



**HAL**  
open science

## Analyse de texture hyperspectrale par modélisation markovienne

Guillaume Rellier, Xavier Descombes, Frédéric Falzon, Josiane Zerubia

► **To cite this version:**

Guillaume Rellier, Xavier Descombes, Frédéric Falzon, Josiane Zerubia. Analyse de texture hyperspectrale par modélisation markovienne. RR-4479, INRIA. 2002. inria-00072109

**HAL Id: inria-00072109**

**<https://inria.hal.science/inria-00072109>**

Submitted on 23 May 2006

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

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

*Analyse de texture hyperspectrale par modélisation  
markovienne*

Guillaume Rellier — Xavier Descombes — Frédéric Falzon — Josiane Zerubia

**N° 4479**

Juin 2002

THÈME 3



*Rapport  
de recherche*



## **Analyse de texture hyperspectrale par modélisation markovienne**

Guillaume Rellier , Xavier Descombes , Frédéric Falzon \* , Josiane Zerubia

Thème 3 —Interaction homme-machine,  
images, données, connaissances  
Projet Ariana

Rapport de recherche n° 4479 —Juin 2002 —51 pages

**Résumé :** L'analyse de texture est l'objet de nombreuses recherches dans le domaine de l'imagerie mono et multispectrale. En parallèle, sont apparus ces dernières années de nouveaux instruments spectro-imageurs ayant un grand nombre de canaux (supérieur à 10), fournissant des images appelées *hyperspectrales* qui sont une représentation du paysage échantillonnée à la fois spatialement et spectralement. Le but de ce travail est de réaliser une analyse de texture qui se déroule conjointement dans ces deux espaces discrets. Pour ce faire, on utilise une modélisation probabiliste vectorielle de la texture via un champ de Markov gaussien. Les paramètres de ce champ permettent la caractérisation de différentes textures présentes dans les images hyperspectrales. L'application visée dans cette étude étant la classification du tissu urbain, qui est mal caractérisée par la seule radiométrie, on utilise ces paramètres comme de nouvelles bandes afin d'effectuer la classification par le critère du Maximum de Vraisemblance. Les résultats sur des images AVIRIS montrent une nette amélioration de la classification due à l'utilisation de l'information de texture.

**Mots-clés :** imagerie hyperspectrale, classification, paramètres de texture, champs de Markov.

\* Alcatel Space Industries, Cannes

## **Hyperspectral texture analysis using a Gauss-Markov model**

**Abstract:** Texture analysis has been widely investigated in monospectral and multispectral imagery domain. In the same time, new image sensors with a large number of bands (more than 10) have been designed. They are able to provide images with both fine spectral and spatial sampling, called *hyperspectral images*. The aim of this work is to perform a joint texture analysis in both discrete spaces. To achieve this goal, we have a probabilistic vectorial texture modeling, with Gauss-Markov Random Field. The MRF parameters allow for the characterisation of different hyperspectral textures. A likely application of this work being the classification of urban areas, which are not well characterized by radiometry alone, we use these parameters as new features is a Maximum Likelihood classification algorithm. The results obtain on AVIRIS hyperspectral images show better classifications when using texture information.

**Key-words:** hyperspectral image processing, classification, texture features, Markov random fields.

## Table des matières

|  |           |
|--|-----------|
| <b>Introduction</b>  | <b>5</b>  |
| <b>1 Modèle markovien gaussien de texture</b>                                      | <b>7</b>  |
| 1.1 Modèles markoviens . . . . .   | 7         |
| 1.2 Modèle markovien gaussien . . . . .  | 8         |
| 1.3 MGMRF . . . . .  | 8         |
| 1.3.1 Loi conditionnelle . . . . .   | 8         |
| 1.3.2 Loi globale . . . . .  | 10        |
| 1.3.3 Caractéristiques des espaces de grande dimension . . . . .                   | 11        |
| 1.3.4 Domaine de validité des paramètres . . . . .                                 | 12        |
| 1.3.5 Corrélations du modèle . . . . .   | 12        |
| 1.4 Estimation des paramètres du MGMRF . . . . .                                   | 14        |
| 1.4.1 Maximum de vraisemblance . . . . .   | 14        |
| 1.4.2 Maximum de pseudo-vraisemblance . . . . .                                    | 16        |
| 1.4.3 Estimateur MMSE . . . . .  | 17        |
| 1.5 Simulations . . . . .  | 18        |
| <b>2 Classification de texture</b>   | <b>22</b> |
| 2.1 Pré-traitement . . . . .   | 22        |
| 2.1.1 Extraction des images de texture . . . . .                                   | 24        |
| 2.1.2 Réduction d'espace . . . . .   | 24        |
| 2.2 Classification . . . . .   | 25        |
| <b>3 Extraction de paramètres de texture</b>                                       | <b>26</b> |
| 3.1 Paramètres envisageables . . . . .   | 26        |
| 3.2 Calcul des paramètres . . . . .  | 26        |
| 3.3 Retour sur le calcul de l'indice de projection pour la poursuite de projection | 30        |
| 3.3.1 Approximation de la loi de probabilité du paramètre de texture . . .         | 31        |
| 3.3.2 Distance . . . . .   | 35        |
| 3.3.3 Poursuite de projection sur les paramètres de texture . . . . .              | 36        |
| <b>4 Résultats</b>   | <b>37</b> |
| 4.1 Algorithme de classification . . . . .   | 37        |
| 4.2 Résultats sur des images de synthèse . . . . .                                 | 39        |
| 4.3 Résultats sur des images AVIRIS . . . . .                                      | 41        |

|   |           |
|---|-----------|
| <b>Conclusion</b>   | <b>45</b> |
| <b>A Annexe : Propriétés de la matrice <math>A</math></b> | <b>48</b> |
| A.1 Valeurs propres et déterminant . . . . .              | 48        |
| A.2 Dérivée . . . . .                                     | 49        |
| <b>Références</b>   | <b>51</b> |

## Introduction

Une image de télédétection est une image échantillonnée spatialement, prise par 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. Les images hyperspectrales, par opposition aux images monospectrales, sont des images de télédétection ayant la particularité de fournir à la fois un échantillonnage spatial et un échantillonnage spectral. Pour chaque échantillon spatial, elles ne fournissent plus 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. sa longueur d'onde correspondante).

Ces images ont un nombre de canaux en général supérieur à 10, et chacun de ces canaux est associé à une longueur d'onde du spectre allant du visible à l'infrarouge. Par rapport aux images multispectrales (dont le nombre de canaux est supérieur à 1 et inférieur à 10), ce découpage spectral plus fin autorise des mesures radiométriques plus précises, et permet donc une caractérisation plus exacte du matériau présent sur le terrain. Ceci autorise en particulier des applications liées à la reconnaissance du sol en géologie, à l'aide à l'agriculture de précision, etc. . .

En contrepartie, ces mesures très précises spectralement sont moins 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, alors que la nouvelle génération des satellites monospectraux atteint des résolutions métriques.

Dans ce travail, on s'intéresse à l'analyse des textures présentes dans des images hyperspectrales tirant parti à la fois des relations spatiales et spectrales de celles-ci, visant ainsi à ajouter l'information spectrale aux méthodes classiques d'analyse de texture prenant en compte les variations spatiales des niveaux de gris.

L'application dans laquelle s'inscrit ce travail est la classification des zones urbaines. En effet, ces zones sont constituées de nombreux matériaux, ce qui est la source d'une grande variabilité de réflectance, qui se traduit à la fois entre différents pixels et à l'intérieur d'un même pixel dont la réflectance peut alors être due à un mélange de plusieurs composantes "pures". De plus, une partie de ces matériaux sont aussi caractéristiques de certains paysages naturels, donnant à certains pixels urbains les mêmes signatures spectrales que ces paysages naturels. De ce fait, l'étude des réponses radiométriques de ces zones est insuffisante pour



aboutir à un résultat satisfaisant. Il est donc nécessaire d'étudier les zones urbaines à travers leur texture.

Les méthodes d'analyse de texture couramment utilisées en traitement d'image, comme les matrices de coocurrence [1] ou les filtres de Gabor [2], ne peuvent être adaptées simplement au cas des images multicomposantes qu'au prix d'une complexité algorithmique qui peut s'avérer rédhibitoire si l'on veut tirer parti des relations spectrales. Dès lors, la seule adaptation envisageable consiste à faire une analyse scalaire bande par bande, les unes indépendamment des autres, suivie par une étape nécessaire de fusion de données pour la classification. Pour mettre en œuvre l'analyse de texture telle qu'on la désire, une méthode vectorielle semble mieux adaptée, car les méthodes de ce type prennent en compte les relations interbandes, qui, on le verra plus tard, sont porteuses d'information. De telles approches ont déjà été considérées, en détection d'anomalies [3] par exemple.

Dans la première partie de ce rapport, on propose une modélisation des images hyperspectrales prenant en compte la texture, fondée sur les champs de Markov, ainsi que les méthodes d'estimation de paramètres associées à ce modèle. Le dernier paragraphe de cette partie est consacré à des simulations de texture par ce modèle de façon à en montrer l'adéquation avec le problème posé.

Les parties suivantes sont consacrées à l'application de cette modélisation à l'analyse et la classification de textures d'images hyperspectrales. Dans la deuxième partie, on explique l'algorithme de classification que l'on va mettre en œuvre. Il est constitué de deux étapes : une étape de pré-traitement durant laquelle une réduction de dimension est effectuée et des paramètres de textures sont extraits sur la représentation résultant d'une autre réduction de l'espace de l'image, indépendante de la première. La seconde étape est l'étape de classification proprement dite. Dans la troisième partie, l'étape d'extraction de paramètres est expliquée plus en détails, car elle découle de la modélisation proposée au premier chapitre. On opère une réduction de dimension à deux reprises. En effet, lorsque l'on considère l'analyse de données de grande dimension par des méthodes statistiques, on est confronté aux problèmes liés au phénomène de Hughes [4], se caractérisant par la difficulté de réaliser de bonnes estimations dans des espaces qui se trouvent être pratiquement "vides". La méthode de réduction de dimension est expliquée dans le chapitre 2, et on y apporte un complément à la fin du chapitre 3.

Les résultats de la méthode appliquée à des images du capteur AVIRIS sont exposés, commentés et comparés avec d'autres méthodes plus classiques, dans la dernière partie de ce rapport.

## 1 Modèle markovien gaussien de texture

Les approches en matière d'analyse de texture sont nombreuses et il serait difficile d'en faire une revue exhaustive. Parmi les plus populaires, et sans doute les plus efficaces en analyse de texture monodimensionnelle, on trouve les approches utilisant les matrices de cooccurrence pour extraire des paramètres caractéristiques de la texture observée [1]. Ces approches donnent de bons résultats pour les images monodimensionnelles, mais leur adaptation aux images multidimensionnelles paraît irréalisable. En effet, une matrice de cooccurrence est une matrice de dimension  $N \times N$ , où  $N$  est le nombre de niveaux de gris quantifiés de l'image. 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, et  $N_b$  le nombre de bande. Or les images AVIRIS ont 224 bandes et sont codées sur 16 bits!

L'approche par filtrage de Gabor [2] est plus rapide, mais l'extension aux données multivariées sans perte d'information n'est pas évidente, pour les mêmes raisons que précédemment concernant la prise en compte de l'information inter-bande.

La modélisation par champs de Markov, que l'on présente dans les paragraphes suivants, 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 sur l'ensemble des bandes conjointement, permet de prendre en compte les relations inter-bandes.

### 1.1 Modèles markoviens

Les champs de Markov sont des outils très utilisés 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, le pixel).  $X^s$  désigne le champ  $X$  privé de  $X_s$ , et  $V_s$  désigne le voisinage de  $s$ . 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 :

$$P(X_s = x_s / X^s = x^s) = P(X_s = x_s / \{x_t\}, t \in V_s) \quad (1)$$

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 modé-

lisation permet d'introduire explicitement des interactions à courte portée, et induit implicitement des interactions à plus longue portée par "transitivité".

C'est ainsi que les champs de Markov sont efficacement utilisés en segmentation/classification 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 [5, 6]. Dans ce cadre, il existe de nombreux modèles markoviens de texture, comme par exemple le modèle auto-binomial [7], le modèle de Derin-Elliott [8] ou le modèle markovien gaussien aussi appelé auto-normal, que l'on détaille par la suite. Bien qu'il existe des modèles markoviens de texture plus sophistiqués [9], le modèle markovien gaussien est particulièrement bien adapté au problème d'analyse de textures urbaines.

## 1.2 Modèle markovien gaussien

Le modèle markovien gaussien a beaucoup été utilisé en imagerie monodimensionnelle pour l'analyse de texture, notamment urbaine [10, 11]. Mais il est aussi possible d'étendre cette modélisation dans le cadre de l'imagerie multispectrale ; deux approches peuvent être distinguées dans ce cas. Dans la première approche [3, 12], 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. Le problème de cette approche est que l'on suppose, en plus de la stationnarité spatiale, une stationnarité spectrale qui n'est en général pas vérifiée. La deuxième approche, notamment celle qu'a utilisée Hazel dans [13], 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 deuxième approche, les interactions entre les valeurs à différentes longueurs d'onde ne sont pas modélisées explicitement, mais elles existent tout de même, et découlent des paramètres qui sont 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 ce rapport sont reliées à une modélisation de ce dernier type, que l'on appelle MGMRF (en anglais *Multivariate Gaussian Markov Random Field*).

## 1.3 MGMRF

### 1.3.1 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 précédent, 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, qui est supposée être de la forme suivante :

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

la notation  $\|a\|_{\Sigma}^2$  signifiant  $a^t \Sigma^{-1} a$ , les  $\theta_i$  étant des matrices de transformation, ou paramètres d'interaction et  $\Sigma$  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 valable 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 :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_{t \in V_s} \theta_{t-s} x_t - (I_{N_b} - \sum_{t \in V_s} \theta_{t-s}) \mu\|_{\Sigma}^2 \right\} \quad (3)$$

où  $I_{N_b}$  désigne la matrice identité, et  $\mu$  la moyenne du champ.

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

La formulation (2) permet d'écrire :

$$x_s = \sum_{t \in V_s} \theta_{t-s} x_t + u_s \quad (4)$$

avec  $u_s$  bruit blanc gaussien, centré, de matrice de covariance  $\Sigma$ .

De plus, pour des contraintes de cohérence et de stationnarité du champ, on doit avoir  $\theta_i = \theta_{-i}$ , ce qui conduit à :

$$x_s = \sum_{k=0}^K \theta'_k m_s^k + u_s \quad (5)$$

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  $t$  d'un pixel  $s$  ayant la même

contribution dans le modèle, ce qui se traduit par le même paramètre  $\theta_{s-t}$ . A 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 voisinage en 8-connexité anisotrope, on peut considérer quatre types de voisins : les voisins horizontaux, verticaux, et les deux types de voisins correspondant aux deux diagonales.

### 1.3.2 Loi globale

Pour obtenir la formule de la loi globale, on revient sur la formule (4) :

$$u_s = x_s - \sum_{t \in V_s} \theta_{t-s} x_t \quad (6)$$

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  $u = (u_{(0,0)}^t \dots u_{(N_v, N_h)}^t)^t$ . Il vient alors :

$$u = Ax \quad (7)$$

où la structure de  $A$  est déterminée par les paramètres  $\theta$  et le système de voisinage considéré.

Le vecteur  $u$  suit une loi normale, donc le vecteur  $x$  également. On peut écrire :

$$P(x) = \frac{1}{Z(S)} \exp\left(-\frac{1}{2} x^t S^{-1} x\right) \quad (8)$$

avec  $S = E\{xx^t\} = A^{-1}E\{ux^t\}$  et  $Z(S)$  la constante de normalisation.

On peut prouver, d'après le principe d'orthogonalité, que cette matrice s'écrit comme suit [13] :

$$E\{ux^t\} = I_{N_h} \otimes I_{N_v} \otimes \Sigma \quad (9)$$

où  $\otimes$  désigne le produit de Kronecker<sup>1</sup> [14], et  $I_n$  la matrice identité de dimension  $n$ .

Donc l'équation (8) peut être réécrite:

$$P(x) = \frac{1}{Z(S)} \exp\left(-\frac{1}{2} x^t (I_{N_h} \otimes I_{N_v} \otimes \Sigma^{-1}) Ax\right) \quad (10)$$

---

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  $AB = C = \{c_{ij}\}$  de dimension  $(mn,mn)$  telle que  $c_{im+k, jm+l} = a_{i,j} b_{k,l}$

### 1.3.3 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 la classification des images AVIRIS à 224 bandes spectrales.

Le processus de classification tel qu'on l'envisage est constitué dans un premier temps d'une phase d'apprentissage durant laquelle les classes à distinguer sont définies par un opérateur, puis caractérisées par l'estimation de leurs paramètres.

Le nombre de paramètres du modèle markovien gaussien est variable, selon l'ordre du voisinage, son anisotropie, ou les contraintes appliquées à la matrice  $\theta$ . Dans le cas d'un modèle anisotrope à 4 voisins par exemple, il est nécessaire de calculer les deux matrices  $\theta_v$  et  $\theta_h$  correspondant aux voisinages vertical et horizontal, ainsi que la matrice de covariance conditionnelle  $\Sigma$  (soit, si le vecteur d'observation est de dimension  $N$ ,  $3N^2$  paramètres). Dans le cas de données AVIRIS à 224 bandes, l'estimation d'un nombre aussi élevé de paramètres est très difficile, surtout étant donné le nombre d'échantillons disponibles pour effectuer le calcul, qui ne peut excéder quelques milliers. Or plus le nombre de paramètres est élevé, plus le phénomène de Hughes [4] 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, 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, la simplification consistera à donner une certaine forme *a priori* aux matrices de transformation  $\theta_i$  :

$$\theta_i = \text{diag}(a_i) = a_i \text{ Id} \quad (11)$$

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 dans ce rapport, cela se traduit par une même vitesse de décroissance des corrélations dans les différentes bandes.

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

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_{t \in V_s} a_{t-s} x_t\|_{\Sigma}^2 \right\} \quad (12)$$

On peut alors écrire :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z'} \exp \left\{ - \sum_{t \in V_s} \alpha_{t-s} \|x_s - x_t\|_{\Sigma}^2 - \beta \|x_s\|_{\Sigma}^2 \right\} \quad (13)$$

### 1.3.4 Domaine de validité des paramètres

Pour que le champ  $X$  de loi (8) existe, il faut que la matrice  $S$  soit définie positive, par conséquent il faut que  $A$  soit définie positive si l'on considère que  $\Sigma$  l'est. Cette dernière supposition est vraisemblable, si l'on considère un nombre d'échantillons suffisamment élevé.

Par la suite, on prendra la formulation de la probabilité conditionnelle suivante :

$$P(x_s | \{m_k\}) = \frac{1}{Z} \exp \left\{ - \frac{1}{2} \|x_s - \sum_k a'_k m_s^k\|_{\Sigma}^2 \right\} \quad (14)$$

où les  $a'_k$  sont obtenus à partir des  $a_k$  comme les  $\theta'$  à partir des  $\theta$  (cf. équation (5)).

Par un calcul similaire à celui mené dans [13], on montre qu'une condition nécessaire pour l'existence du champ markovien gaussien de paramètres  $\{a'_k\}$  est que  $\sum_{k=0}^{K-1} a'_k < 1$ , où  $K$  désigne le nombre de voisins.

### 1.3.5 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 1.4.3).

La loi de probabilité globale du MGMRF décrite en (10) vient de la formule (9), que l'on peut récrire comme suit :

$$E\{xx^t\} = A^{-1} (I_{N_h} \otimes I_{N_v} \otimes \Sigma) \quad (15)$$

On en déduit une première information sur les matrices de corrélation entre «hyper-pixels»  $x_s$  :

$$C_d = A_{(i,i+d)}^{-1} \Sigma \quad (16)$$

où  $C_d = E\{x_s x_{s+d}\}$ .

Celles-ci sont donc proportionnelles à la matrice de covariance conditionnelle, avec un coefficient de proportionnalité dépendant du déplacement  $d$ . Ce coefficient peut être calculé

| Cas<br>$n^{\circ}$ | Texture  | Nb<br>Bandes | Moyenne<br>$\bar{Q}_{01}$ | Ecart type<br>$\sigma_{Q_{01}} (\% \bar{Q}_{01})$ | Moyenne<br>$\bar{Q}_{10}$ | Ecart type<br>$\sigma_{Q_{10}} (\% \bar{Q}_{10})$ |
|--------------------|----------|--------------|---------------------------|---|---------------------------|---|
| 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 %  |

FIG. 1 – Moyennes 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

en inversant la matrice  $A$ , par exemple en passant par la transformation de Fourier (cf. Annexe).

On peut, grâce à cette propriété, vérifier l'adéquation 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 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. Le tableau donne 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 différentes zones urbaines, provenant de l'image de Moffet 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. Par conséquent, dans le cas de textures urbaines à faible résolution, l'hypothèse simplificatrice sur les paramètres est réaliste. De même pour les cas 3 et 4, on garde pour les mêmes textures 25 bandes provenant cette fois-ci de l'infrarouge. On constate dans ces deux cas, que l'on a également une très faible variabilité des quotients  $Q$ . L'hypothèse est donc toujours réaliste. Enfin, les cas 5 et 6 montrent les résultats pour les mêmes textures, mais pour lesquelles on a gardé 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, l'on considèrera que l'on est dans ce cas. Les images hyperspectrales couvrant en général diverses parties du spectre, il faudra découper le spectre en zones aux propriétés spectrales similaires, et les traiter séparément.

## 1.4 Estimation des paramètres du MGMRF

Considérons une image  $x$  de taille  $N_v \times N_h$ , de moyenne nulle, supposée avoir été générée par le modèle décrit par l'équation (12), dont les paramètres sont inconnus. On désire estimer ces paramètres d'une façon rapide et précise.

### 1.4.1 Maximum de vraisemblance

Un critère très utilisé pour l'estimation de paramètres est le critère du Maximum de Vraisemblance (MV). Il consiste à retenir le vecteur de paramètres  $\hat{\phi}$  (dans le cas présent  $\phi = (\{a_k\}, \Sigma)$ ), tel que :

$$\hat{\phi} = \arg \max_{\phi} (P(x/\phi)) \quad (17)$$

En revenant à l'équation (8) et en explicitant la constante de normalisation, il vient :

$$P(x/\phi) = \frac{(2\pi)^{-\frac{N_v N_h N_b}{2}}}{|S|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}x^t S^{-1}x\right) \quad (18)$$

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

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

$$LV(\phi) = C - \frac{1}{2} \ln |S| - \frac{1}{2}x^t S^{-1}x \quad (19)$$

où  $C$  est une constante. Par conséquent :

$$\max_{\phi} (LV(\phi)) = \min_{\phi} (\ln |S| + x^t S^{-1}x) \quad (20)$$

D'où :

$$\hat{\phi} = \arg \min_{\phi} (\ln |S| + x^t S^{-1}x) \quad (21)$$

L'optimisation de cette expression est complexe, du fait du calcul du déterminant de  $S$  (même si cela reste possible efficacement pour certaines variantes du modèle [3, 15],

moyennant quelques approximations). Dans notre cas, cette fonction de log-vraisemblance est calculable. Nous avons (cf. équation 10) :

$$S^{-1} = (I_{N_h} \otimes I_{N_v} \otimes \Sigma^{-1}) A \quad (22)$$

Donc :

$$|S|^{-1} = |\Sigma|^{-N_h N_v} |A| \quad (23)$$

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

$$|A| = \prod_{x=0, y=0}^{N_h, N_v} \sum_{m=0, n=0}^{N_h, N_v} a_{m,n} \exp \left( 2\pi i \left( \frac{xm}{N_h} + \frac{yn}{N_v} \right) \right) \quad (24)$$

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

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  $\Sigma$  connaissant les  $\{a_k\}$ .

$$\hat{\Sigma}(\{a_k\}) = \frac{1}{N} \sum_i x_i u_i^t \quad (25)$$

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

$$\max_{\phi} LV(\phi) = \max_{\{a_k\}} LV(\{a_k, \hat{\Sigma}(\{a_k\})\}) \quad (26)$$

L'espace de recherche du maximum est réduit de façon considérable, et la recherche est plus efficace.

**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 (26) et (23), on obtient l'équation suivante :

$$\hat{a} = \arg \max_a (LV(a)) = \arg \min_a \left( \sum_s x_s^t \hat{\Sigma}^{-1} u_s + N_h N_v \ln |\hat{\Sigma}| - \ln |A| \right) \quad (27)$$

où  $a$  est le vecteur des paramètres  $a_k$ , et  $\hat{\Sigma} = \hat{\Sigma}(a)$ .

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

$$\frac{\partial}{\partial a_k} LV = \frac{\partial}{\partial a_k} \ln |A| - N_h N_v \frac{\partial}{\partial a_k} \ln |\hat{\Sigma}| \quad (28)$$

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 :

$$\frac{\partial |M|}{\partial x} = |M| \text{trace} \left( M^{-1} \frac{\partial M}{\partial x} \right) \quad (29)$$

les dérivées partielles du deuxième terme s'écrivent :

$$\frac{\partial}{\partial a_k} \ln |\hat{\Sigma}| = \text{trace}(\hat{\Sigma}^{-1} C_k) \quad (30)$$

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

Donc, quand la dérivée de  $LV$  est nulle, on est au maximum de vraisemblance, et l'on a :

$$A_{m,n}^{-1} = \text{trace}(\hat{\Sigma}^{-1} C_{m,n}), \forall (m,n) \in \mathcal{N}_0 \quad (31)$$

L'estimateur du maximum de vraisemblance fait donc coïncider la corrélation empirique et la corrélation théorique d'un pixel avec son voisinage.

#### 1.4.2 Maximum de pseudo-vraisemblance

Un critère permettant une estimation plus rapide est celui du maximum de pseudo-vraisemblance [7]. Ce critère est fondé sur une hypothèse d'indépendance des probabilités conditionnelles et mène à l'approximation suivante concernant la probabilité du champ  $x$  :

$$P(x) \approx \prod_{s \in \Omega} P(x_s | \{x_t\}, t \in V_s) \quad (32)$$

L'estimateur de maximum de pseudo-vraisemblance est :

$$\phi = \arg \max_{\phi} \left( \prod_{s \in \Omega} P(x_s | \{x_t\}, \phi) \right) \quad (33)$$

$$\phi = \arg \min_{\phi} \left( \sum_s \|x_s - \sum_k a'_k m_s^k\|_{\Sigma}^2 + N_v N_h \ln |\Sigma| \right) \quad (34)$$

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

$$\begin{pmatrix} \langle (m_s^0)^t \Sigma^{-1} m_s^0 \rangle & \dots & \langle (m_s^0)^t \Sigma^{-1} m_s^K \rangle \\ \vdots & & \vdots \\ \langle (m_s^K)^t \Sigma^{-1} m_s^0 \rangle & \dots & \langle (m_s^K)^t \Sigma^{-1} m_s^K \rangle \end{pmatrix} \begin{pmatrix} a'_0 \\ \vdots \\ a'_K \end{pmatrix} = \begin{pmatrix} \langle x_s^t \Sigma^{-1} m_s^0 \rangle \\ \vdots \\ \langle x_s^t \Sigma^{-1} m_s^K \rangle \end{pmatrix} \quad (35)$$

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  $\Sigma$ .

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

$$\Sigma = \langle u_s u_s^t \rangle = \left\langle \left( x_s - \sum_k a'_k m_s^k \right) \left( x_s - \sum_k a'_k m_s^k \right)^t \right\rangle \quad (36)$$

### 1.4.3 Estimateur MMSE

On peut également dériver un estimateur à partir d'un autre critère : l'erreur quadratique moyenne. Le 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 :

$$E\{x_r^t u_s\} = 0, \quad \forall s \neq r \quad (37)$$

En explicitant  $u_s$  et en développant, l'équation (37) devient :

$$E\{x_r^t x_s\} - \sum_{t \in V_s} a_{t-s} E\{x_r^t x_t\} = 0, \quad \forall s \neq r \quad (38)$$

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

$$c_d - \sum_{t \in V_0} a_t c_{d+t} = 0, \quad \forall d \neq 0 \quad (39)$$

Il faut autant d'équations que d'inconnues pour résoudre le système, un choix naturel est donc de prendre le système formé des équations (39) 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 (35).

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 :

$$a' = 4a = \frac{16c_{0,1}}{4c_{0,0} + 8c_{1,1} + 4c_{0,2}} \quad (40)$$

## 1.5 Simulations

Les simulations vont nous permettre de vérifier deux points. D'une part, nous allons pouvoir valider la qualité des estimateurs grâce à des tests d'estimation sur des images simulées dont on connaît les paramètres, 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 que nous adoptons est la suivante :

- Dans un premier temps, on réduit la dimension de la texture proposée par ACP, 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 extrayant  $\hat{a}$  et  $\hat{\Sigma}$  correspondant à un modèle isotrope 4-connex.
- Dans un troisième temps, on échantillonne une texture ayant ces paramètres par l'algorithme de Metropolis.
- Enfin, on replonge la texture dans l'espace initial (en effectuant la rotation inverse), de façon à avoir une idée visuelle du résultat.

La figure 2 montre les résultats obtenus pour des images de synthèse (dont on connaît les paramètres). Les paramètres choisis pour les simulations ne correspondent pas à des situations réelles, mais ils permettent de tester l'efficacité de l'estimation. Ici, on ne passe pas par l'ACP, car les données image ont été simulées avec 3 bandes spectrales, afin de faciliter la simulation.

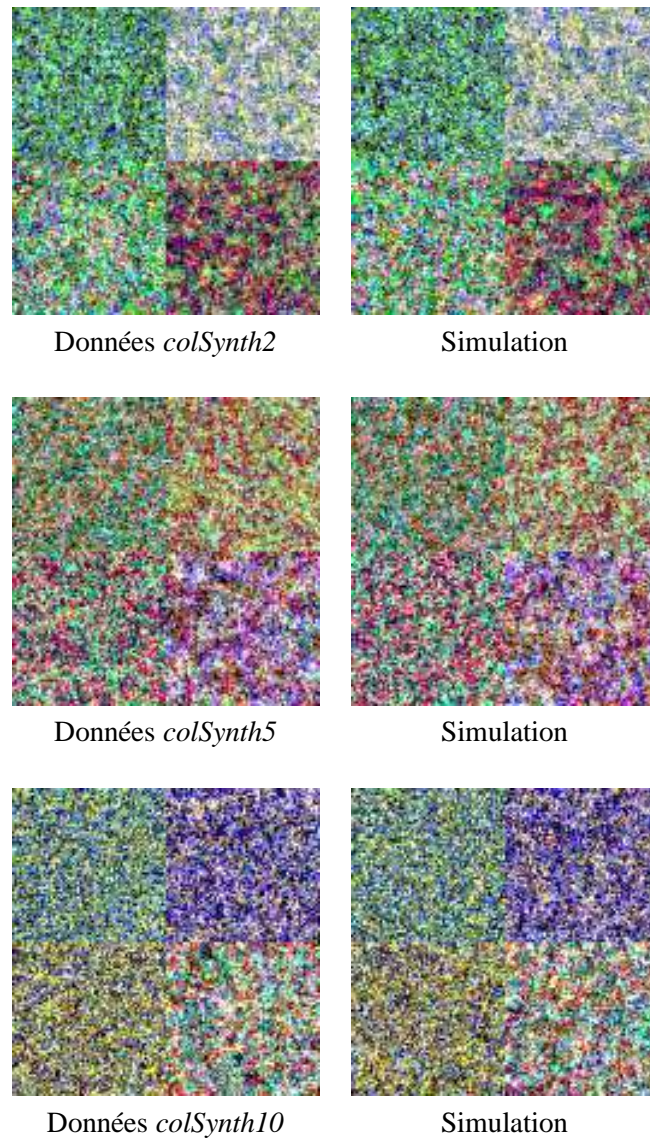


FIG. 2 – Données simulées à différents rapports Signal sur Bruit ( $S/B$ ) (colonne de gauche), simulation à partir des paramètres estimés pour chaque quart de l'image (colonne de droite).

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 sont données dans le tableau 3. Les résultats numériques sont très satisfaisants, puisqu'ils montrent que pour ces images "idéales" (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  $\Sigma$  (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 grossière dans l'estimation du paramètre d'interaction pour la zone en haut à droite (0.843 au lieu de 0.9). 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. Ceci a pour conséquence que 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.

Pour ce qui est de la deuxième partie des simulations, on montre des résultats d'analyse/synthèse sur des images réelles dans la figure 4. 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 4, 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. [9] 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 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.

|   | <i>colSynth2</i> N-O |           |            |
|---|----------------------|-----------|------------|
|   | Réel                 | Estimé MV | Estimé MPV |
| Paramètre $a$                                       | 0.8                  | 0.805     | 0.770      |
| Matrice $\Sigma$ : écart moyen des coefficients (%) | -                    | 2.5       | 3.8        |
| Matrice $\Sigma$ : 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 $\Sigma$ : écart moyen en %   | -                    | 3         | 3          |
| Matrice $\Sigma$ : 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 $\Sigma$ : écart moyen en %   | -                    | 1         | 2.7        |
| Matrice $\Sigma$ : 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 $\Sigma$ : écart moyen en %   | -                    | 1.3       | 1.3        |
| Matrice $\Sigma$ : écart maximal en % | -                    | 3.8       | 3.8        |

FIG. 3 – Paramètres réels et estimés des 4 parties (NO, SO, NE, SE) de l'image *colSynth2*.



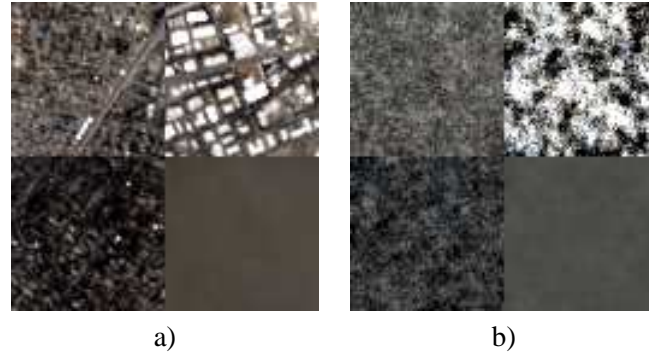


FIG. 4 – a) Image initiale, b) Image simulée avec les paramètres extraits de l'image a).

## 2 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).

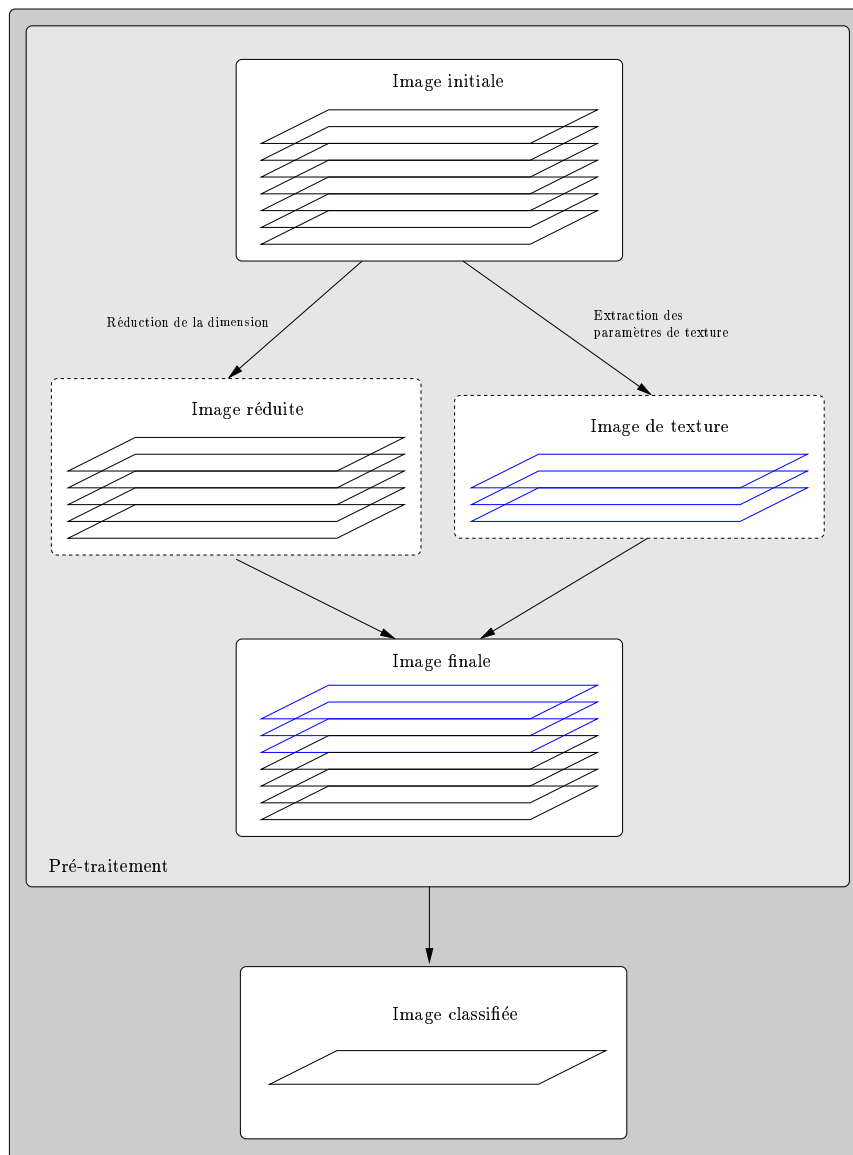
En matière de classification, une méthode envisageable d'analyse de texture consiste à faire l'hypothèse d'un modèle probabiliste, et de tester la vraisemblance de ce modèle sur les données. C'est la méthode que l'on utilise dans [16]. Cette méthode s'avère efficace dans le cas où les textures sont irrégulières, et où la moyenne de la texture est peu significative (peu discriminante). En effet, celle-ci rentre pour une très faible part dans le calcul de la vraisemblance du modèle.

Dans un cas plus général, on souhaite avoir une méthode où l'influence de la radiométrie et de la texture soient 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).

L'algorithme de classification proposé ici 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.

### 2.1 Pré-traitement

Comme l'on vient de l'expliquer, le pré-traitement est l'étape d'extraction de caractéristiques. Cette étape est constituée de deux sous-étapes bien distinctes, qui ont chacune un but

FIG. 5 – *Algorithme de classification.*

précis. La première sous-étape consiste à chercher un sous-espace de l'espace initial dans lequel projeter les données. Son but est de limiter les problèmes liés à l'estimation statistique dans des espaces de grande dimension (cf. introduction). L'autre sous-étape est l'extraction des paramètres de texture nécessaires à l'analyse, accompagnée également d'une réduction de dimension.

### 2.1.1 Extraction des images de texture

Cette sous-étape est bien entendu la phase clé de notre analyse. Le chapitre suivant est consacré à cette phase.

### 2.1.2 Réduction d'espace

De nombreuses publications ont eu pour objet d'expliquer le besoin de la réduction de dimension dans l'analyse de données de grande dimension [4, 17]. Beaucoup de méthodes ont été développées dans ce cadre ou adaptées dans ce but, la plus connue étant l'analyse en composantes principales (ACP [18]). Cette méthode vise à déterminer le sous-espace de dimension donnée qui minimise l'erreur quadratique totale. Le fait que cette méthode soit très utilisée alors qu'elle est très simple en montre bien l'intérêt et l'efficacité dans le cas général. Pourtant, il s'avère que dans le cadre de la classification, cette méthode est moins intéressante que certaines autres, ou que d'autres en combinaison avec elle. Parmi celles-ci, on retient les méthodes de poursuite de projection [19, 20].

La poursuite de projection 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 [19], 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 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 [20]. 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.

Nous avons proposé une variante de ce dernier algorithme [16], fondée sur la modélisation par MGMRF, utilisant une distance de Bhattacharyya liée au modèle. Néanmoins, dans cette application, on en restera à l'indice de projection de Jimenez et Landgrebe pour ce qui concerne la réduction de dimension des images de radiométrie. Quant à l'indice de projection optimisé pour les images de textures, on y reviendra dans le paragraphe suivant.

## 2.2 Classification

Le critère qui dirige la phase de classification est le maximum de vraisemblance. Soit  $X$  l'image pré-traitée, destinée à être classifiée. En chaque pixel  $s$ , la mesure vectorielle  $x^s$  est constituée de la mesure radiométrique  $x_r^s$ , et de la mesure de texture  $x_t^s$ .

considérons un problème à  $L$  classes  $C_l$ ,  $l = 0 \dots L - 1$ . La règle de décision du maximum de vraisemblance (MV) consiste à assigner le label  $\hat{\lambda}_s$  au pixel  $s$  de la façon suivante :

$$\hat{\lambda}_s = \arg \max_{l \in \{0, L-1\}} P_l(x^s) = \arg \max_{l \in \{0, L-1\}} P_l(x_r^s, x_t^s) \quad (41)$$

où  $P_l(x^s) = P(x^s / \lambda_s = l)$ .

Or, on peut écrire :  $P_l(x_r^s, x_t^s) = P_l(x_r^s / x_t^s) P_l(x_t^s)$ . A ce moment, on fait l'hypothèse que la mesure de radiométrie est indépendante de la texture, connaissant le label du pixel observé, c'est-à-dire :  $P_l(x_r^s / x_t^s) = P_l(x_r^s)$ . Par conséquent,  $P_l(x_r^s, x_t^s) = P_l(x_r^s) P_l(x_t^s)$ . Et, il vient :

$$\hat{\lambda}_s = \arg \max_{l \in \{0, L-1\}} P_l(x_r^s) P_l(x_t^s) \quad (42)$$

La partie radiométrie du vecteur  $X^s$  est supposée suivre une loi conditionnelle normale  $\mathcal{N}(\mu_l, \Gamma_l)$ . En ce qui concerne la partie texture de ce vecteur, on y reviendra après avoir décrit les paramètres choisis.

### 3 Extraction de paramètres de texture

Nous avons donc décrit un modèle d'image texturée multidimensionnelle adapté aux images hyperspectrales, ainsi que les méthodes d'estimation qui y sont liées. Ce modèle va servir à décrire les textures, et donc à les caractériser.

Dans ce chapitre, on s'attache à expliquer et justifier le choix des grandeurs qui serviront à identifier les textures.

#### 3.1 Paramètres envisageables

Les outils d'analyse permettant d'obtenir les images de texture peuvent être, par exemple, les ondelettes, les filtres de Gabor, 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 s'agit donc de paramètres de deux types : la matrice de covariance conditionnelle  $\Sigma$ , et des paramètres d'interaction qui, dans le cas particulier du MGMRF que l'on a décrit, sont réduits à un vecteur, le vecteur  $a$ .

#### 3.2 Calcul des paramètres

Dans un premier temps, il est nécessaire d'obtenir les images de textures :

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 1.4.

Dans les applications, nous utilisons l'estimateur MMSE, pour des raisons de rapidité.

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 fenêtres 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 des fenêtres, on augmente la possibilité d'y trouver plusieurs textures. Pour remédier à ce problème, on passe par une euristique. Prenons le cas d'une fenêtre dans laquelle on aurait deux classes représentées,  $C_0$  et  $C_1$ , de moyennes  $\mu_0$  et  $\mu_1$  et de matrices de covariance  $\Gamma_0$  et  $\Gamma_1$ , à raison de proportions  $n_0$  et  $n_1$ , tq.  $n_0 + n_1 = 1$ .

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

$$\hat{\Gamma} = \frac{1}{N-1} \sum_s (x_s - \hat{\mu})(x_s - \hat{\mu})^t \quad (43)$$

Soit  $\Omega_0$  (resp.  $\Omega_1$ ), la zone correspondant à la classe  $C_0$  (resp.  $C_1$ ).

$$\hat{\Gamma} = \frac{1}{N-1} \left( \sum_{s \in \Omega_0} (x_s - \hat{\mu})(x_s - \hat{\mu})^t + \sum_{s \in \Omega_1} (x_s - \hat{\mu})(x_s - \hat{\mu})^t \right) \quad (44)$$

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 :

$$\begin{aligned} (N-1)\hat{\Gamma} &= \left( \sum_{s \in \Omega_0} (x_s - \hat{\mu}_0 + n_1(\hat{\mu}_1 - \hat{\mu}_0))(x_s - \hat{\mu}_0 + n_1(\hat{\mu}_1 - \hat{\mu}_0))^t \right. \\ &\quad \left. + \sum_{s \in \Omega_1} (x_s - \hat{\mu}_1 + n_0(\hat{\mu}_0 - \hat{\mu}_1))(x_s - \hat{\mu}_1 + n_0(\hat{\mu}_0 - \hat{\mu}_1))^t \right) \end{aligned} \quad (45)$$

Soit :

$$\begin{aligned} (N-1)\hat{\Gamma} &= \sum_{s \in \Omega_0} (x_s - \hat{\mu}_0)(x_s - \hat{\mu}_0)^t \\ &\quad + \sum_{s \in \Omega_1} (x_s - \hat{\mu}_1)(x_s - \hat{\mu}_1)^t \\ &\quad + n_1 n_0 (n_1 - n_0) (\hat{\mu}_1 - \hat{\mu}_0) (\hat{\mu}_1 - \hat{\mu}_0)^t \end{aligned} \quad (46)$$

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

$$\hat{\Gamma} = n_0 \hat{\Gamma}_0 + n_1 \hat{\Gamma}_1 + n_1 n_0 (n_1 - n_0) M_{12} \quad (47)$$

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  $\mu_0$  et  $\mu_1$ . 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  $\Gamma_0$  et  $\Gamma_1$ .

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  $\Gamma$ , et l'autre sur  $a$ .

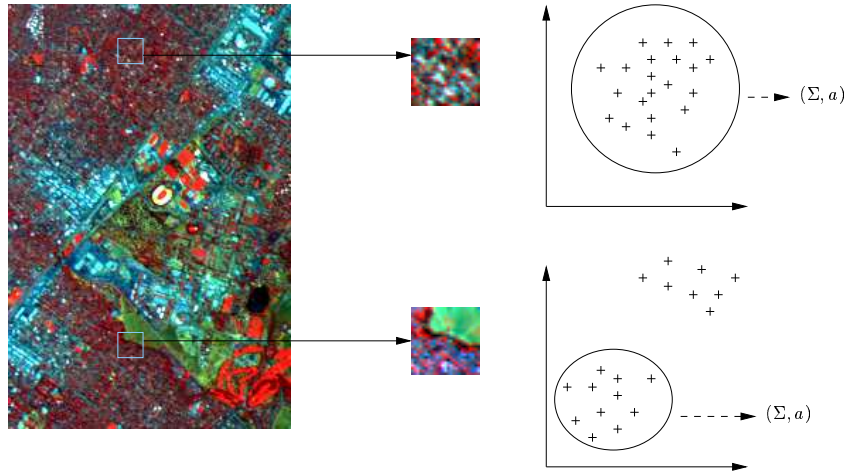


FIG. 6 – Illustration de l’heuristique : dans le cas du haut, une seule texture est présente, on calcule les paramètres sur tous les échantillons, dans le cas du bas, il y a deux classes, on calcule les paramètres sur la classe la plus présente.

L’heuristique (cf. fig 6) proposée ici est fondée sur le fait que le problème 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 supérieure 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 7 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 plus franches. En revanche, on n’élimine pas le problème 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  $\Sigma$  et on rejette le vecteur  $a$ . Pour justifier ce choix, la figure 8 montre le

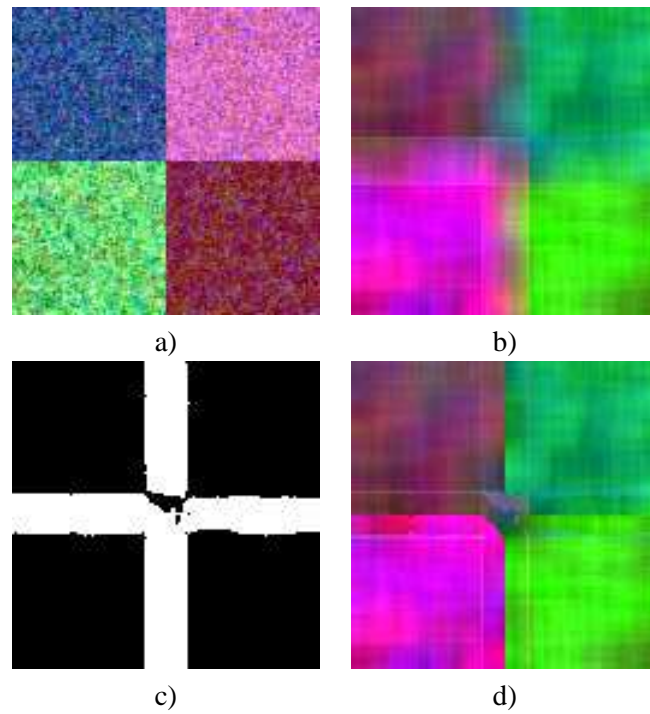


FIG. 7 – a) collage de textures à 3 bandes spectrales simulées par MGMRF, b) image en fausses couleurs des éléments diagonaux de  $\hat{\Sigma}$  sans euristique, c) transitions détectées par l'euristique, d) éléments diagonaux de  $\hat{\Sigma}$  avec l'euristique.



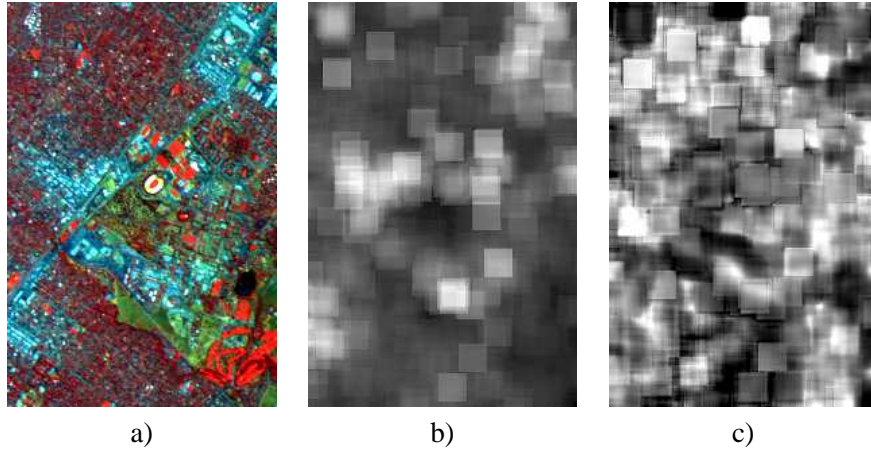


FIG. 8 – a) Extrait de 3 bandes visibles de l’image AVIRIS de Moffett Field, b) variance conditionnelle  $\sigma$  extraite d’une bande de cette image, c) paramètres d’interaction  $a$  extrait d’une bande de cette image.

résultat de l’extraction de  $\Sigma$  et  $a$  dans le cas d’une dimension finale égale à 1 concernant la texture et d’une modélisation isotrope. Dans ce cas,  $\Sigma$  est réduite à un scalaire, 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.

### 3.3 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  $\Sigma$  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 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 [20]. 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.

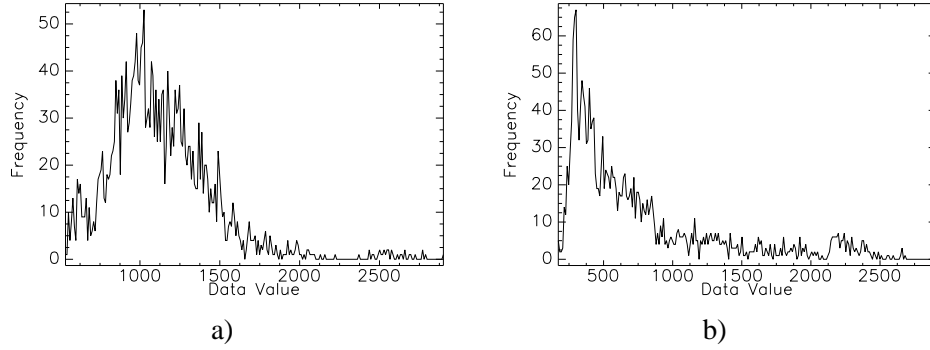


FIG. 9 – 2 histogrammes du paramètre  $\sigma$  calculé sur une bande du spectre visible de 2 zones de l'image AVIRIS de Moffett Field.

### 3.3.1 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 9 montre des exemples d'histogrammes du paramètre  $\sigma$  calculé sur une bande du spectre visible de 2 zones d'une image AVIRIS, ceux-ci n'ont pas la forme de gaussiennes.

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

$$\hat{\Sigma} = \frac{1}{N} \sum_s u_s u_s^t \quad (48)$$

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

Si les  $u_s$  avaient été indépendants, la loi de  $S_u$  aurait été une loi de Wishart à  $N$  degrés de liberté ( $N$  étant le nombre de pixels servant à l'estimation), de paramètre  $\Sigma$  (cf. [21]). Cette loi est le pendant multidimensionnel de la loi  $\chi^2$ , et sa densité s'écrit de la façon suivante :

$$p(S_u) = \frac{|S_u|^{\frac{1}{2}(N-p-1)} \exp\left(-\frac{1}{2}\text{tr}\Sigma^{-1}S_u\right)}{K |\Sigma|^{\frac{1}{2}N}} \quad (49)$$

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  $\Sigma$ .

$$E\{S_{ij}\} = N\Sigma_{ij} \quad (50)$$

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

Et en ce qui concerne les moments d'ordre deux :

$$E\{(S_{ij} - N\Sigma_{ij})(S_{kl} - N\Sigma_{kl})\} = N(\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \quad (51)$$

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

$$E\{(\hat{\Sigma}_{ij} - \Sigma_{ij})(\hat{\Sigma}_{kl} - \Sigma_{kl})\} = \frac{1}{N}(\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \quad (52)$$

La distribution de  $\hat{\Sigma}$  est de plus en plus "piquée" autour de  $\Sigma$  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  $a$ , qui est supposé connu. 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  $S_x = A S_u A^t$ . Et enfin, soit  $y$  la transformée de Fourier de  $x$  et  $S_y = \sum_s y_s y_s^t$ .  $S_y = S_x$ .

$$E\{(S_y)_{ij}\} = E\left\{\sum_s y_{s,i} y_{s,j}\right\} \quad (53)$$

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. Annexes). Donc :

$$E\{(S_y)_{ij}\} = \sum_s E\{y_{s,i} y_{s,j}\} \quad (54)$$

$$E\{(S_y)_{ij}\} = \sum_s E\{\lambda_s z_{s,i} z_{s,j}\} \quad (55)$$

où les  $z_s$  sont des variables décorréelées de matrice de covariance  $\Sigma$ , et les  $\lambda_s$  sont les valeurs propres de la matrice  $A$ .

$$E\{(S_y)_{ij}\} = \sum_s \lambda_s \Sigma_{ij} = \Sigma_{ij} \sum_s \lambda_s \quad (56)$$

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

Donc :

$$E\{(S_y)\} = N\Sigma \quad (57)$$

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

Calculons maintenant les moments d'ordre deux :

$$E\{(S_y)_{ij}(S_y)_{kl}\} = E\left\{\sum_{s,t} y_{s,i} y_{s,j} y_{t,k} y_{t,l}\right\} \quad (58)$$

$$E\{(S_y)_{ij}(S_y)_{kl}\} = E\left\{\sum_s y_{s,i} y_{s,j} y_{s,k} y_{s,l}\right\} + E\left\{\sum_{s \neq t} y_{s,i} y_{s,j} y_{t,k} y_{t,l}\right\} \quad (59)$$

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

$$E\{(S_y)_{ij}(S_y)_{kl}\} = \sum_s E\{y_{s,i} y_{s,j} y_{s,k} y_{s,l}\} + \sum_{s \neq t} E\{y_{s,i} y_{s,j} y_{t,k} y_{t,l}\} \quad (60)$$

Pour le premier terme, on se réfère à la formule des moments d'ordre quatre d'une variable gaussienne [21] :

$$E\{z_{s,i} z_{s,j} z_{s,k} z_{s,l}\} = \Sigma_{ij} \Sigma_{kl} + \Sigma_{ik} \Sigma_{jl} + \Sigma_{il} \Sigma_{jk} \quad (61)$$

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

$$\sum_s E\{y_{s,i} y_{s,j} y_{s,k} y_{s,l}\} = \sum_s \lambda_s^2 (\Sigma_{ij} \Sigma_{kl} + \Sigma_{ik} \Sigma_{jl} + \Sigma_{il} \Sigma_{jk}) \quad (62)$$

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

$$E\{y_{s,i} y_{s,j} y_{t,k} y_{t,l}\} = E\{y_{s,i} y_{s,j}\} E\{y_{t,k} y_{t,l}\} \quad (63)$$

$$\sum_{s \neq t} E \{ y_{s,i} y_{s,j} y_{t,k} y_{t,l} \} = \sum_{s \neq t} \lambda_s \lambda_t \Sigma_{ij} \Sigma_{kl} = \Sigma_{ij} \Sigma_{kl} \sum_{s \neq t} \lambda_s \lambda_t \quad (64)$$

Par conséquent :

$$E\{(S_y)_{ij}(S_y)_{kl}\} = (\Sigma_{ij}\Sigma_{kl} + \Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \sum_s \lambda_s^2 + \Sigma_{ij}\Sigma_{kl} \sum_{s \neq t} \lambda_s \lambda_t \quad (65)$$

$$E\{((S_y)_{ij} - N\Sigma_{ij})((S_y)_{kl} - N\Sigma_{kl})\} = (\Sigma_{ij}\Sigma_{kl} + \Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \sum_s \lambda_s^2 + \Sigma_{ij}\Sigma_{kl} \sum_{s \neq t} \lambda_s \lambda_t - N^2 \Sigma_{ij}\Sigma_{kl} \quad (66)$$

$$E\{((S_y)_{ij} - N\Sigma_{ij})((S_y)_{kl} - N\Sigma_{kl})\} = (\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \sum_s \lambda_s^2 \quad (67)$$

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

$$E\{(\hat{\Sigma}_{ij} - \Sigma_{ij})(\hat{\Sigma}_{kl} - \Sigma_{kl})\} = (\Sigma_{ik}\Sigma_{jl} + \Sigma_{il}\Sigma_{jk}) \frac{1}{N^2} \sum_s \lambda_s^2 \quad (68)$$

On approxime donc  $\hat{\Sigma}$  par une variable  $\Sigma_w$  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 :

$$\lambda_{ij} = 1 - 2a \cos\left(\frac{2\pi i}{N_r}\right) - 2a \cos\left(\frac{2\pi j}{N_c}\right) \quad (69)$$

Donc :

$$\sum_{i,j} \lambda_{ij}^2 = \sum_{i,j} \left(1 - 2a \cos\left(\frac{2\pi i}{N_r}\right) - 2a \cos\left(\frac{2\pi j}{N_c}\right)\right)^2 \quad (70)$$

$$\sum_{i,j} \lambda_{ij}^2 = N^2 + 4a^2 \sum_{i,j} \cos^2\left(\frac{2\pi i}{N_r}\right) + 4a^2 \sum_{i,j} \cos^2\left(\frac{j}{N_c}\right) \quad (71)$$

Or :

$$\sum_{i=0}^{N_r-1} \cos^2\left(\frac{2\pi i}{N_r}\right) = \frac{N_r}{2} \quad (72)$$

Donc :

$$\sum_{i,j} \lambda_{ij}^2 = N + 4a^2N = N(1 + 4a^2) \quad (73)$$

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

### 3.3.2 Distance

On va maintenant chercher à déterminer une expression de la distance de Bhattachayya entre deux distributions du type décrit au chapitre précédent. La distance de Bhattacharya 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  $(\Sigma_1, n_1)$  et que  $n_2\hat{\Sigma}_2$  suive une distribution de Wishart de paramètres  $(\Sigma_2, n_2)$ , peut se calculer comme suit :

$$B_{12} = \int_S \sqrt{p_1(S)p_2(S)} dS \quad (74)$$

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 :

$$B_{12} = \frac{n_1^{\frac{pn_1}{4}} n_2^{\frac{pn_2}{4}} |\Sigma_{12}|^{\frac{n_1+n_2}{4}}}{|\Sigma_1|^{\frac{n_1}{4}} |\Sigma_2|^{\frac{n_2}{4}}} \quad (75)$$

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 :

$$B_{12} = \frac{|\tilde{\Sigma}_{12}|^{\frac{n}{2}}}{|\Sigma_1|^{\frac{n}{4}} |\Sigma_2|^{\frac{n}{4}}} \quad (76)$$

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

On a donc, grâce à l'équation (75), une expression simple de la distance de Bhattachayya entre deux distributions de  $\hat{\Sigma}$  pour deux classes différentes.

### **3.3.3 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 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. A partir d'un espace vide, on augmente cet espace pas à pas 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.

## 4 Résultats

### 4.1 Algorithme de classification

On a brièvement expliqué l'algorithme de classification au chapitre 2.2. Dans l'équation (42), on note que pour chaque classe  $l$ , on a deux lois de probabilités,  $P_l(X_r^s = x_r^s)$  et  $P_l(X_t^s = x_t^s)$ , que l'on note  $P_l(x_r^s)$  et  $P_l(x_t^s)$ .

La première loi est celle des pixels de radiométrie, directement mesurés par l'instrument. Cette loi est modélisée par une loi normale.

En ce qui concerne la deuxième loi, on dispose déjà d'une approximation de la loi conditionnelle  $P_l(x_t^s/a)$ . Pour la loi du vecteur  $a$ ,  $p(a)$ , on remarque qu'il est difficile de modéliser simplement cette loi, puisque le vecteur est soit issu d'une optimisation numérique sur laquelle on a peu de contrôle (MV), soit le quotient de deux fonctions de variables aléatoires (les matrices de corrélation) dont les lois ne sont pas des lois simples. La figure 10 représente des exemples d'histogramme du paramètre d'interaction  $a$  calculé pour un modèle isotrope en 4-connexité, sur des images de synthèse, et sur des images réelles AVIRIS de zones urbaines.

On remarque que les valeurs de  $a$  sont très concentrées autour des valeurs réelles, ou du moins que la variance est assez faible, spécialement pour les estimations faites sur des images réelles, car elles sont très proches de la limite du domaine. Aussi va-t-on confondre la probabilité conditionnelle avec la probabilité intégrée sur le domaine de  $a$ .

Revenons sur la règle de décision (42). Si l'on prend deux fois l'opposé de la log vraisemblance, il vient :

$$\hat{\lambda}_s = \arg \min_{l \in \{0, L-1\}} (-2 \log P_l(x_r^s) - 2 \log P_l(x_t^s)) \quad (77)$$

Or :

$$-2 \log P_l(x_r^s) = (x_r^s)^t \Sigma_l^{-1} x_r^s + \log |\Sigma_l| \quad (78)$$

Et l'on a fait l'approximation suivante :

$$-2 \log P_l(x_t^s) = \text{tr} \left( \Sigma_l^{-1} \hat{\Sigma} \right) - (n_l - p - 1) \log \left| \hat{\Sigma} \right| + n_l \log |\Sigma_l| \quad (79)$$

où  $x_t^s$  est le résultat de la réorganisation de  $\hat{\Sigma}$  en un vecteur colonne, et  $n_l$  le nombre de degrés de liberté tel que défini au paragraphe 3.3.1.

Par conséquent, la règle de décision finale est la suivante :



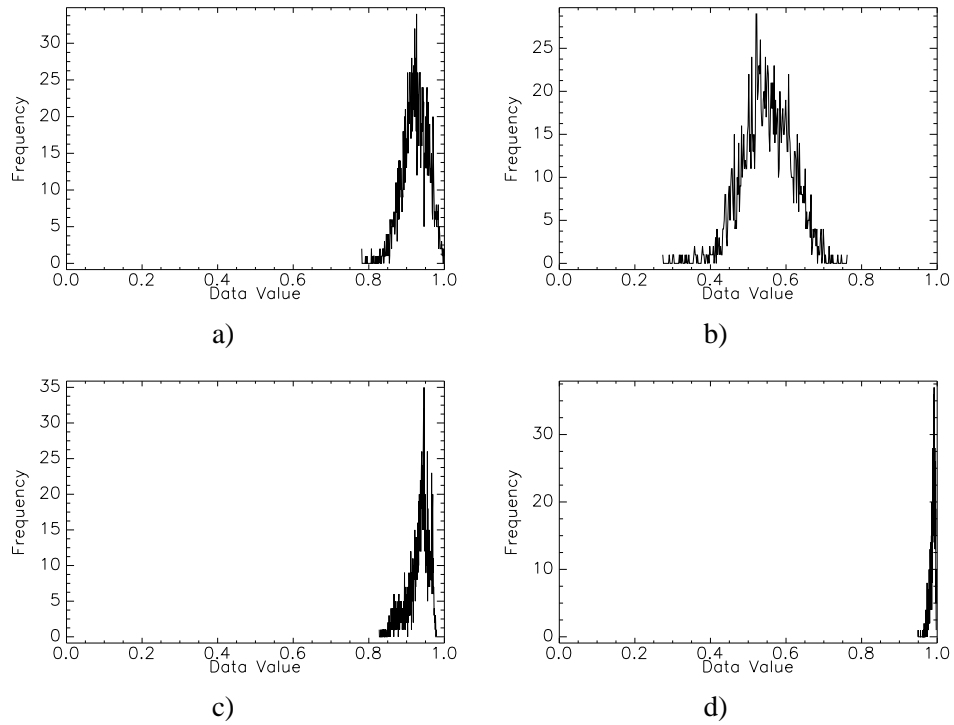


FIG. 10 – Histogrammes du paramètre  $a$  : a) texture de synthèse,  $a_{réel} = 0.9$ , b) texture de synthèse,  $a_{réel} = 0.6$ , c) et d) parties d'une image réelle AVIRIS.

$$\hat{\lambda}_s = \arg \min_{l \in \{0, L-1\}} \left( (x_r^s)^t \Sigma_l^{-1} x_r^s + \text{tr} \left( \Sigma_l^{-1} \hat{\Sigma} \right) - (n_l - p - 1) \log \left| \hat{\Sigma} \right| + (n_l + 1) \log |\Sigma_l| \right) \quad (80)$$

Il est à noter que, dans les applications sur des images réelles, on choisit le nombre de degrés de liberté bien inférieur à ce que l'on a calculé théoriquement, car les champs que l'on traite sont beaucoup plus irréguliers que de vrais MGMRF. Ceci a pour effet de rendre plus variables les mesures de  $\hat{\Sigma}$ . Or, quand le nombre de degré de liberté est grand, cela signifie que la distribution est plus "piquée" autour de la moyenne, et toute valeur qui s'en éloigne donne une probabilité très faible.

## 4.2 Résultats sur des images de synthèse

Nous avons d'abord testé l'algorithme 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 [22, 23]. On teste l'algorithme de classification sur des images ayant différents rapports signal sur bruit.

Les images sur lesquelles on teste les algorithmes sont les images de texture du paragraphe 1.5 : *colSynth1*, *colSynth2*, *colSynth5* et *colSynth10* de la figure 2 (dans l'ordre décroissant de S/B<sup>2</sup>). Chaque quart de l'image est constitué d'un carré de texture à 3 bandes spectrales possédant ses propres paramètres  $\Sigma$  et  $a$  ( $a$  est un scalaire).

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.

Les résultats sont présentés en figure 11. On constate que l'on n'obtient des bords francs que si les textures sont très bien 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 est simple : la figure 12 représente la projection dans un plan du nuage de point 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

---

2. Dans notre définition du 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.

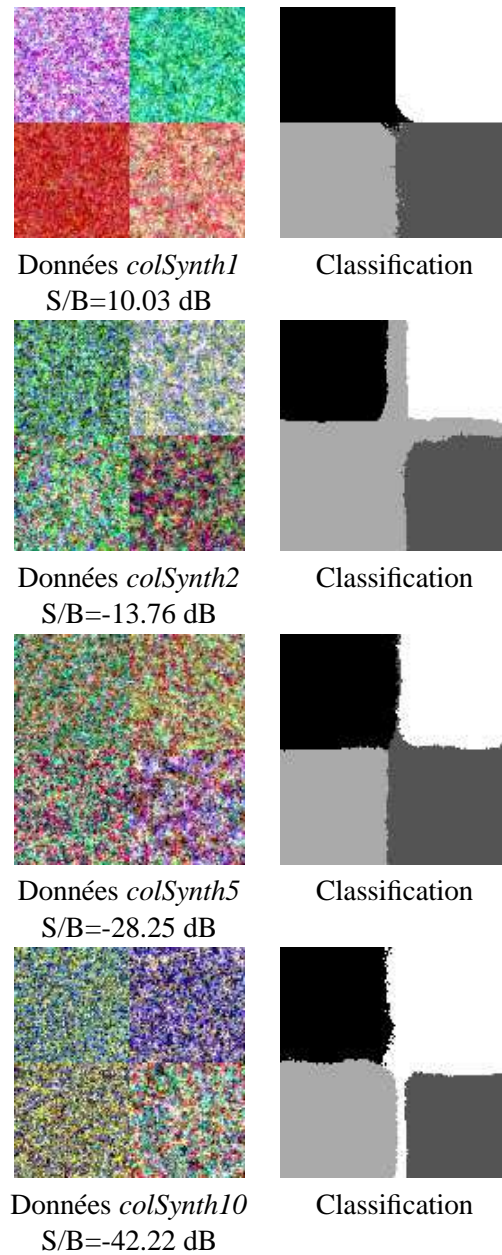


FIG. 11 – Données simulées à différents rapports  $S/B$  (colonne de gauche), simulation à partir des paramètres estimés pour chaque quart de l'image (colonne de droite).

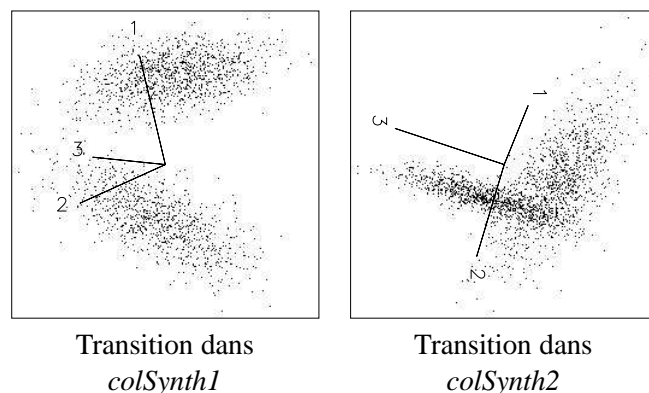


FIG. 12 – Projection du nuage de points correspondant à une zone de transition entre deux textures : a) les deux zones sont distinctes, b) les deux zones sont entremêlées. A noter que les points de vue, différents, ont été choisis parce qu'ils présentaient les nuages de la façon la plus séparée possible.

nuages par l'algorithme de *k-means* beaucoup plus délicate. Pour cette raison, les contours des textures sont mal détectés, et l'euristique est inefficace dans le cas de l'image *colSynth2*.

### 4.3 Résultats sur des images AVIRIS

Il reste donc à tester l'algorithme de classification proposé 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. La méthode présentée dans ce rapport est testée, ainsi que deux autres méthodes de référence. On appelle *ACP* –  $n$  la méthode de classification consistant à réduire l'image en gardant les  $n$  composantes principales, puis à classifier selon le maximum de vraisemblance (MV), en supposant les données gaussiennes sans interaction. On appelle *PP* –  $n$  la méthode proposée par Jimenez et Landgrebe consistant à rechercher  $n$  bandes par poursuite de projection, avec comme indice de projection la distance de Bhattacharyya calculée sur les bandes de radiométrie, suivie du même algorithme de classification que pour *ACP* –  $n$ . On compare également avec la méthode *PP* –  $n$  suivie d'une régularisation markovienne (par un modèle de Potts) obtenue par recuit simulé. On note cette dernière méthode *PPR* –  $n$ . Enfin, la méthode que l'on a présentée dans ce rapport sera notée *PP* –  $(n, m)$ , dans le cas où l'on a d'une part réduit la dimension (par la même méthode que dans le cas précédent) à  $n$  bandes, et d'autre part extrait les images de paramètres  $\hat{\Sigma}$  calculées sur  $m$  bandes.

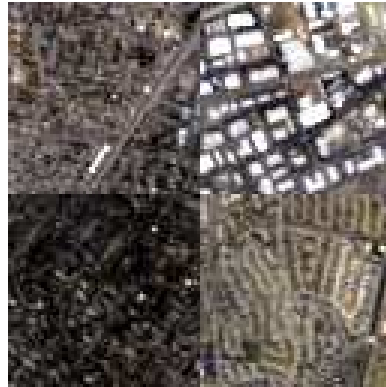


FIG. 13 – 3 bandes extraites d'un collage de 4 textures urbaines AVIRIS.

La figure 13 représente une mosaïque de quatre types différents de texture urbaine. On teste les différents algorithmes sur cette image dans un premier temps, pour avoir une idée de la qualité des résultats que l'on peut obtenir, même si les transitions entre textures qui sont introduites artificiellement sont bien plus abruptes que celles qui existent naturellement dans les images réelles. Donc sur ce type d'image, notre méthode aura un désavantage *a priori* du fait de l'utilisation de fenêtre d'analyse.

On réalise les tests sur un sous-ensemble de 25 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  $n$ , car celle-ci donne les meilleurs résultats.

On constate sur la figure 14, où les résultats de classification sur la mosaïque de textures AVIRIS sont donnés, que la méthode proposée donne de meilleurs taux de bonne classification, y-compris par rapport à la méthode avec régularisation par modèle de Potts. Pourtant, on remarque que les résultats obtenus par notre méthode reflètent 2 problèmes. D'une part, la solution n'est pas régulière, et d'autre part, la classification est mauvaise au niveau des zones de transition entre textures. Le premier problème peut être surmonté, par exemple en utilisant également une régularisation markovienne. Quant au deuxième problème, la mise au point d'une meilleure euristique permettant de détecter les bords reste nécessaire.

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 8. 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). On a défini deux sortes de tissus urbains différents, et trois classes non-urbaines.

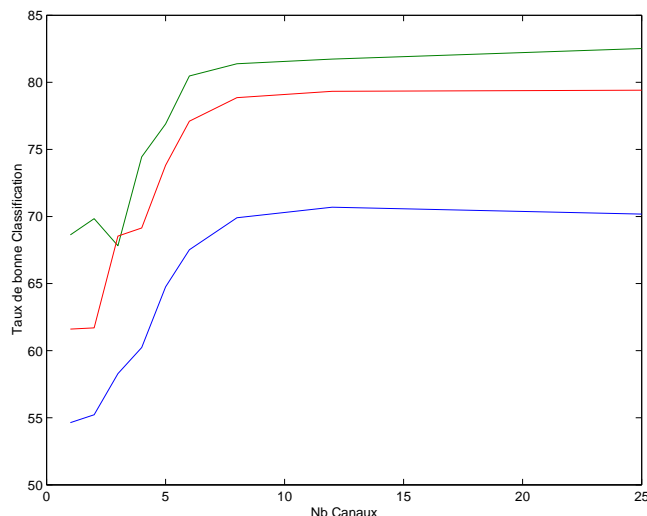


FIG. 14 – Classification par différentes méthodes de la mosaïque de textures AVIRIS. En bleu :  $PP - N$ , en rouge :  $PPR - N$ , en vert :  $PP - NN$ .

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 pour les autres. Les courbes de la figure 15 donnent l'évolution du coefficient  $Kappa$  ( $\kappa$ ) de la classification pour les différentes méthodes dont on a parlé :  $PP - n$ ,  $PP - (n, m)$ ,  $ACP - n$ ,  $PPR - n$ , quand  $n$  évolue. Le coefficient  $Kappa$  est une valeur comprise entre 0 et 1 permettant de mesurer la qualité d'une classification, 1 étant une classification sans erreur.

Pour des raisons de temps de calcul et de limitation de la capacité de mémoire, la classification  $PP - (n, m)$  est une classification  $PP - (n, 3)$ , où l'on a gardé une matrice de covariance conditionnelle calculée sur 3 bandes spectrales. En examinant la courbe de classification, on remarque avant tout que la courbe  $PP - (n, 3)$  est bien au-dessus des autres, quasiment quelle que soit la dimension de l'espace dans lequel la classification est faite. Les valeurs de  $Kappa$  sont assez élevées pour la méthode proposée, puisqu'elles restent supérieures à 0.8, alors que les autres méthodes ne passent pas au-dessus de 0.8, y-compris la méthode  $PPR - n$ , qui donne de meilleurs résultats que les deux autres méthodes non régularisées.

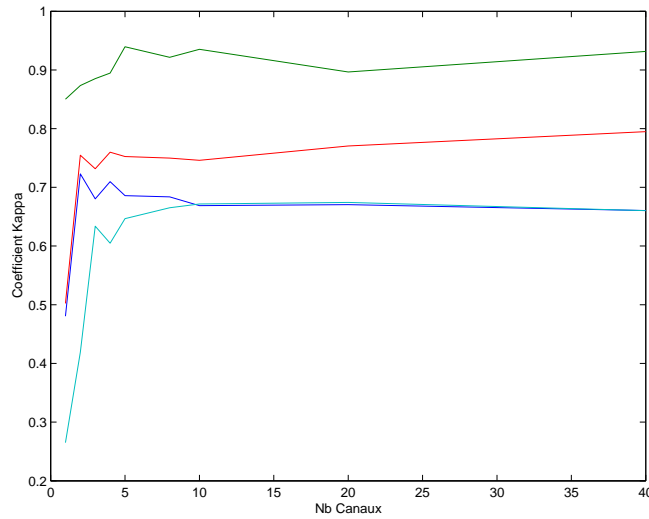


FIG. 15 – Classification par différentes méthodes de l’image AVIRIS. En bleu :  $PP - N$ , en rouge :  $PPR - N$ , en vert :  $PP - NN$ , en turquoise :  $ACP - N$ .

Comme le montre la figure 16, la classification par la méthode proposée apporte une sorte de régularisation fondée sur la texture, même si l’on trouve encore un nombre important de pixels isolés.

Enfin, on montre un dernier exemple de classification, cette fois 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. La première image (a) de la figure 17 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 des bandes visibles de l’image originale. Les trois autres images sont les résultats de classification par la méthode  $PP - 10$ , la méthode  $PPR - 10$ , et la méthode  $PP - 10,3$ . L’indice  $Kappa$  des classification par les deux premières méthodes est  $\kappa_{PP-10} = 0.55$  et  $\kappa_{PPR-10} = 0.62$ , alors que l’indice correspondant à la méthode proposée est nettement supérieur :  $\kappa_{PP-10,3} = 0.92$ . Malgré cette grande différence, comme dans les résultats précédents, cette dernière méthode donne des résultats visuels qui ne sont pas toujours satisfaisants en ce qui concerne les transitions entre classes (par exemple au niveau des bords du lac dans la partie en haut à gauche).

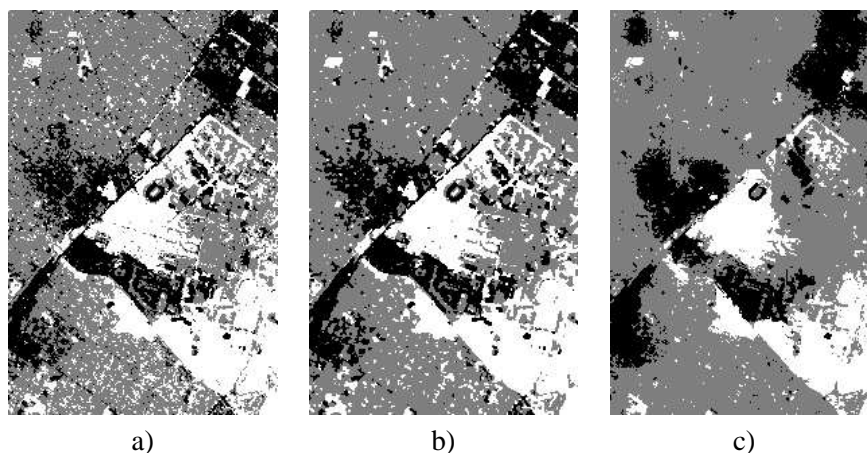


FIG. 16 – Classification par 3 méthodes de l’image AVIRIS : a)  $PP - 5$ , b)  $PPR - 5$ , c)  $PP - 5,3$ .

## Conclusion

Peu de méthodes ont été développées pour étudier la texture présente dans les images à grand nombre de composantes comme les images hyperspectrales. Nous proposons, dans ce rapport, une méthode utilisant le formalisme des champs de Markov permettant de prendre en compte la texture dans son ensemble, c’est-à-dire dans ses relations spatiales et spectrales. Nous avons présenté une version d’un champ de Markov gaussien multivarié, ainsi que les méthodes d’estimation des paramètres associés. Ce champ s’avère adapté à la modélisation des images auxquelles on s’intéresse. Pour l’analyse de texture urbaine et la classification, nous proposons un algorithme dans lequel les paramètres de texture sont extraits de l’image en même temps qu’un ensemble réduit de nouvelles bandes obtenues par projection des bandes initiales dans un sous-espace adapté. La classification est ensuite menée sur la concaténation de ces deux ensembles d’images, par application du critère du maximum de vraisemblance, suite à une approximation de la loi de probabilité des paramètres de texture. Les tests de classification menés sur des images issues du capteur aéroporté américain AVIRIS montre l’apport de l’analyse de texture sur des méthodes non-contextuelles même régularisées. Cependant, il apparaît que la méthode d’analyse par fenêtre glissante crée certains problèmes au niveau des frontières entre textures, même si l’on utilise une euristique destinée à détecter ces frontières. Une alternative plus efficace à cette euristique permettrait



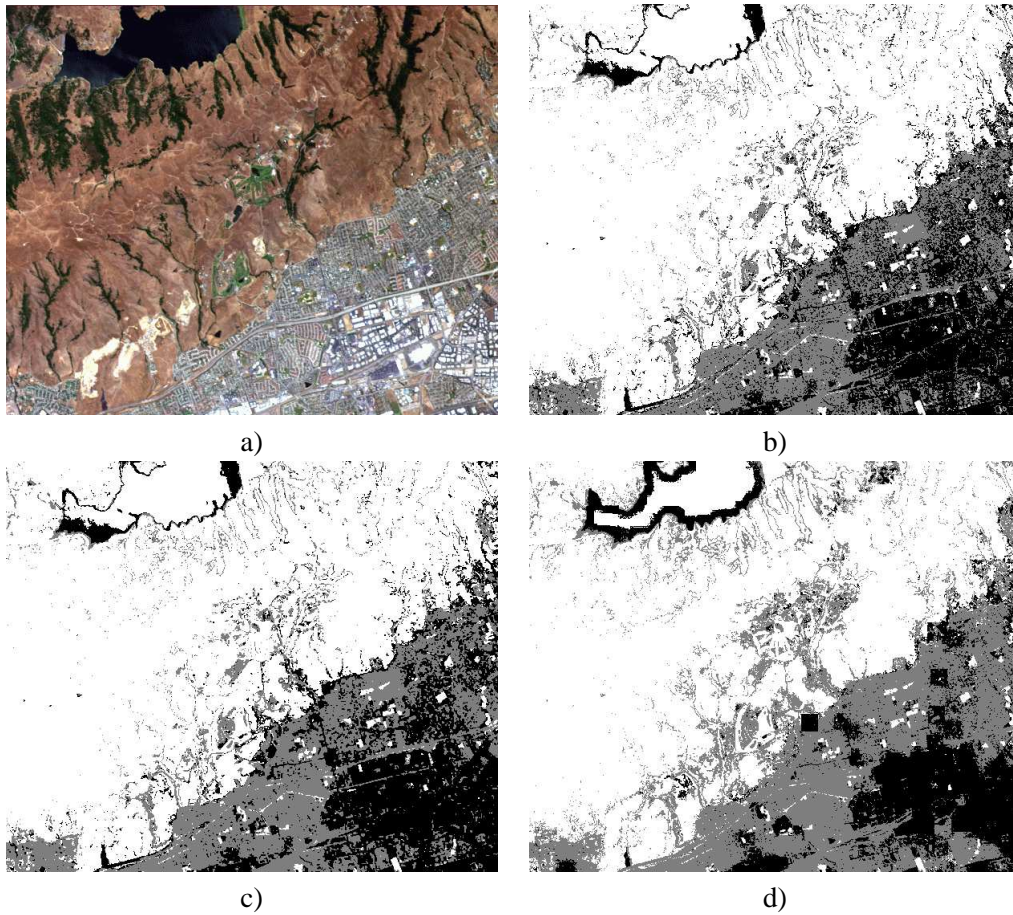


FIG. 17 – a) affichage en fausse couleur des canaux  $\{30,20,10\}$  de la scène AVIRIS, b) résultat de la classification par  $PP - 10$ , c)  $PPR - 10$ , d)  $PP - 10,3$ .

d'augmenter sérieusement le taux de bonne classification de la méthode proposée. Elle est actuellement à l'étude.

## **Remerciements**

Le premier auteur souhaite remercier la région Provence-Alpes-Côte d'Azur et Alcatel Space Industries pour le soutien financier apporté à sa thèse.

## A Annexe : 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).

Prenons comme exemple un modèle anisotrope en 4-connexité. On construit  $A$  comme suit :

$$B = \text{circulant}(1, -a_h, 0, \dots, 0, -a_h) \quad (81)$$

$$A = \text{circulant}(B, -a_v I_{N_h}, 0, \dots, 0, -a_v I_{N_h}) \quad (82)$$

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  $N_h \times N_v$ , rangée en un vecteur colonne  $X$ , les lettres calligraphiques représentant des images sous forme de matrices et les autres leur pendant vectoriel.

$$AX = \mathcal{A} * \mathcal{X} \quad (83)$$

Si l'on calcule la transformée de Fourier de chaque membre :

$$FAX = \tilde{\mathcal{A}} \times \tilde{\mathcal{X}} \quad (84)$$

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 :

$$\tilde{\mathcal{A}} \times \tilde{\mathcal{X}} = \Delta \tilde{\mathcal{X}} = \Delta FX \quad (85)$$

où  $\Delta$  est une matrice diagonale. Par conséquent :

$$FAX = \Delta FX \quad (86)$$

Et ceci quel que soit  $X$ , donc  $FA = \Delta F$ , et  $|A| = |\Delta|$ . Les éléments de  $\Delta$  sont donc les valeurs propres de  $A$ .

$$\lambda_{s_1, s_2} = \sum_{m=0, n=0}^{N_h, N_v} A_{m, n} \exp \left( 2\pi i \left( \frac{s_1 m}{N_h} + \frac{s_2 n}{N_v} \right) \right) \quad (87)$$

où les  $A_{m, n}$  sont les coefficients de la matrice  $A$ . Pour connaître  $|A|$ , il suffit donc de faire le produit des termes de la transformée de Fourier du masque  $\mathcal{A}$  :

$$|A| = \prod_{x=0, y=0}^{N_h, N_v} \sum_{m=0, n=0}^{N_h, N_v} A_{m, n} \exp \left( 2\pi i \left( \frac{xm}{N_h} + \frac{yn}{N_v} \right) \right) \quad (88)$$

## A.2 Dérivée

L'expression du logarithme du déterminant de  $A$  est donc :

$$\ln |A| = \sum_{x=0, y=0}^{N_h, N_v} \ln \left( \sum_{m=0, n=0}^{N_h, N_v} A_{m, n} \exp \left( 2\pi i \left( \frac{xm}{N_h} + \frac{yn}{N_v} \right) \right) \right) \quad (89)$$

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 :

$$\ln |A| = \sum_{x=0, y=0}^{N_h, N_v} \ln \left( 1 - \sum_{m=1, n=1}^{\frac{N_h}{2}, \frac{N_v}{2}} 2a_{m, n} \cos \left( 2\pi i \left( \frac{xm}{N_h} + \frac{yn}{N_v} \right) \right) \right) \quad (90)$$

Par conséquent, la dérivée peut s'écrire :

$$\frac{\partial}{\partial a_{m, n}} \ln |A| = \sum_{x=0, y=0}^{N_h, N_v} \frac{-2 \cos \left( 2\pi i \left( \frac{xm}{N_h} + \frac{yn}{N_v} \right) \right)}{1 - \sum_{m=1, n=1}^{\frac{N_h}{2}, \frac{N_v}{2}} 2a_{m, n} \cos \left( 2\pi i \left( \frac{xm}{N_h} + \frac{yn}{N_v} \right) \right)} \quad (91)$$

C'est-à-dire :

$$\frac{\partial}{\partial a_{m, n}} \ln |A| = -C_{m, n} \quad (92)$$

où  $C_{m, n}$  est le coefficient de corrélation théorique liée au modèle, c'est-à-dire  $(A^{-1})_{m, n}$ .

## Références

- [1] R. M. Haralick. Statistical and structural approaches to texture. *Proc of the IEEE*, 67(5):786–804, 1979.
- [2] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
- [3] S. M. Schweizer and J. Moura. Hyperspectral imagery : Clutter adaptation in anomaly detection. *IEEE, Trans. on Information Theory*, 46(5):1855–1871, 2000.
- [4] G.F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. on Information Theory*, 14(1):55–63, 1968.
- [5] X. Descombes. *Champs de markoviens en analyse d’images*. PhD thesis, ENST Paris, 1993.
- [6] G. Winkler. *Image analysis, random fields and dynamic Monte Carlo methods : a mathematical introduction*. Springer, 1995.
- [7] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc. B*, 36:192–236, 1974.
- [8] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE. Trans. on Pattern Anal. and Mach. Intell.*, 9:721–741, 1987.
- [9] G.L. Gimel’farb. *Image texture and Gibbs random fields*. Kluwer, 1999.
- [10] X. Descombes, M. Sigelle, and F. Prêteux. Estimating Gaussian Markov random field parameters in a nonstationary framework : application to remote sensing imaging. *IEEE Trans. on Image Processing*, 8(4):490–503, 1999.
- [11] A. Lorette, X. Descombes, and J. Zerubia. Urban areas extraction based on texture analysis through a Markovian modelling. *Int. Journal of Computer Vision*, 36(3):219–234, 2000.
- [12] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE, Trans. on Pattern Anal. Machine Intell.*, 17(10):939–954, 1995.
- [13] G. G. Hazel. Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection. *IEEE Trans. on Geoscience and Remote Sensing*, 38(3):1199–1211, 2000.
- [14] A. K. Jain. *Fundamentals of image processing*, chapter 2 : Two-dimensional systems and mathematical preliminaries. Prentice Hall, 1989.
- [15] J. Bennett and A. Khotanzad. Maximum likelihood estimation methods for multispectral random field image models. *IEEE, Trans. on Pattern Anal. and Machine Intell.*, 21(6):537–543, 1999.

- 
- [16] G. Rellier, X. Descombes, F. Falzon, and J. Zerubia. La poursuite de projection pour la classification d'image hyperspectrale texturée. Rapport de Recherche INRIA 4152, Mars 2001.
  - [17] P. Hall and K. Li. On almost linearity of low dimensionality projections from high dimensional data. *Ann. Stat.*, 21(2):867–889, 1993.
  - [18] G. H. Dunteman. *Principal Components Analysis*. Sage Inc., 1989.
  - [19] P. J. Huber. Projection pursuit. *The annals of Statistics*, 13(2):435–475, 1985.
  - [20] L. O. Jimenez and D. A. Landgrebe. Hyperspectral data analysis and supervised feature reduction via projection pursuit. *IEEE Trans. on Geoscience and Remote Sensing*, 37(6):2653–2667, 1999.
  - [21] T.W Anderson. *Introduction to multivariate statistical analysis*. J. Wiley and Sons, 1958.
  - [22] N. Metropolis, A.W. Rosenbluth, N.M. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chemical Physics*, 21:1087–1091, 1953.
  - [23] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated annealing : theory and applications*. D. Reidel, 1987.



---

Unité de recherche INRIA Sophia Antipolis

2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

---

Éditeur

INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)

<http://www.inria.fr>

ISSN 0249-6399