapport de recherche, Brown University, 1987.
- [34] G.L. Gimel'farb. *Image texture and Gibbs random fields*. Kluwer, 1999.
- [35] A.A. Green, M. Berman, P. Switzer et M.D. Craig. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 26(1):65–74, 1988.
- [36] R.O. Green, D.A. Roberts et J.E. Conel. Characterization and compensation of the atmosphere for the inversion of AVIRIS calibrated radiance to apparent surface reflectance. In *Summaries of the Sixth Annual JPL Airborne Earth Science Workshop*, pages 135–146, Mars 1996.
- [37] P. Hall et K. Li. On almost linearity of low dimensionality projections from high dimensional data. *The Annals of Statistics*, 21(2):867–889, 1993.
- [38] R. Haralick, K. Shanmugan et I. Dinstein. Textural features for images classification. *IEEE Transactions on Systems, Man and Cybernetic*, 3:610–621, 1973.
- [39] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [40] J.C. Harsanyi et C-I Chang. Hyperspectral image classification and dimensionality reduction : an orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 32(4):779–785, 1994.
- [41] G. G. Hazel. Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 38(3):1199–1211, 2000.
- [42] J.P. Hoffbeck et D.A. Landgrebe. Covariance matrix estimation and classification with limited training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(7):763–767, Juillet 1996.

- [43] C. Huang, L.S. Davis et J.R.G. Townshend. An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23(4):725–749, Février 2002.
- [44] D.H. Hubel. *Eye, brain and vision*. Freeman, 1988.
- [45] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.
- [46] G.F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, 1968.
- [47] M. Idrissa et Marc Acheroy. Texture classification using Gabor filters. *Pattern Recognition Letters*, 23(9):1095–1102, Juillet 2002.
- [48] A. K. Jain. *Fundamentals of image processing*, chapter 2: Two-dimensional systems and mathematical preliminaries. Prentice Hall, 1989.
- [49] A. K. Jain et F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [50] X. Jia et J. A. Richards. Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 37(1):538–542, 1999.
- [51] Q. Jiang. Automatic extraction of urban regions from multispectral SPOT satellite imagery. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Vancouver, 2000.
- [52] L. O. Jimenez et D. A. Landgrebe. Hyperspectral data analysis and supervised feature reduction via projection pursuit. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 37(6):2653–2667, 1999.
- [53] T. Kailath. On a measure of divergence between two statistical populations defined by their probability distributions. *IEEE Transactions on Communication Technology*, 15(1):52–60, Février 1967.
- [54] T. Kasparis, D. Charalampidis, M. Georgiopoulos et J. Rolland. Segmentation of textured images based on fractals and image filtering. *Pattern Recognition*, 34(10):1963–1973, Octobre 2001.
- [55] J.M. Keller, S. Chen et R.M. Crownover. Texture description and segmentation through fractal geometry. *Computer Vision Graphics Image Processing*, 45(2):150–166, Février 1989.
- [56] C. Kervrann et F. Heitz. A Markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics. *IEEE Transactions on Image Processing (IP)*, 4(6):856–862, Juin 1995.
- [57] P. Kruizinga et N. Petkov. Nonlinear operator for oriented texture. *IEEE Transactions on Image Processing (IP)*, 8(10):1395–1407, 1999.

-
- [58] A. Laine et J. Fan. Texture classification by wavelet packet signature. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(11):1186–1191, Novembre 1993.
- [59] C. Lee et E. Choi. Bayes error evaluation of the Gaussian ML classifier. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 38(3):1471–1475, 2000.
- [60] T.M. Lillesand et R.W. Kiefer. *Remote sensing and image interpretation*. J. Wiley and Sons, 3eme edition, 1979.
- [61] C-T Lin, Y-C Lee et H-C Pu. Satellite sensor image classification using cascaded architecture of neural fuzzy network. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 38(2):1033–1043, Mars 2000.
- [62] A. Lorette, X. Descombes et J. Zerubia. Urban areas extraction based on texture analysis through a Markovian modelling. *International Journal of Computer Vision (IJCV)*, 36(3):219–234, 2000.
- [63] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2eme edition, 1999.
- [64] N. Metropolis, A.W. Rosenbluth, N.M. Rosenbluth, A.H. Teller et E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.
- [65] J. M. F. Moura et N. Balram. Recursive structure of noncausal gauss-markov random fields. *IEEE Transactions on Information Theory*, 38:334–354, Mars 1992.
- [66] D. Nuzillard et A. Bijaoui. Blind source separation and analysis of multispectral astronomical images. *Astronomy and Astrophysics*, 147(1):129–138, Novembre 1993.
- [67] D. K. Panjwani et G. Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(10):939–954, 1995.
- [68] A.P. Pentland. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(6):661–674, Novembre 1984.
- [69] M. Rast et J.L Besy. The ESA Medium Resolution Imaging Spectrometer MERIS a review of the instrument and its mission. *International Journal of Remote Sensing*, 20(9):1681–1702, 1999.
- [70] G. Reulier, X. Descombes, F. Falzon et J. Zerubia. La poursuite de projection pour la classification d’image hyperspectrale texturée. Rapport de Recherche 4152, INRIA, Mars 2001.
- [71] G. Reulier, X. Descombes, F. Falzon et J. Zerubia. Analyse de texture hyperspectrale par modélisation markovienne. Rapport de Recherche 4479, INRIA, Juin 2002.
- [72] G. Reulier, X. Descombes, F. Falzon et J. Zerubia. A Gauss-Markov model for hyper-

- spectral texture analysis of urban areas. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Québec City, Août 2002.
- [73] S. M. Schweizer et J. M. F. Moura. Hyperspectral imagery: Clutter adaptation in anomaly detection. *IEEE Transactions on Information Theory*, 46(5):1855–1871, 2000.
- [74] S. Singh, J. Haddon et M. Markou. Nearest-neighbour classifiers in natural scene analysis. *Pattern Recognition*, 34(8):1601–1612, Août 2001.
- [75] B. Thai et G. Healey. Invariant subpixel material detection in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing (GRS)*, 40(3):599–608, Mars 2002.
- [76] S. Theodoridis et K. Koutroumbas. *Pattern Recognition*. Academic Press, New York, 1999.
- [77] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing (IP)*, 4(11):1549–1560, Novembre 1995.
- [78] P.J.M. van Laarhoven et E.H.L. Aarts. *Simulated annealing: theory and applications*. D. Reidel, 1987.
- [79] G. Vane, R.O. Green, T.G. Chrien, H.T. Enmark, E.G. Hansen et W.M. Porter. The airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of the Environment*, 44:127–143, 1993.
- [80] V. Vapnik. *The nature of statistical learning*. Springer, 2^e édition, 1999.
- [81] M. Vetterli et J. Kovacevic. *Wavelets and subband coding*. Prentice Hall, 1995.
- [82] G. Winkler. *Image analysis, random fields and dynamic Monte Carlo methods: a mathematical introduction*. Springer, 1995.
- [83] F.Y. Wu. The Potts model. *Reviews of modern physics*, 54(1):235–267, 1982.
- [84] J. Zhang, Y. Zhang et T. Zhou. Classification of hyperspectral data using support vector machine. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Thessaloniki, 2001.

Publications de G. Rellier

Journaux

- G. Rellier, X. Descombes, F. Falzon et J. Zerubia. La poursuite de projection pour la classification d'images hyperspectrales texturées. Soumis à *Traitement du Signal*, Juin 2001.
- G. Rellier, X. Descombes et J. Zerubia. Local registration and deformation of a road cartographic database on a SPOT Sattelite Image. *Pattern Recognition*, 35(10): 2213-222, Octobre 2002.

Conférences internationales

- G. Rellier, X. Descombes, F. Falzon et J. Zerubia. A Gauss-Markov model for hyperspectral texture analysis of urban areas. *Proc. of the International Conference on Pattern Recognition (ICPR)*, Québec City, Août 2002.
- G. Rellier, X. Descombes et J. Zerubia. Deformation of a cartographic road network on a SPOT sattelite image. *Proc. of the IEEE International Conference on Image Processing (ICIP)*, Vancouver, Septembre. 2000.

Conférences nationales

- G. Rellier, X. Descombes, F. Falzon et J. Zerubia. Un modèle markovien gaussien pour l'analyse de texture hyperspectrale en milieu urbain. *GRETSI*, Toulouse, Septembre 2001.
- G. Rellier, X. Descombes, F. Falzon et J. Zerubia. La poursuite de projection pour la classification d'images hyperspectrales texturées. *ORASIS*, Cahors, Juin 2001.

Rapports de recherche

- G.Rellier, X. Descombes, F. Falzon, J. Zerubia. Analyse de texture hyperspectrale par modélisation markovienne, Rapport de Recherche INRIA No 4479, Juin 2002.
- G. Rellier, X. Descombes, F. Falzon et J. Zerubia. La poursuite de projection pour la classification d'images hyperspectrales texturées. Rapport de Recherche INRIA No 4152, Mars 2001.
- G.Rellier, X. Descombes, J. Zerubia. Local registration and deformation of a road cartographic database on a SPOT satellite image, Rapport de Recherche INRIA No 3939, Mai 2000

RÉSUMÉ

Dans cette thèse, on aborde le problème de l'analyse de texture pour l'étude des zones urbaines.

La texture est une notion spatiale désignant ce qui, en dehors de la couleur ou du niveau de gris, caractérise l'homogénéité visuelle d'une zone donnée d'une image. Le but de cette étude est d'établir un modèle qui permette une analyse de texture prenant en compte conjointement l'aspect spatial et l'aspect spectral, à partir d'images hyperspectrales. Ces images sont caractérisées par un nombre de canaux largement supérieur à celui des images multispectrales classiques. On désire tirer parti de l'information spectrale pour améliorer l'analyse spatiale.

Les textures sont modélisées par un champ de Markov gaussien vectoriel, qui permet de prendre en compte les relations spatiales entre pixels, mais aussi les relations inter-bandes à l'intérieur d'un même pixel. Ce champ est adapté aux images hyperspectrales par une simplification évitant l'apparition de problèmes d'estimation statistique dans des espaces de grande dimension.

Dans le but d'éviter ces problèmes, on effectue également une réduction de dimension des données grâce à un algorithme de poursuite de projection. Cet algorithme permet de déterminer un sous-espace de projection dans lequel une grandeur appelée indice de projection est optimisée. L'indice de projection est défini par rapport à la modélisation de texture proposée, de manière à ce que le sous-espace optimal maximise la distance entre les classes prédéfinies, dans le cadre de la classification.

La méthode d'analyse de texture est testée dans le cadre d'une classification supervisée. Pour ce faire, on met au point deux algorithmes que l'on compare avec des algorithmes classiques utilisant ou non l'information de texture. Des tests sont réalisés sur des images hyperspectrales AVIRIS.

Mots clefs: Images hyperspectrales, analyse de texture, classification, champs de Markov, réduction d'espace, poursuite de projection.

ABSTRACT

In this work, we investigate the problem of texture analysis of urban areas.

Texture is a spatial concept that refers to the visual homogeneity characteristics of an image, not taking into account color or grey level. The aim of this research is to define a model which allows a joint spectral and spatial analysis of texture, and then to apply this model to hyperspectral images. These images have many more bands than classical multispectral images. We intend to make use of spectral information and improve simple spatial analysis.

Textures are modeled by a vectorial Gauss-Markov random field, which allows us to take into account the spatial interactions between pixels as well as inter-band relationships for a single pixel. This field has been adapted to hyperspectral images by a simplification which avoids statistical estimation problems common to high dimensional spaces. In order to avoid these problems, we also reduce the dimensionality of the data, using a projection pursuit algorithm. This algorithm determines a projection subspace in which an index, called projection index, is optimized. This index is defined in relation to the proposed texture model so that, when a classification is being carried out, the optimal subspace maximizes the distance between predefined training samples.

This texture analysis method is tested within a supervised classification framework. For this purpose, we propose two classification algorithms that we compare to two classical algorithms, one which uses texture information and one which does not. Tests are carried out on AVIRIS hyperspectral images.

Keywords: Hyperspectral images, texture analysis, classification, Markov random fields, space dimension reduction, projection pursuit.