INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Extraction des Zones Urbaines Fondée sur une Analyse de la Texture par Modélisation Markovienne

A. Lorette — X. Descombes — J. Zerubia

N° 3423

Mai 1998

THÈME 3

Rapport de recherche

ISSN 0249-6399

Extraction des Zones Urbaines Fondée sur une Analyse de la Texture par Modélisation Markovienne

A. Lorette* , X. Descombes , J. Zerubia

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

Rapport de recherche n° 3423 — Mai 1998 — 49 pages

Résumé : Pour délimiter un masque urbain précis à partir d'une image satellitaire la seule information du niveau de gris est insuffisante. La plupart des méthodes font donc appel à une analyse de la texture de l'image. Nous nous sommes placés dans ce cadre. Dans une première étape, nous avons défini un nouveau paramètre de texture à partir d'un modèle markovien gaussien. Nous obtenons ce nouveau paramètre en calculant la variance conditionnelle de l'image dans huit directions. Ainsi, nous éliminons la mauvaise classification d'objets ayant une orientation privilégiée tels que les vignes et les serres par exemple. Dans une seconde étape, nous proposons un algorithme de *fuzzy Cmeans* modifié incluant un terme d'entropie et pour lequel le nombre de classes n'est pas fixé a priori. Cet algorithme nous permet d'obtenir une première classification de l'image. Enfin, nous régularisons l'image ainsi obtenue grâce à une modélisation par champs de Markov. Des résultats obtenus sur des simulations d'images SPOT5 fournies par le CNES sont présentés.

Mots-clés : Texture, Champs de Markov, *Fuzzy Cmeans*, Entropie, Urbain, Observation de la Terre

* Boursière CNES/INRIA

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Urban Areas Extraction Based on Texture Analysis through a Markovian Modelling

Abstract: Urban areas cannot be extracted from satellite images through only grey level information. Hence most methods analyze the texture of the image to discriminate between urban areas and non urban areas. We define a new texture parameter derived from a Markovian Gaussian model. This new parameter takes into account the variance of the image in eight directions. Consequently it copes with the misclassification of objects with a privileged orientation like vineyards or greenhouses for instance. Afterwards we develop a modified fuzzy Cmeans algorithm including an entropy term. The advantage of such an algorithm is that the number of classes does not need to be known a priori. By applying this modified fuzzy Cmeans algorithm on the parameter image we obtain a first classification. Finally we regularize the segmented image by using a Markov random field modelling. Some results on SPOT5 simulated images are presented. These images are provided by the CNES (French Space Agency).

Key-words: Texture, Markov Random Field, Fuzzy Cmeans, Entropy, Urban Area, Earth Observation

Table des matières

1 Introduction

La connaissance du milieu urbain passe entre autre par une appréciation aussi précise que possible de la ville en tant qu'entité physique [30]. Dans cette optique, les techniques de traitement d'image peuvent apporter une aide précieuse. Cette connaissance de la ville peut être abordée de deux manières différentes.

La première consiste à considérer la ville comme un seul et même objet par opposition aux milieux non urbains (champs, forêts, mer ...). Le but est alors d'extraire un masque urbain. Nous travaillons dans cette optique. C'est également l'objectif du projet EUROSTAT intitulé DÉLIMITATION DES AGGLOMÉRATIONS URBAINES EUROPÉENNES PAR TÉLÉDÉTECTION [1].
La seconde approche vise à obtenir une connaissance plus précise du milieu urbain lui même, en distinguant différents types de tissus à l'intérieur même de la ville [23]. Plusieurs nomenclatures [2] ont été définies à ce sujet en fonction des besoins des utilisateurs. Toutefois, il faut souligner "les problèmes liés à l'incompatibilité des nomenclatures urbaines en vigueur par rapport à l'information biophysique fournie" ([30] page 101). Pour y remédier il faut que "l'image devienne un support de réflexion accepté en fonction de ce qu'il a à fournir et non en fonction de ce que les usagers attendent de lui" ([30] page 123). Ces propos montrent bien toute la difficulté qu'il y a à trouver un cadre de travail précis et commun dans le domaine de la caractérisation de l'occupation des sols.

Les deux approches présentées ci-dessus sont en fait le résultat d'une réalité urbaine duale [30]. En effet, il existe d'une part une certaine hétérogénéité due à la variété des matériaux de construction qui va permettre de distinguer l'urbain du non urbain (plus homogène). D'autre part, l'existence d'une certaine homogénéité locale va nous permettre de distinguer différents types de tissus urbains. Cette dualité nous amène à penser que les techniques utilisées pour distinguer les zones urbaines des zones non urbaines seront différentes de celles utilisées pour caractériser la ville intrinséquement. Il faut certainement envisager un traitement en deux étapes.

Les méthodes utilisées jusqu'à present dans le domaine de l'extraction d'un masque urbain à partir d'images satellitaires optiques sont pour la plupart fondées

1. Pour plus d'information le lecteur pourra consulter le site Web suivant: http://www.city.cict.fr:8001/euro1.htm
2. SPOT image: http://www.spotimage.fr/francais/news/thema/classif.htm et EUROSTAT: http://www.city.cict.fr:8001/nomencl.htm

sur une analyse de la texture de l'image. En effet, la seule information apportée par le niveau de gris n'est pas suffisante.
Hormis l'analyse de la texture, on rencontre également des méthodes fondées sur une analyse par morphologie mathématique. Ainsi, M. Serendero [27] utilise la transformation du chapeau haut de forme (image originale soustraite à son ouverture) afin d'extraire des amas de pixels traduisant la présence d'une ville. Pour colmater les zones urbaines elle applique ensuite un filtre alterné séquentiel (suite d'ouvertures et de fermetures de taille croissante). Son but est d'obtenir une localisation grossière de la ville et non pas une extraction précise.
Le CNES a, quant à lui, collaboré avec plusieurs laboratoires de recherche en vue de développer des méthodes d'extraction des zones urbaines[3]. Les approches proposées font appel aux opérateurs de contours (collaboration avec l'IRIT), à la morphologie mathématique [27] et aux réseaux neuromimétiques [31] (collaborations avec l'INRIA Sophia-Antipolis).

Les applications de ces études sont nombreuses. On citera notamment le suivi de l'aménagement du territoire en “temps réel” (principalement dans les pays en voie de développement où l'expansion urbaine est souvent anarchique et rapide), l'analyse de l'évolution des surfaces urbaines, l'observation des populations urbaines (étude de QUITO par l'ORSTOM: “une meilleure connaissance de l'espace urbain grâce à l'information apportées par les images satellitaires constitue un facteur important d'amélioration d'un système d'enquête démographique par sondage”), l'étude de l'occupation des sols en vue de la mise à jour cartographique [19]. La cartographie représente 41% des applications des images SPOT. Dans les applications plus récentes nous pouvons citer l'aide à la mise en place de réseaux de radiomobile en télécommunication [25].

L' objectif du travail présenté est d'extraire un masque urbain précis à partir d'une image satellitaire.
Nous proposons à cet effet dans un premier temps un nouveau paramètre de texture. Il est caractéristique de la texture urbaine et est défini à partir d'une modélisation markovienne. Il tient compte de différentes orientations (en l'occurrence 8).
Dans une seconde étape, nous proposons une version modifiée de l'algorithme du *fuzzy Cmeans* incluant un terme d'entropie. Ceci dans le but de segmenter l'image du paramètre de texture extrait. La modification provient de la prise en compte d'un terme d'entropie dans le critère à minimiser. L'avantage de ce nouvel algorithme est

3. http://www-projet.cst.cnes.fr:8090/agglomeration/menu.html

qu'il détermine le nombre de classe optimal automatiquement.
Enfin, nous régularisons l'image segmentée à l'aide d'un champ de Markov.

2 Analyse de la texture urbaine

2.1 Introduction

Les travaux effectués à partir d'images satellitaires portent en majorité sur l'étude des forêts [15], des glaces [16], de l'agriculture [29]. La segmentation puis la classification d'images satellitaires contenant des zones urbaines sont particulièrement difficiles à réaliser. La plupart du temps, l'information apportée par le seul niveau de gris est largement insuffisante[4]. Parmi les diverses approches utilisées la plupart font appel à des paramètres de texture. Dans la majorité des nouveaux travaux sur l'extraction de l'urbain par une analyse de la texture les auteurs ne proposent pas de nouveaux paramètres. La nouveauté provient plutôt des algorithmes de segmentation ou d'estimation des paramètres, sachant que ces derniers sont classiquement les mêmes (issus des matrices de coocurrence, indice de végétation ...).

Ainsi, C. Gouinaud dans sa thèse [13] propose un comparatif des divers paramètres de texture utilisés pour la classification d'images satellitaires optiques. Il montre bien qu'il n'existe pas de paramètre "idéal" mais plutôt un ensemble de paramètres "utiles". Certains d'entre eux étant aptes à distinguer relativement correctement l'urbain du non urbain (indice de température [4]) mais échouant lors de la distinction des différents types d'urbain, d'autres ayant des capacités inverses (indice de taille), d'autres encore échouant dans tous les cas (indice de brillance).

Parmi les travaux effectués dans ce domaine on citera ceux réalisés par S. Houzelle [17] qui fusionne l'information obtenue à l'aide de deux capteurs (SPOT et SAR). En effet, les agglomérations ont la particularité d'apparaitre de manière brillante sur les images radar. Ainsi, par un simple seuillage, il peut déterminer des masques d'apprentissage pour le calcul des paramètres des zones urbaines. Puis, par des considérations de distance par rapport aux centres de ces zones, il détermine des masques d'apprentissage pour les paramètres des autres zones. La classification est réalisée à l'aide d'un simple classifieur gaussien multivarié classique. Il utilise quatre attributs de texture (deux liés à l'histogramme local, une mesure locale du laplacien et une

4. "*la couleur* ou l'aspect de [l'image satellitaire], le ton, ne peuvent être pris en considération, seuls. La réflectance d'un objet ne permet pas en général d'en déterminer sa nature." [30] page 91.

mesure d'orientation du gradient).
Haralick [14] a proposé des paramètres calculés à partir de la matrice de coocurrence (entropie, énergie, corrélation ...). Il les a utilisés pour segmenter des images aériennes ainsi que des images satellitaires LANDSAT contenant une ou plusieurs classes d'urbain. Cependant, C. Gouinaud [13] ainsi que Baraldi et Parmiggiani [1] soulignent l'inadéquation de ces paramètres avec l'analyse du milieu urbain sur des images SPOT. Ceci est dû au fait que les images SPOT (notamment les images du futur capteur SPOT5) ont une bien meilleure résolution que les images LANDSAT et donc, qu'à cette résolution, la texture urbaine est d'échelle variable.
D'autres approches sont basées sur la modélisation de l'image. Les paramètres du modèle lorsqu'ils sont estimés sur la zone urbaine constituent alors les paramètres de texture [4]. Parmi les modélisations de l'image nous citerons l'approche markovienne qui est celle que nous avons utilisée dans nos travaux. En ce qui nous concerne, nous considérons comme seule source de données une image SPOT.

2.2 Champs de Markov

2.2.1 Introduction

Les champs de Markov sont largement utilisés en analyse d'image [8][11]. Nous les avons utilisés à deux reprises dans notre travail. Dans une première étape, nous modélisons la texture de l'image par un champ markovien 4 connexe (paragraphe 2.3), ce qui nous permet de définir un paramètre caractéristique de la texture urbaine. Dans une dernière étape, la modélisation markovienne nous permet d'introduire de l'information contextuelle et donc de segmenter notre image (paragraphe 4.1.1).

2.2.2 Définition

Les champs de Markov [11][32] permettent de modéliser des propriétés globales en utilisant des contraintes locales.
Nous notons S ($card(S) = N$) le réseau des sites de l'image et $V = \{Vs | s \in S\}$ le système de voisinage (Vs est le voisinage du site s).
Nous notons $X = \{X_1, ..., X_N\}$ une famille de variables aléatoires définies sur le réseau S. Chaque variable aléatoire est à valeur dans Ω. Nous notons $X_i = x_i$ pour signifier que la variable aléatoire X_i prend la valeur x_i ($x_i \in \Omega$). De même nous notons $X = x$, $x \in \Omega^N$, pour signifier l'événement suivant: $(X_1 = x_1, ..., X_N = x_n)$. x, appelé configuration, correspond à une réalisation du champ aléatoire X. Cette configuration a une certaine probabilité que nous notons $P(X = x)$. X est appelé

champ de Markov par rapport au système de voisinage V s'il vérifie les 2 propriétés suivantes :

- $\forall x \in \Omega^N$, $P(X = x) > 0$ (positivité)
- $\forall s \in S$ et $\forall x \in \Omega^N$, $P(X_s = x_s | X_r = x_r, r \in S - \{s\}) = P(X_s = x_s | X_r = x_r, r \in Vs)$ (propriété markovienne)

Dans la suite, pour faciliter les notations, nous confondons X et x. $P(X = x)$ est alors notée $P(X)$.
Le théorème de Hammersley-Clifford [2][11] fait le lien avec les champs de Gibbs. Ce théorème permet de faire le lien entre la propriété de localité des champs de markov (propriété markovienne) et la propriété de globalité des champs de Gibbs (distribution de Gibbs) [2][20]. X est un champ de Markov par rapport au système de voisinage V si et seulement si sa distribution $P(X)$ est une distribution de Gibbs, i.e. :

$$P(X) = \frac{\exp(-E(X))}{Z} \tag{1}$$

$Z = \sum_{X \in \Omega^N} E(X)$ est une constante de normalisation appelée fonction de partition.

$E(X)$ est une fonction d'énergie définie comme suit :

$$E(X) = \sum_{cl \in CL} V_{cl}(X) \tag{2}$$

cl est une clique, i.e. un ensemble de sites voisins deux à deux ou un site singleton.
CL est l'ensemble des cliques liées au système de voisinage considéré.
$V_{cl}(X)$ est une fonction de potentiel.
L'énergie globale E(X) s'écrit donc comme une somme d'énergies locales (ou somme de potentiels).

2.3 Paramètre de texture

Dans cette partie nous présentons un nouveau paramètre caractéristique des textures urbaines. Ce paramètre est défini à partir des travaux de X. Descombes [4].

2.3.1 Modèle 2D isotrope

a. Le modèle

X. Descombes propose dans [4] un nouveau paramètre de texture utilisé afin d'extraire un masque urbain. Ce paramètre, appelé température et noté T, est fondé sur

la modélisation de l'image par un modèle markovien gaussien 4 connexe (équation 3). Les zones urbaines qui sont des zones à forte variance et peu corrélées ont une température plus élevée que les zones de champs.
L'intérêt du paramètre proposé est qu'il est directement lié à la définition d'un modèle markovien gaussien 4 connexe. En ce qui concerne le modèle isotrope, X. Descombes montre dans un premier temps que la loi conditionnelle $P(Xs|Xr \in Vs)$ ne dépend que de la moyenne m_s des 4 voisins ce qui favorise la robustesse des estimateurs (équation 4).

$$P(Xs|Xr \in Vs) = \frac{1}{Z_{Vs}} \quad \exp\Big(- \beta\Big(\sum_{r \in V_s}(Xs - Xr)^2 + \lambda(Xs - \mu)^2\Big)\Big) \tag{3}$$

$$= P(Xs|m_s) \tag{4}$$

$\beta = \dfrac{1}{T}$, λ et μ sont les paramètres du modèle.
Vs est le voisinage du site s.
Z_{Vs} est la constante de normalisation.

Dans un second temps, il montre que la probabilité conditionnelle locale est une loi normale définie comme suit :

$$P(Xs|Xr \in Vs) \quad \equiv \quad \mathcal{N}\left(\frac{4m_s + \mu\lambda}{4 + \lambda}, \frac{1}{2\beta(4 + \lambda)}\right) \tag{5}$$

La variance conditionnelle $\sigma^2_{P(Xs|m_s)} = \dfrac{1}{2\beta(4+\lambda)}$ est estimée à partir de la méthode des queues de comète (cf. paragraphe suivant).
Toutefois, utilisé seul pour extraire un masque urbain, ce paramètre amène à la confusion des vignes, des serres et des villes (FIG. 1 et 2). Les figures 1 et 2 montrent bien l'intérêt que l'on a à considérer séparément les différentes directions. En effet, la serre n'apparait pas sur l'image 2(b) (qui correspond à la direction NOo qui est la direction dans laquelle est orientée la serre et qui est la direction de plus faible variance) mais apparait nettement sur l'image 2(a). D'autre part, sur l'image 2(c) du paramètre proposé par X. Descombes cette serre apparait également. C'est pourquoi, dans la suite de ce rapport, nous allons proposer une nouvelle approche tenant compte de différentes directions (en l'occurrence huit). Ces 8 directions sont représentées sur la figure 4.

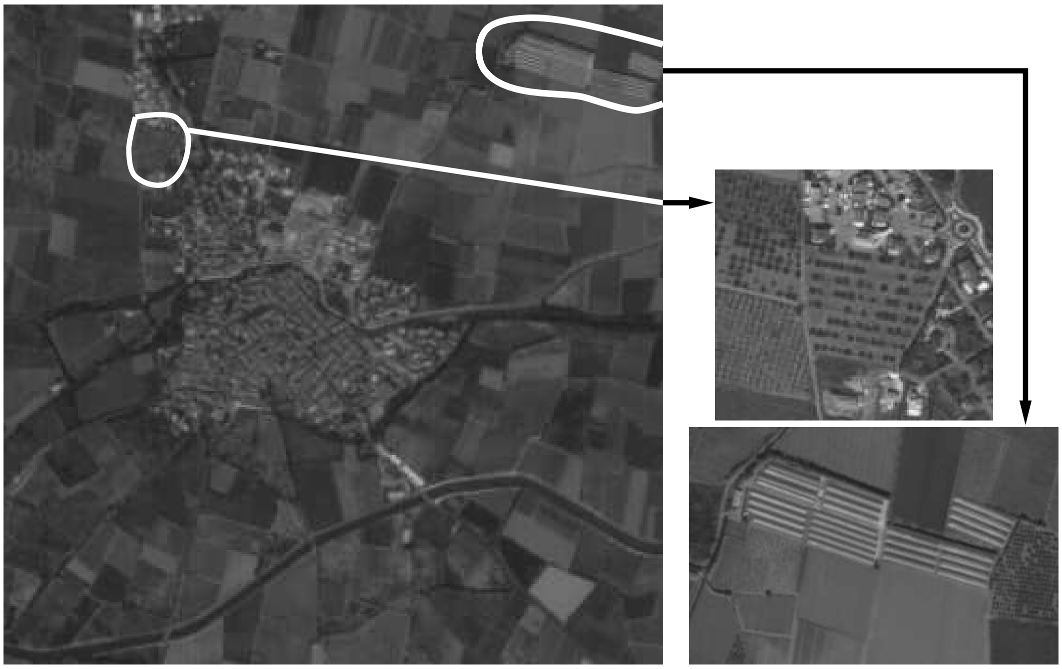


FIG. 1 – *Image de Toreilles, zone de serres et vigne*

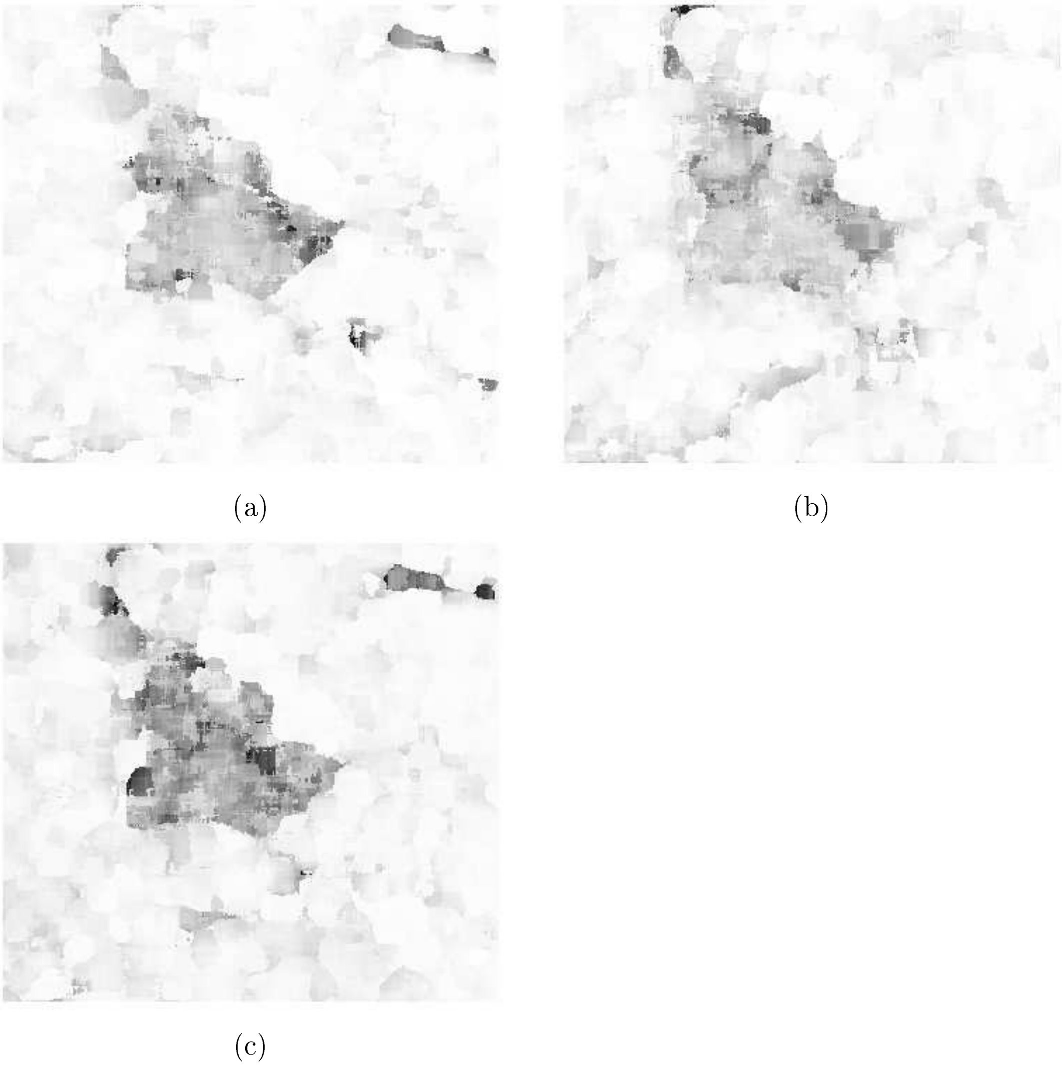


FIG. 2 – *Images de l'estimée du paramètre de texture considéré (Mode vidéo inversé): (a) direction N, (b) NOo et (c) modèle 2D isotrope*

b. La méthode d'estimation: Queues de comète

Pour estimer la variance conditionnelle nous employons la méthode des queues de comètes [4].

Le but est d'estimer ${\sigma^2}_{P(Xs|m_s)} = \dfrac{1}{2\beta(4+\lambda)}$.

Nous estimons donc la matrice des probabilités conditionnelles [6][7]. Pour ce faire, pour chaque site s nous incrémentons le point de la matrice dont les coordonnées correspondent respectivement à la valeur Xs du site et à la valeur m_s de la moyenne de ses voisins (FIG. 3). Puis pour chacune des distibutions $P(Xs|m_s)$ nous estimons la variance conditionnelle ${\sigma^2}_{P(Xs|m_s)}$ par l'estimateur empirique. L'estimation de la variance est effectuée sur une fenêtre centrée sur le pixel Xs.

Se pose alors le problème de la taille de la fenêtre qui doit être assez grande pour avoir des statistiques fiables et assez petite pour éviter au maximum de traiter des mélanges de texture. Dans cette méthode d'estimation le problème est en partie résolu en pondérant l'estimateur de la variance par le nombre d'échantillons qui ont servis à l'obtenir. Ainsi nous obtenons des bords francs (FIG. 2).

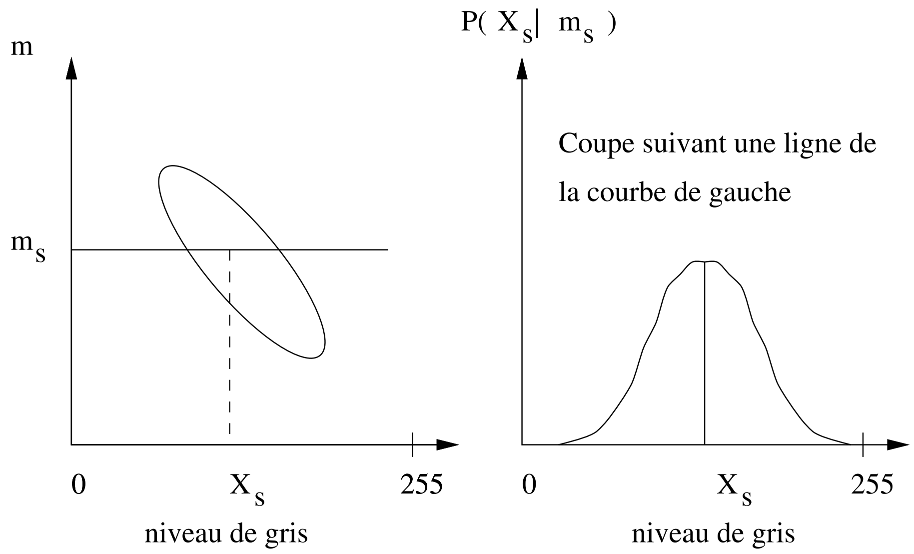


FIG. 3 – *Queue de comète*

2.3.2 Série de modèles 1D

En ce qui nous concerne, nous avons tenu compte de l'anisotropie en considérant une série de modèles 1D pris dans les 8 directions principales[5] (FIG. 4).
Dans ce cas la probabilité conditionnelle pour le modèle défini dans la direction d est la suivante :

$$\begin{aligned} P(Xs|Xr \in V{s}^{d}) &= P(Xs|{m_s}^{d}) \\ &= \frac{1}{Z_{V{s}^{d}}} \quad \exp\left(-\beta^{(d)}\left(\sum_{r\in {V_s}^{d}}(Xs-Xr)^2+\lambda^{(d)}(Xs-\mu)^2\right)\right) \\ &\equiv \mathcal{N}\left(\frac{2{m_s}^{d}+\mu\lambda^{(d)}}{2+\lambda^{(d)}}, \frac{1}{2\beta^{(d)}(2+\lambda^{(d)})}\right) \end{aligned}$$

d représente la direction considérée.
${m_s}^{d}$ représente la moyenne des 2 voisins du site s dans la direction d.

Ainsi, dans la direction (NOo/SEe), la probabilité conditionnelle locale est :

$$P(Xs|Xr \in Vs^{(d=NOo/SEe)}) = P(Xs|(X(NOo)+X(SEe))/2)$$

Nos paramètres de texture sont les variances conditionnelles estimées par la méthode des queues de comète dans les huit directions :

$$\sigma_d^2 = \frac{1}{2\beta^{(d)}(2+\lambda^{(d)})}$$

Comme le montre la figure 4, si l'on considère la direction (E/O), les pixels voisins sont situés à une distance 1 du pixel s. En revanche, si l'on considère la direction (NOo/SEe) les pixels voisins sont situés à une distance $\sqrt{5}$ du pixel s. Afin de corriger le biais introduit par ces distances et de pouvoir comparer les paramètres calculés dans les huit directions, nous avons normalisé les paramètres considérés.

5. il faut noter que les directions NOo et SEe n'en font qu'une

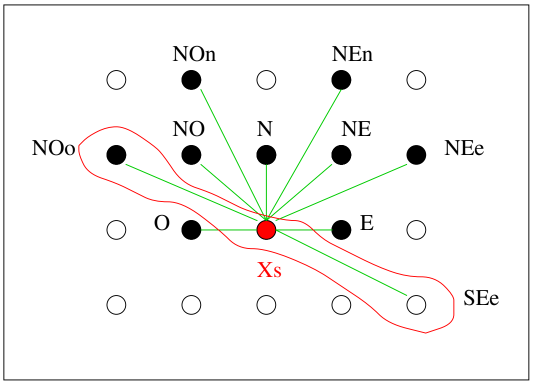


FIG. 4 – *Représentation des 8 directions prises en compte*

2.3.3 Correction des paramètres

Pour normaliser nous choisissons un modèle qui représente le réseau avec un pas d'échantillonnage de référence. Puis nous calculons les paramètres des différents modèles en fonction des paramètres du modèle sur ce réseau de référence.
Dans ce but, nous avons utilisé la méthode de renormalisation par décimation [12]. Cette méthode consiste à intégrer le modèle de référence par rapport aux pixels du réseau de référence que l'on souhaite supprimer pour obtenir le réseau fourni par les données, ce qui revient à calculer la loi marginale sur ce dernier.
Ainsi sur la figure 5 les paramètres β_1 et λ_1 (l'indice est égal au nombre de variables sur lesquelles on intègre) sont obtenus à partir des paramètres β et λ en intégrant le modèle de référence par rapport à un pixel sur deux du réseau de référence.

Comme nous cherchons à calculer les paramètres de modèles dont les pixels sont distants de 1, $\sqrt{2}$ ou $\sqrt{5}$ par la méthode ci-dessus, il faut approximer ces nombres non forcément rationnels par des fractions rationnelles afin que nous puissions intégrer sur un nombre entier de sites. Pour obtenir cette approximation, nous considérons la suite suivante :

$$U_{n+1} = \frac{U_n + \frac{a}{U_n}}{2} \tag{6}$$

Cette suite est obtenue par la méthode de Newton [26]. Ce dernier a proposé de déterminer les zéros d'une fonction $g(x)$ à l'aide d'un processus itératif. Ainsi, $\sqrt{a}$

n	$a = 2$	$a = 5$
0	1	1
1	$\frac{3}{2}$	3
2	$\frac{17}{12}$	$\frac{28}{12}$
3	$\frac{577}{408}$	$\frac{47}{21}$

TAB. 1 – *Premières itérations de la suite (donnée par l'équation 6) pour 2 valeurs de a: 2 et 5.*

est solution de l'équation $x^2 - a = 0$. Nous posons $g(x) = x^2 - a$. La suite définie de la manière suivante :

$$U_{n+1} = U_n - \frac{g(U_n)}{g'(U_n)} \tag{7}$$

a alors $\sqrt{a}$ comme point fixe.

Pour approximer les deux racines, nous conservons les fractions rationnelles qui offrent le meilleur compromis entre une approximation précise de $\sqrt{2}$ et $\sqrt{5}$ et un nombre de variables, sur lesquelles on intègre lors de la phase de décimation, le plus faible possible.
A l'itération $n = 2$ (TAB. 1), nous obtenons les résultats suivants :

$$\left|\sqrt{2} - \frac{17}{12}\right| < 0.1$$
$$\left|\sqrt{5} - \frac{28}{12}\right| < 0.1$$

Nous conservons donc les valeurs $\frac{17}{12}$ et $\frac{28}{12}$ pour approximer respectivement $\sqrt{2}$ et $\sqrt{5}$. Ces valeurs sont le meilleur compromis entre une bonne approximation et un

nombre réduit de variables sur lesquelles intégrer.
Nous posons $\epsilon = \dfrac{1}{12}$, alors :

$$\begin{aligned} 1 &= 12\epsilon \\ \sqrt{2} &\approx 17\epsilon \\ \sqrt{5} &\approx 28\epsilon \end{aligned}$$

Nous pouvons associer le couple de paramètres (β, λ) à ϵ (relativement au réseau de référence). Nous allons estimer les paramètres sur l'image (suivant la direction considérée). Puis nous allons rechercher les paramètres équivalents pour le réseau de pas ϵ.
Voici les résultats obtenus pour certaines de ces variables. Le détail des calculs pour le couple (β_1, λ_1) est proposé en annexe (cf. annexe A).
Dans ce cas ils sont plus simples à obtenir dans la mesure où les intégrales sont séparables.
Dans les autres cas, les calculs ont été effectués en utilisant la méthode de Laplace [24]. Lorsque $Q(x)$ est une forme quadratique l'intégrale sur $\mathbb{R}$ de $\exp(-Q(x))$ est proportionnelle à la valeur au col de cette fonction (i.e. à $exp(-Qmin)$). Les intégrales que nous calculons sont toujours de cette forme.

La loi jointe est la suivante :

$$P(x_1, x_2, ..., x_N) \quad \alpha \quad \exp\left(-\beta\left(\sum_{k=1}^{N}\lambda(x_k-\mu)^2 + \sum_{k=1}^{N-1}(x_k - x_{k+1})^2\right)\right) \tag{8}$$

Nous intégrons cette loi par rapport à 11, 16 puis 27 variables [6]. Les paramètres sont les suivants :

$$\begin{aligned} \lambda_{11} &= \lambda(\lambda+1)^2(\lambda+2)^2(\lambda+3)^2(\lambda+4)(\lambda^2+4\lambda+1)^2 \\ \lambda_{16} &= \lambda(\lambda^8+17\lambda^7+119\lambda^6+442\lambda^5+935\lambda^4+1122\lambda^3+714\lambda^2+204\lambda+17)^2 \\ \lambda_{27} &= \lambda(\lambda+2)^2(\lambda+4)(\lambda^3+7\lambda^2+14\lambda+7)^2(\lambda^3+5\lambda^2+6\lambda+1)^2 \\ &\quad * \ (\lambda^6+12\lambda^5+53\lambda^4+104\lambda^3+86\lambda^2+24\lambda+1)^2 \end{aligned}$$

6. Les calculs ont été effectués avec le logiciel Maple

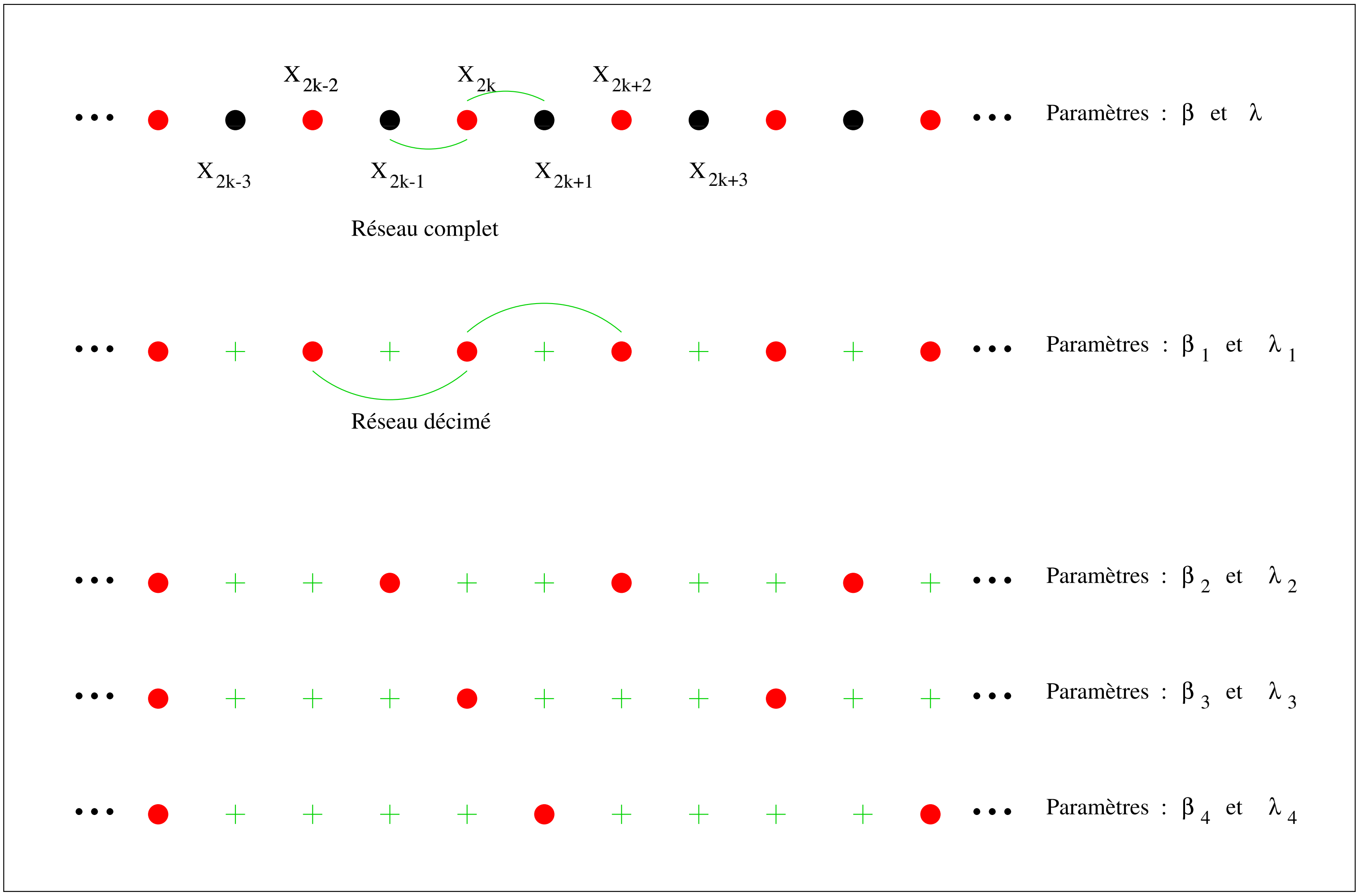


FIG. 5 – *Représentation du réseau complet et des réseaux décimés*

$$\beta_{11} = \frac{\beta}{P_{11}(\lambda)},\ \beta_{16} = \frac{\beta}{P_{16}(\lambda)} \text{ et } \beta_{27} = \frac{\beta}{P_{27}(\lambda)} \text{ avec :}$$

$$\begin{aligned}
P_{11}(\lambda) &= (\lambda+1)(\lambda+2)(\lambda+3)(\lambda^2+4\lambda+1)(\lambda^2+4\lambda+2)(\lambda^4+8\lambda^3+20\lambda^2+16\lambda+1)\\
P_{16}(\lambda) &= (\lambda^8+17\lambda^7+119\lambda^6+442\lambda^5+935\lambda^4+1122\lambda^3+714\lambda^2+204\lambda+17)\\
&* (\lambda^8+15\lambda^7+91\lambda^6+286\lambda^5+495\lambda^4+462\lambda^3+210\lambda^2+36\lambda+1)\\
P_{27}(\lambda) &= (\lambda+2)(\lambda^2+4\lambda+2)(\lambda^6+12\lambda^5+53\lambda^4+104\lambda^3+86\lambda^2+24\lambda+1)\\
&* (\lambda^{12}+24\lambda^{11}+252\lambda^{10}+1520\lambda^9+5813\lambda^8+14672\lambda^7+24648\lambda^6+27104\lambda^5\\
&\quad +18646\lambda^4+7344\lambda^3+1400\lambda^2+96\lambda+1)(\lambda^3+7\lambda^2+14\lambda+7)(\lambda^3+5\lambda^2+6\lambda+1)
\end{aligned}$$

Ainsi, si nous considérons par exemple la direction (d =NOo/SEe), les pixels voisins de Xs sont situés à une distance $\sqrt{5}$ de celui-ci.

La variance conditionnelle estimée sur l'image dans les directions (d =NOo/SEe, d =NOn/SEs, d =NEn/SOs, d =NEe/SOo) est :

$$\begin{aligned}
\sigma_d^2 &= \frac{1}{2\beta_{27}(2+\lambda_{27})}\\
&= \frac{P_{27}(\lambda)}{2\beta(2+\lambda_{27})}
\end{aligned}$$

La variance conditionnelle estimée sur l'image dans les directions (d =NE/SO, d =NO/SE) est :

$$\begin{aligned}
\sigma_d^2 &= \frac{1}{2\beta_{16}(2+\lambda_{16})}\\
&= \frac{P_{16}(\lambda)}{2\beta(2+\lambda_{16})}
\end{aligned}$$

La variance conditionnelle estimée sur l'image dans les directions (d =N/S, d =E/O) est :

$$\begin{aligned}
\sigma_d^2 &= \frac{1}{2\beta_{11}(2+\lambda_{11})}\\
&= \frac{P_{11}(\lambda)}{2\beta(2+\lambda_{11})}
\end{aligned}$$

En pratique, nous ne travaillons pas sur les couples (β, λ) mais sur les variances conditionnelles (qui sont des fonctions de β et λ). De plus, nous ne chercherons pas l'équivalent des paramètres par rapport au réseau de référence (correspondant à une distance de ϵ entre 2 sites voisins) pour les trois distances ($\sqrt{2}$, $\sqrt{5}$, 1). Mais nous allons chercher, pour les distances $\sqrt{2}$ et $\sqrt{5}$, l'équivalent de la variance conditionnelle relativement au réseau de pas 1. Ainsi nous conservons les estimées des variances conditionnelles dans les directions N/S et E/O et nous corrigeons les estimées dans toutes les autres directions. Le réseau de référence (correspondant à ϵ) facilite les calculs des couples (β, λ).

Dans les directions (d =NOo/SEe, d =NOn/SEs, d =NEn/SOs, d =NEe/SOo) la variance conditionnelle corrigée (i.e. ramenée au réseau de pas 1) est :

$$\begin{aligned} \sigma^2_{d\,corr} &= h(\lambda) * \sigma^2_d \quad \text{où} \\ h(\lambda) &= \frac{\frac{P_{11}(\lambda)}{(2+\lambda_{11})}}{\frac{P_{27}(\lambda)}{(2+\lambda_{27})}} \end{aligned}$$

Dans les directions (d =NE/SO, d =NO/SE) la variance conditionnelle corrigée (i.e. ramenée au réseau de pas 1) est :

$$\begin{aligned} \sigma^2_{d\,corr} &= g(\lambda) * \sigma^2_d \quad \text{où} \\ g(\lambda) &= \frac{\frac{P_{11}(\lambda)}{(2+\lambda_{11})}}{\frac{P_{16}(\lambda)}{(2+\lambda_{16})}} \end{aligned}$$

Les 2 fonctions $h(\lambda)$ et $g(\lambda)$ sont tracées en figure 6.

$$\begin{aligned} h(0) &= \frac{12}{17} \\ &\approx 1/\sqrt{2} \\ g(0) &= \frac{12}{28} \\ &\approx 1/\sqrt{5} \end{aligned}$$

$\sigma^2_{16}corr$ et $\sigma^2_{27}corr$ sont les paramètres corrigés, i.e. ramenés à une distance unité. Nous corrigeons nos paramètres avec $h(0)$ et$g(0)$ (ce qui est valable dans le cas où λ reste petit).

Le processus est le suivant :

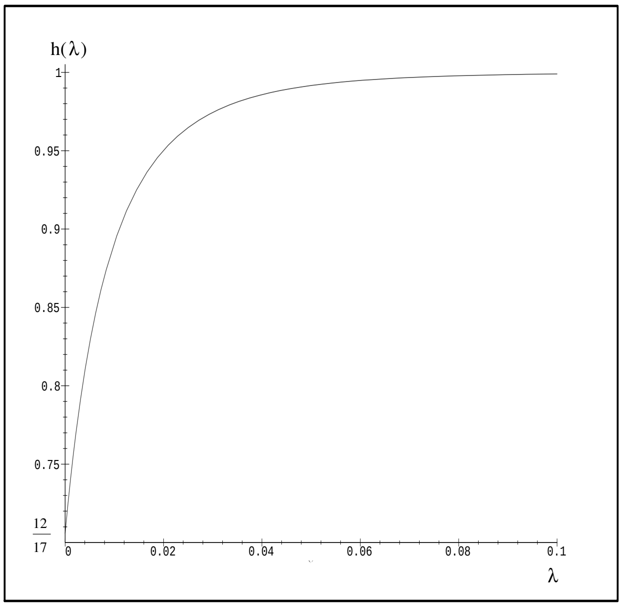
1. Nous estimons les variances conditionnelles dans toutes les directions à partir de l'image.

2. Nous corrigeons les variances conditionnelles par $h(0)$ dans les directions NE/SO et NO/SE et par $g(0)$ dans les directions NOo/SEe, NOn/SEs, NEn/SOs et NEe/SOo.

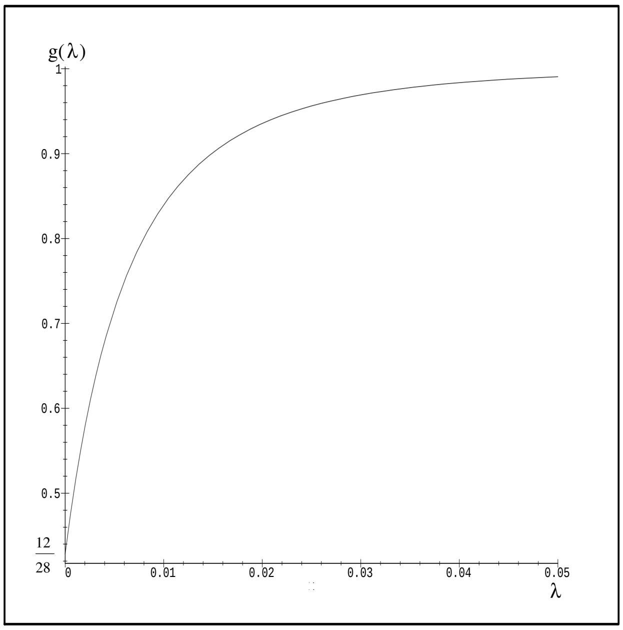
Afin de voir l'apport d'une telle correction, nous avons découpé dans une image une zone dans la ville telle que celle ci soit quasi isotrope (FIG. 7). Dans ce cas les distributions des huit paramètres devraient être proches. Nous avons donc représenté les histogrammes avant et après correction (FIG. 8, 9 et 10). Afin de comparer ces distributions entre elles, avant et après correction, nous avons calculé certaines valeurs caractéristiques (la moyenne, la variance, la valeur minimale, la valeur maximale) ainsi que la distance de Kolmogorov-Smirnov (notée d_{KS} dans la suite du texte) par rapport à la distribution de la variance conditionnelle dans la direction NE (TAB.2). Les directions E et N sont privilégiées (orientation des rues, des bâtiments). C'est pourquoi, nous ne comparons que les distributions correspondant aux 6 autres directions. Par ailleurs, dans le cas d'une isotropie parfaite, la distance calculée entre les distributions relatives aux directions NE et NO (réseau de même pas $\sqrt{2}$) devrait être nulle avant et après correction. Cette distance vaut dans notre cas 0.12 (TAB. 2). Cette valeur est une valeur de référence pour les autres d_{KS} que l'on doit obtenir après correction.
Les d_{KS} entre les distributions diminuent après correction et se rapprochent de la valeur de référence 0.12. De même, les autres valeurs caractéristiques (moyenne, variance) sont plus proches après correction. Ces observations montrent la nécessité d'une telle correction.

2.4 Résultats

L'image de départ est issue d'une simulation de données SPOT5 à 5m de résolution à partir d'une image aérienne de type push-broom (données fournies par le CNES).
En chaque pixel on estime donc la variance conditionnelle à partir de la méthode des queues de comète et ceci dans huit directions. La plupart des milieux urbains sont caractérisés par une forte variance dans toutes les directions. Par ailleurs les zones de champs, de forêt ou encore de mer présentent une faible variance dans toutes les directions. Enfin des constructions réalisées par l'homme telles que les serres, les hangars ou encore les routes, de même que les petits parcellaires ont une orientation



a



b

FIG. 6 – *(a): fonction* $h(\lambda)$ *et (b): fonction* $g(\lambda)$

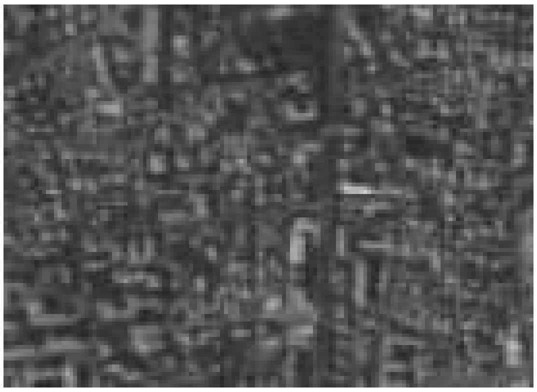
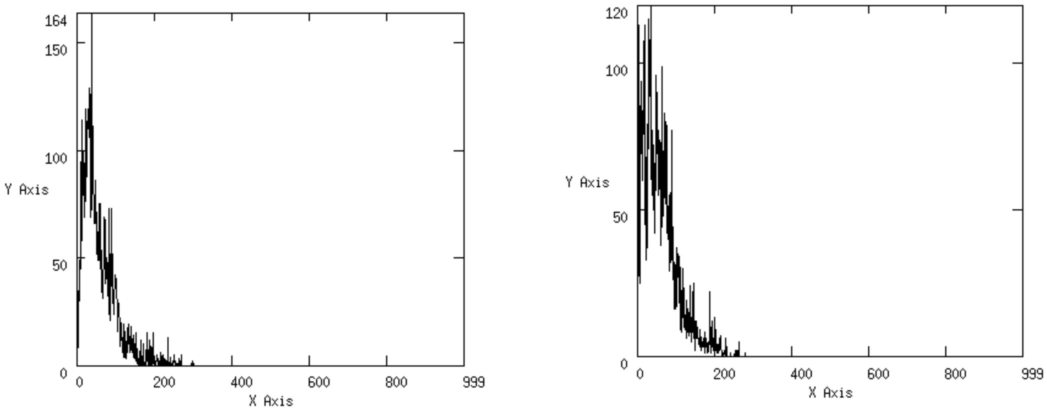


FIG. 7 – *Zone urbaine extraite de l'image d'Aix en Provence (simulation SPOT5 à 5 mètres)*



a b

FIG. 8 – *histogrammes des variances conditionnelles dans les directions (a): E et (b): N*

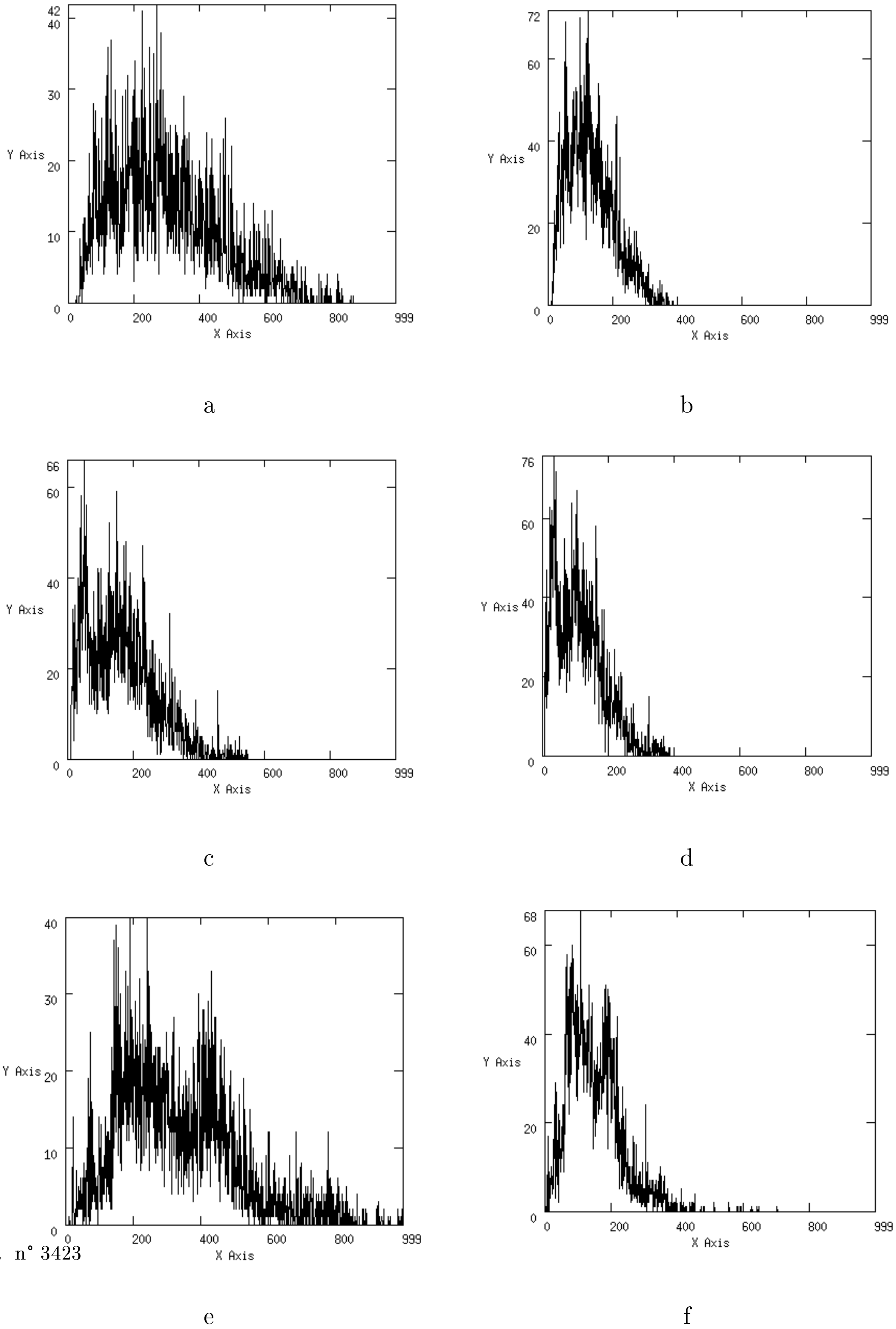
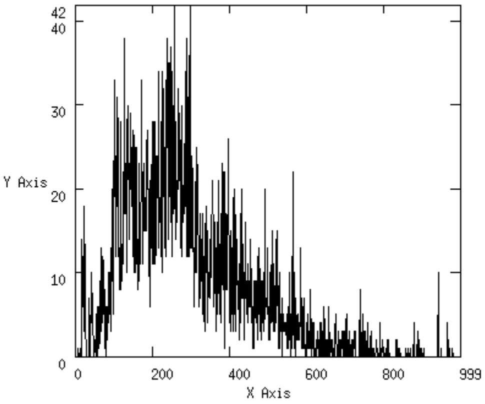
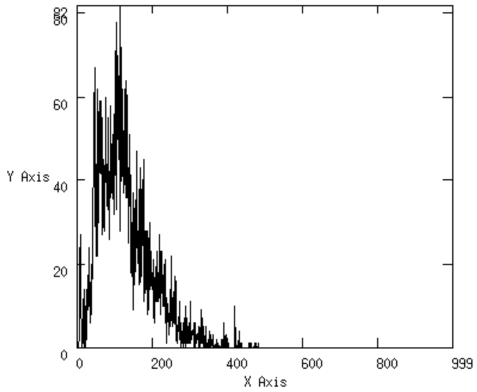


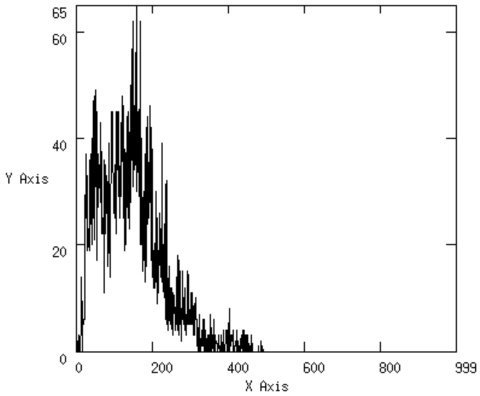
FIG. 9 – *histogrammes des variances conditionnelles dans les directions (a)(b): NEe, (c)(d): NE, (e)(f): NEn ((a)(c)(e): sans correction colonne de gauche, (b)(d)(f): avec correction colonne de droite)*



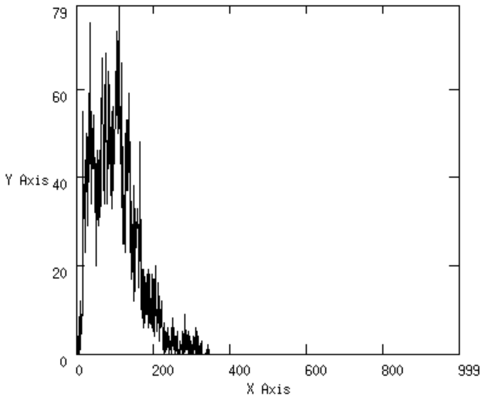
a



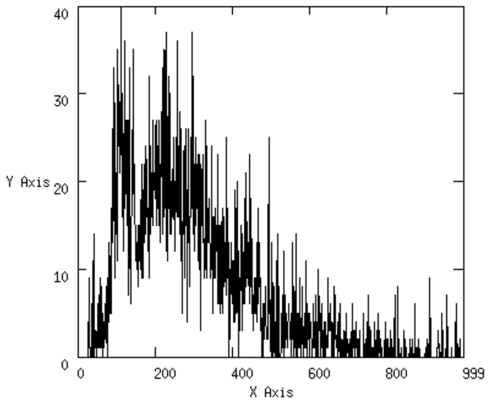
b

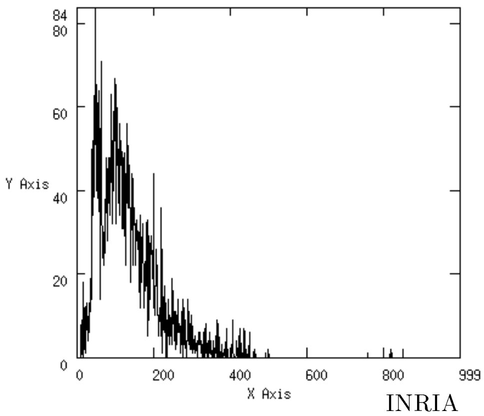


c



d





e

f

FIG. 10 – *histogrammes des variances conditionnelles dans les directions (a)(b): NOn, (c)(d): NO, (e)(f): NOo ((a)(c)(e): sans correction colonne de gauche, (b)(d)(f): avec correction colonne de droite)*

	avant correction					après correction				
	moy	var	max	min	d_{KS}	moy	var	max	min	d_{KS}
NE	163	10146	550	12	0	115	5073	389	8.5	0
NEe	299	25067	870	26	0.39	133	5013	389	12	0.12
NEn	335	31020	1576	12	0.46	150	6204	705	5.7	0.19
NOn	290	24979	1081	8	0.37	129	4975	483	4	0.14
NO	146	6799	492	6.8	0.12	103	3399	348	4.8	0.12
NOo	299	32470	1904	27	0.36	134	6494	852	12	0.14

TAB. 2 – *valeurs caractéristiques des distributions de tous les paramètres (FIG. 8, 9, 10) avant et après correction par h(0) ou g(0) suivant la direction.*

privilégiée perpendiculairement à laquelle il existe une forte variance.

Nous avons envisagé dans un premier temps de ne conserver pour chaque pixel que la réponse la plus faible sur les huit calculées. Cependant, certains des pixels situés à la lisière des villes étaient mal classés. En effet, la fenêtre centrée sur un de ces pixels et utilisée pour estimer les variances conditionnelles contient un mélange de deux textures (urbain et non urbain). Nous avons observé dans ces cas que la valeur minimale des variances conditionnelles est faible et conduit à la classification de ces pixels en zone non urbaine. Une grande partie de cette ambiguïté a été levée en considérant comme paramètre de texture la moyenne des deux réponses centrales, i.e. pour chaque pixel on classe les huit réponses dans l'ordre croissant et on ne conserve que la moyenne des deux valeurs centrales.
En effet, pour la plupart des points de la ville répondant fortement dans toutes les directions cette valeur reste élevée. Pour les champs, répondant faiblement dans toutes les directions, cette valeur reste faible. Quant aux points ambigus, on remarque que la plupart se retrouvent bien classés. En effet, les pixels ville qui répondent très faiblement dans une, deux ou même trois directions mais fortement dans les autres seront alors bien classés, i.e. dans la classe zone urbaine. La valeur du paramètre associée à ces pixels est forte comme en plein centre ville.

2.5 Conclusion

Cette approche nous parait intéressante dans la mesure où elle permet de faire la distinction entre la zone urbaine qui répond de manière forte dans quasiment toutes les directions et deux autres types d'objets que sont les champs et les serres ou hangars isolés. En effet, ces types d'objets répondent de manière positive dans une, deux ou trois directions privilégiées. Cette approche permet donc de tenir compte du caractère anisotrope des objets tout en manipulant un modèle 1D très simple. De plus, à la suite des considérations heuristiques exposées dans le chapitre précédent, nous ne conservons qu'une image de paramètre pour la suite du traitement et donc une quantité d'information moins importante à traiter.

3 Classification

Dans l'étape précédente nous avons transformé l'information niveaux de gris en une information caractéristique de la texture urbaine. Dans cette seconde étape, nous utilisons cette information pour classifier la ville. Pour cela nous utilisons une version modifiée de l'algorithme du *fuzzy Cmeans* [18] qui prend en compte un terme d'entropie.

3.1 Introduction

A ce stade de la chaine de traitement, nous ne prenons pas de décision "dure" (fonction caractéristique qui vaut 0 ou 1 traduisant qu'un objet appartient ou non à l'ensemble). Chaque ensemble est caractérisé par une fonction continue définie sur [0,1] (chaque objet appartient alors à chaque ensemble avec un certain degré).
Une partition de l'image est alors caractérisée par une matrice $U = [u_{ij}]$, appelée matrice de partition, où u_{ij} caractérise le degré d'appartenance du pixel j à la région i [18].
Une méthode de *clustering* visant à trouver des groupements naturels dans l'espace d'état est alors mise en place afin de déterminer cette matrice de partition. L'avantage de cette méthode est qu'elle nous permet de garder la majeure partie de l'information issue des données initiales pour la seconde partie de notre travail. En effet, à ce niveau de traitement, aucune décision définitive n'est prise quant à l'appartenance d'un pixel à telle ou telle classe.

3.2 Définition du critère classique

L'étape de classification consiste à minimiser le critère suivant qui est une généralisation du critère classique des k-moyennes (somme des distances intra-classe) :

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d^2(xj, ci) \tag{9}$$

sous la contrainte suivante :

$$\sum_{i=1}^{C} u_{ij} = 1 \qquad \forall j \tag{10}$$

m caractérise le degré de flou (quand m croît on introduit davantage de flou).
C représente le nombre de classes.
N le nombre de pixels.
ci est le vecteur caractéristique de la classe i.
$d^2(xj, ci)$ est la distance euclidienne (il est possible d'utiliser une autre distance [18]) entre le point j et le centroïde de la classe i.

La minimisation de J se déroule en deux étapes. Dans la première étape nous minimisons par rapport aux ci puis, les ci étant fixés, nous minimisons par rapport aux u_{ij}.
L'algorithme est représenté figure 11 (on suppose que C et m sont connus).
Les éléments de la matrice de partition sont mis à jour de la manière suivante :

$$\begin{cases} u_{ij} = \dfrac{1}{\displaystyle\sum_{k=1}^{C} \left(\frac{d^2(xj, ci)}{d^2(xj, ck)} \right)^{\frac{1}{m-1}}} & si \quad I_j = \emptyset \\ \\ \begin{cases} u_{ij} & = 0 \qquad i \notin I_j \\ \sum_{i \in I_j} u_{ij} & = 1 \qquad i \in I_j \end{cases} & si \quad I_j \neq \emptyset \end{cases} \tag{11}$$
$$avec \quad I_j = \{i | 1 \leq i \leq C, d^2(xj, ci) = 0\}$$

3.3 Définition d'un nouveau critère

Il nous a paru intéressant d'avoir un algorithme dans lequel nous n'aurions pas à fixer le nombre de classes mais qui déterminerait automatiquement un nombre de classes optimal. Dans cette optique, nous nous sommes inspirés des travaux réalisés par deux équipes.
Les premiers ont été effectués par H. Frigui et R. Krishnapuram [9]. La fonction

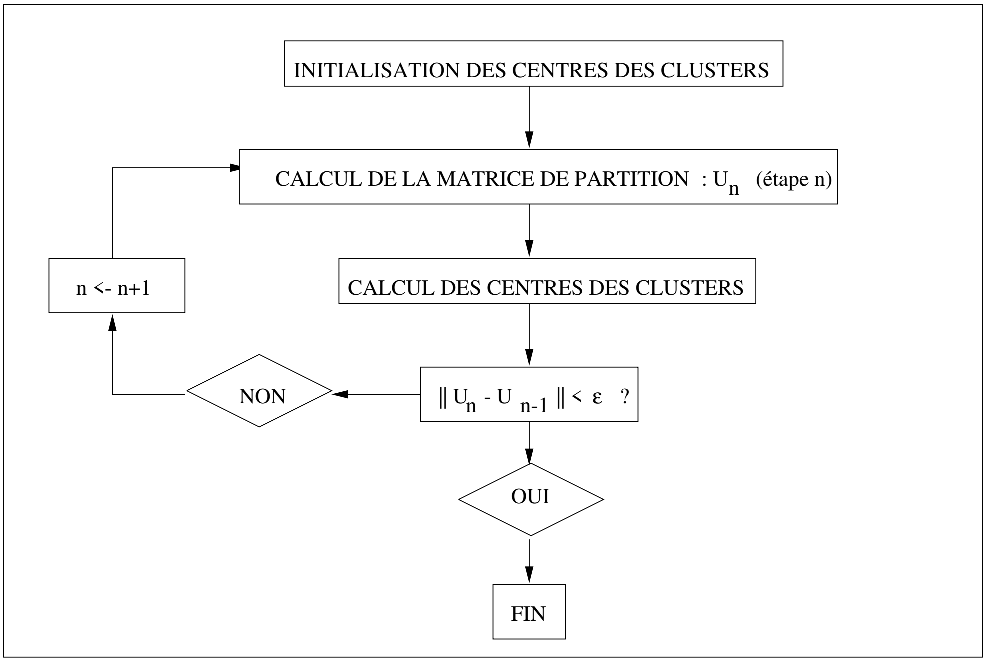


FIG. 11 – *Algorithme de* fuzzy Cmeans

caractéristique à minimiser comporte 2 termes. Le premier est un terme classique caractérisant la distance intraclasse (généralisée au cas *fuzzy*) et correspond à l'équation 9. Le second terme vise à maximiser le nombre de "bons" points dans chaque classe. Il s'agit d'une somme pondérée des degrés d'appartenance u_{ij}.
Une autre équipe [22] a également travaillé sur le sujet mais en restant dans le cas non *fuzzy*. Leur but était de minimiser l'entropie de l'histogramme de l'image classifiée. Le second terme du critère est donc le suivant :

$$J2 = -\alpha \sum_{i=1}^{C} p_i \log(p_i) \tag{12}$$

où p_i est la probabilité a priori de la classe i.

Dans notre cas nous avons étendu les travaux précédents au cas *fuzzy*. Le critère que nous minimisons est le suivant :

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d^2(xj, ci) - \alpha \sum_{i=1}^{C} p_i \log(p_i) \tag{13}$$

sous la contrainte :

$$\sum_{i=1}^{C} u_{ij} = 1 \qquad \forall j \in [0, N]$$

Dans toute la suite on prend $m = 2$.
En considérant que u_{ij} représente la probabilité pour le pixel i d'appartenir à la classe j, nous définissons la probabilité de la classe i comme étant :

$$p_i = \frac{1}{N} \sum_{j=1}^{N} u_{ij}$$

La mise à jour des paramètres est obtenue en minimisant le critère alternativement par rapport aux centroïdes c_i puis par rapport aux u_{ij} (Annexe B). A chaque itération nous ne conservons que les classes dont la probabilité est supérieure à un certain seuil. Dans le cas non *fuzzy* [22], seules les classes de probabilité non nulle sont conservées.

$$ci = \frac{\sum_{j=1}^{N} u_{ij}^2 xj}{\sum_{j=1}^{N} u_{ij}^2} \tag{14}$$

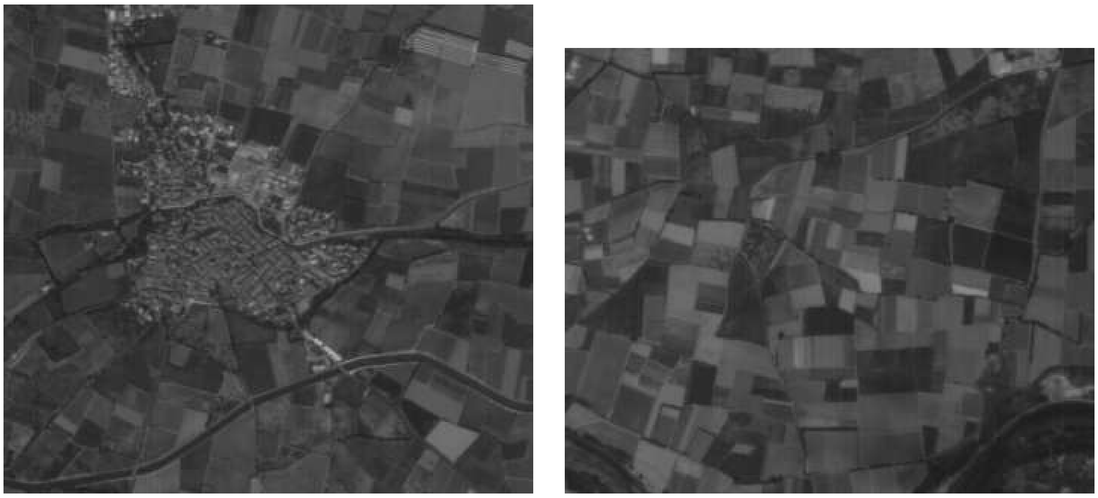
$$u_{ij} = \frac{\frac{1}{d^2(xj, ci)}}{\sum_{k=1}^{C} \frac{1}{d^2(xj, ck)}} + \frac{\alpha}{2N d^2(xj, ci)} \left(1 + \log(p_i) - \frac{\sum_{k=1}^{C} \frac{1 + \log(p_k)}{d^2(xj, ck)}}{\sum_{k=1}^{C} \frac{1}{d^2(xj, ck)}} \right) \tag{15}$$

Dans [9], H. Frigui et R. Krishnapuram font décroitre le paramètre α. De cette manière, leur algorithme trouve une partition en un nombre de classes qui est proche du nombre de classes optimum dès les premières itérations. Nous choisissons la valeur initiale de α de telle manière que les deux termes du critère J (equation 13) soient de même poids puis nous faisons décroître α de la manière suivante :

$$\alpha(k) = \alpha(0) \exp\Big(- \frac{k}{\tau} \Big) \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^2 d^2(xj, ci)}{\sum_{i=1}^{C} p_i \log(p_i)} \tag{16}$$

De cette manière, une fois le nombre de classes optimum atteint, la classification n'est pas biaisée par le terme d'entropie . Dans les tests qui suivent le paramètre τ est fixé à 30 et $\alpha(0)$ à 2. k est le nombre d'itérations.

3.4 Résultats et conclusion



a b

FIG. 12 – *Image initiales: (a) ville et (b) champs*

L'image de données est l'image du paramètre de texture défini dans le chapitre 2.3.

L'algorithme modifié est initialisé par le résultat obtenu à la suite d'un algorithme de *fuzzy Cmeans* classique (5 itérations) et pour un nombre de classes fixé au nombre de classes maximum. Les images testées sont: une image contenant une zone urbaine (12 (a)) et une image de champs (12 (b)) ne contenant aucune ville. Nous avons donc extrait une image de paramètre pour chacune d'elles que nous segmentons avec l'algorithme du *fuzzy Cmeans* modifié. Nous obtenons bien deux classes lorsqu'une ville est présente (image 13 (e)) et une seule dans le cas contraire (image 13 (f)). Nous avons effectué plusieurs tests en considérant un nombre initial de classes variant de 2 à 30. Les résultats sont identiques. Nous avons également appliqué l'algorithme classique du *fuzzy Cmeans*. Nous obtenons les images 13 (c) et (d) avec 2 classes.
Ces résultats montrent l'intérêt de l'algorithme modifié qui ne nécessite pas la connaissance a priori du nombre de classes. En conséquence, lorsqu'il traite l'image de paramètre d'une zone non urbaine il ne détecte qu'une classe. Au contraire, l'algorithme classique détecte plusieurs classes même dans le cas où il n'y en a qu'une.

4 Régularisation par champs de Markov

Nous avons obtenu à l'étape précédente un premier masque urbain qui est le résultat de la classification par l'algorithme du *fuzzy Cmeans* modifié. Nous utilisons la modélisation markovienne pour régulariser notre image en introduisant de l'information contextuelle.

4.1 Introduction

Le critère du MPM (Maximum Posterior Marginal) [21] minimise le nombre de pixels mal classés. Il est donc mieux adapté à des applications de classification-segmentation que le critère du MAP (Maximum A Posteriori) [5][11].

4.1.1 Critère du MPM

Après avoir défini les champs de Markov et les distributions de Gibbs (paragraphe 2.2), nous considérons le contexte particulier de la segmentation d'image.

Nous notons X le champ aléatoire relatif à l'image observée à valeurs dans Ω^N.
Nous notons L le champ aléatoire relatif à l'image labelisée à valeurs dans Λ^N.
X étant connu, nous cherchons la configuration L qui maximise l'expression suivante :

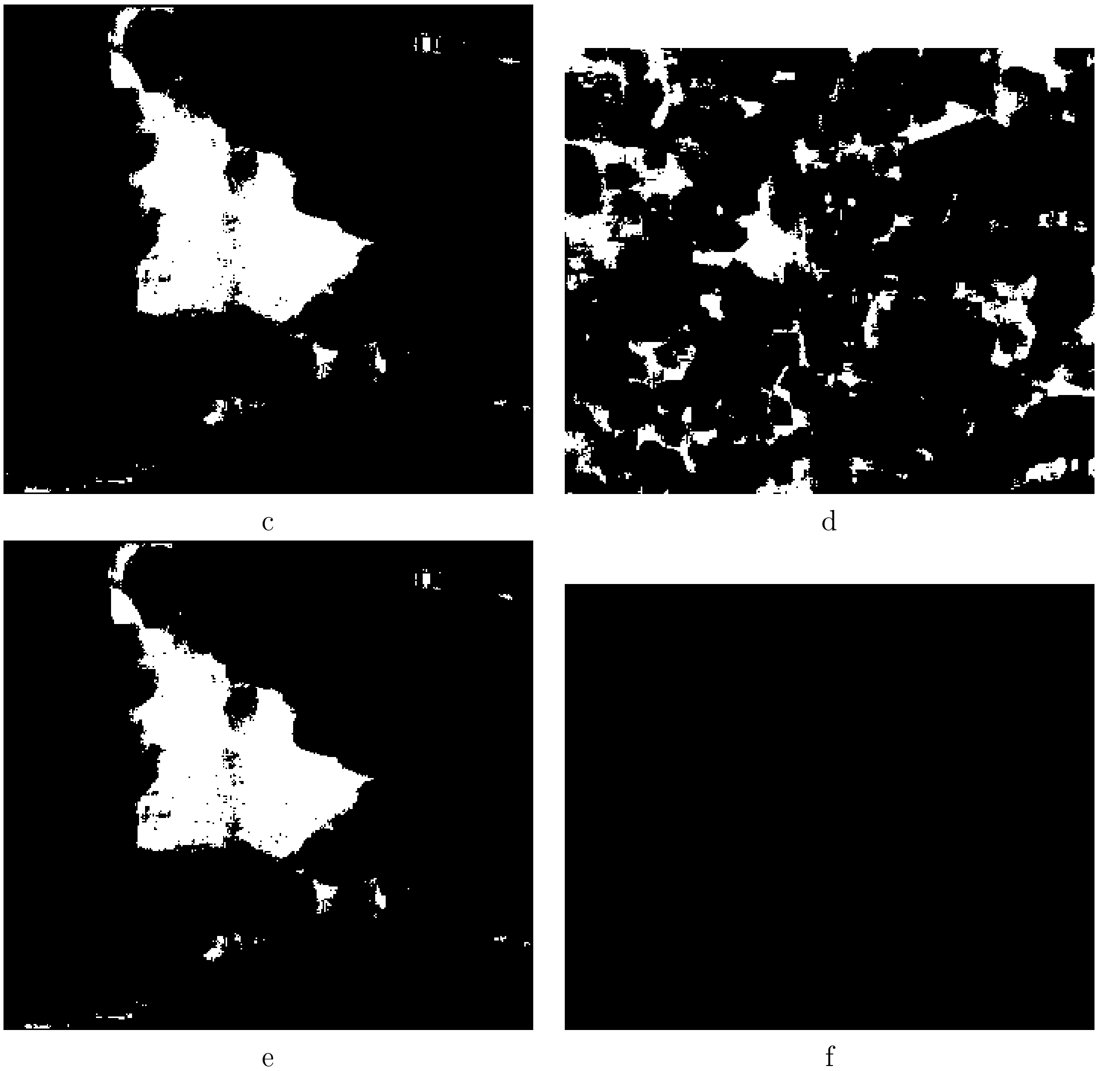


FIG. 13 – *Images segmentées par* Fuzzy Cmeans *classique (*$C_{initial} = 2$*): (c) et (d). Images segmentées par* Fuzzy Cmeans *modifié: (e) et (f).*

$$P(L|X) = \frac{P(X|L)P(L)}{P(X)}$$

ce qui revient, X étant connu et donc $P(X)$ constant, à maximiser l'expression suivante :

$$P(X|L)P(L)$$

Les données sont modélisées (modèle de texture [3], modèle de bruit [10]...) ce qui nous permet de définir la loi $P(X|L)$ sur Ω^N. Dans le chapitre suivant nous considérons deux lois différentes. Nous modélisons le champ L par un champ de Markov ce qui nous permet de définir la loi a priori $P(L)$ sur Λ^N. Le modèle a priori que nous considérons dans le chapitre suivant est le modèle de Potts.

L'approche du MPM consiste à maximiser les probabilités marginales a posteriori. Ainsi, le label $\hat{l}_s$ retenu pour le site s est celui qui maximise la probabilité suivante : $P(L_s = \hat{l}_s | X = x) \geqq P(L_s = l_s | X = x), \quad \forall l_s \in \Lambda$

L'estimateur du MPM est obtenu par l'échantillonneur de Gibbs [11]. L'algorithme consiste à générer une chaine de Markov sur Λ^N. Une fois la convergence atteinte, les probabilités marginales a posteriori en s sont estimées en comptant le nombre de fois où chaque valeur de l'ensemble Λ apparait en s pour une série de configurations. Le label $\hat{l}_s$ prend la valeur qui apparait le plus grand nombre de fois en s.

4.2 Attache aux données

Nous comparons 2 termes d'attache aux données.

4.2.1 Première approche

Dans cette approche nous considérons la fonction d'appartenance de la classification *fuzzy* comme représentant la vraisemblance. Alors la loi d'attache aux données considérée est la suivante :

$$P(X|L) = \prod_{i \in S} u_{il_i} \tag{17}$$

u_{ij} est le degré d'appartenance du pixel i à la classe j. Il est obtenu par l'algorithme du *fuzzy Cmeans.*
Ce terme d'attache aux données est intéressant dans la mesure où il ne fait intervenir aucun paramètre que nous serions obligé d'estimer. Ceci est notamment le cas lorsque l'on fait l'hypothèse d'un modèle gaussien. En effet, dans ce dernier cas nous devons en sus des paramètres liés au modèle a priori estimer deux paramètres (moyenne et variance) par classe.

4.2.2 Deuxième approche

Nous avons voulu comparer les résultats obtenus précédemment avec ceux obtenus en modifiant le terme d'attache aux données. Dans ce cas nous considérons une attache aux données gaussienne dont les paramètres (moyenne et variance des classes) sont estimés à partir de la classification déjà relativement précise (mais qui ne tient compte d'aucune information contextuelle), résultat de l'algorithme du *fuzzy Cmeans.* La loi d'attache aux données considérée est la suivante :

$$P(X|L) = \prod_{i \in S} \frac{1}{\sqrt{2\pi}\sigma_{l_i}} \exp\left(\frac{(x_i - \mu_{l_i})^2}{2\sigma_{l_i}^2}\right) \tag{18}$$

σ_j^2 et μ_j sont respectivement la variance et la moyenne de la classe j.

4.3 Modèle a priori

Le modèle a priori considéré est le modèle de Potts [28]. La probabilité a priori est la suivante :

$$P(L) = \frac{1}{Z_\beta} \exp\left(-\beta \sum_{cl=\{l_i,l_j\}, cl \in CL} \delta(l_i, l_j)\right) \tag{19}$$

$\delta(l_i, l_j) = 1$, si $l_i = l_j$
$\delta(l_i, l_j) = 0$, si $l_i \neq l_j$

5 Résultats

Des tests ont été effectués sur des images de simulation SPOT5, fournies par le CNES. Elles ont été obtenues à partir d'images aériennes push-broom. La résolution des images utilisées est de 5 mètres. Il s'agit du canal panchromatique.

L'algorithme global utilisé est presenté sur la figure 14. Les résultats obtenus sont satisfaisants. Ainsi, sur l'image 15(a) la serre est en zone non urbaine. En ce qui concerne les deux types d'attache aux données, l'avantage va au second. En effet, les contours de la ville sont plus précis (15(b) et 17 (b)), que sur (15(a) et 17(a)). Pour les tests réalisés sur l'image d'Eppeville nous avons appliqué uniquement l'attache aux données 2.
Nous avons donc pensé utiliser une image de marqueurs afin d'éliminer une grande partie de ces fausses alarmes. En effet, comme nous l'avons expliqué dans le paragraphe 2.4, nous avions pensé dans un premier temps ne conserver pour chaque pixel que la réponse minimale. Or, cela conduisait à une mauvaise segmentation et notamment les limites de la ville se trouvaient largement à l'intérieur de celle-ci. Nous avons donc utilisé l'image résultant de la segmentation de l'image correspondant à la valeur minimale de la variance conditionnelle. Nous avons appliqué un filtre médian à cette image segmentée. Nous utilisons donc cette image pour marquer la présence d'une zones urbaine (FIG. 18 (a) et (b)). Les zones détectées par la segmentation ayant une intersection nulle avec le résultat du filtre médian sont interprétées comme fausses alarmes. Cette méthode nous permet d'éliminer la plupart des fausses alarmes et d'obtenir une bonne segmentation des zones urbaines (FIG. 19 (b et d) et FIG. 20 (b et d) et FIG. 21 (b)). Le terme d'attache aux données 2 donne les meilleurs résultats, i.e. des contours précis et peu de fausses alarmes.

6 Conclusion et perspectives

Nous avons donc proposé une méthode d'extraction des zones urbaines fondée sur une analyse de la texture de l'image. Pour ce faire, nous utilisons une approche qui tient compte de la direction. Ceci est très utile pour notre application car très souvent les petits parcellaires, les hangars ou encore les serres sont confondus avec la ville. Or c'est justement une orientation privilégiée qui les différencie.

L'autre intérêt de la modélisation considérée est que nous manipulons des modèles 1D qui sont très simples et demandent peu de temps de calcul. Notons que dans [4] X. Descombes avait introduit un paramètre d'ordre afin d'éliminer les fausses alarmes mais que l'estimation de celui-ci se traduisait par un accroissement important de la charge de calcul.
Enfin, Nous avons également apporté une modification à l'algorithme du *fuzzy Cmeans*

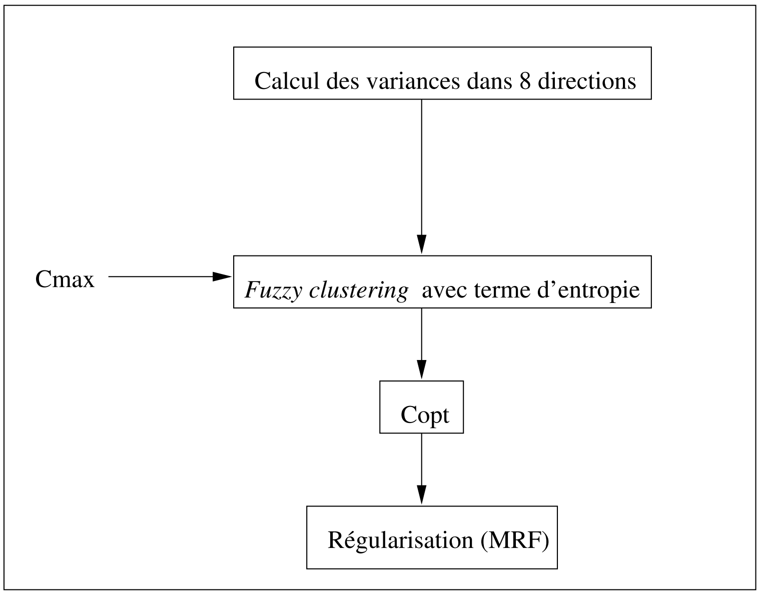


FIG. 14 – *Algorithme global*

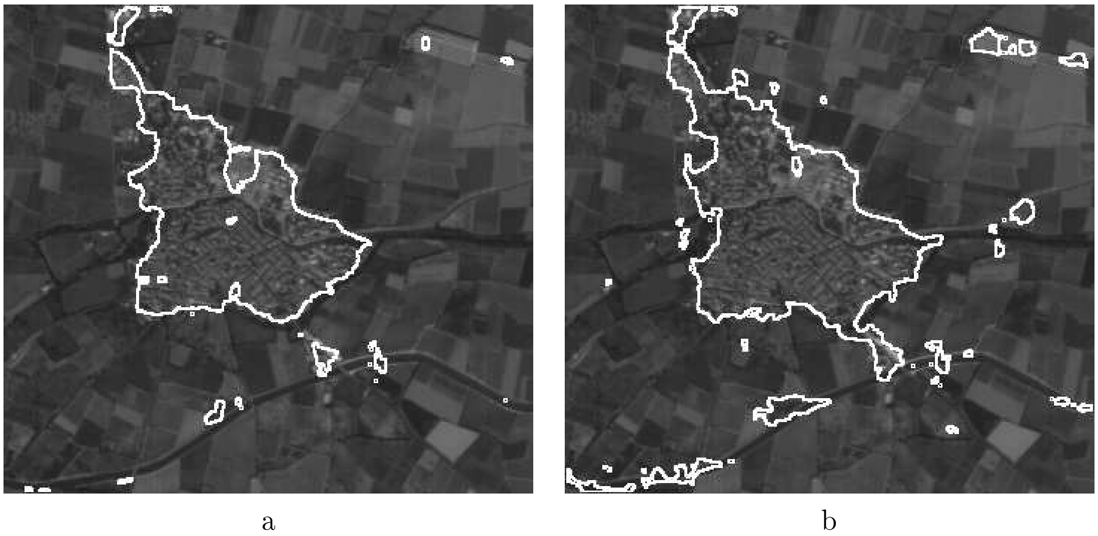
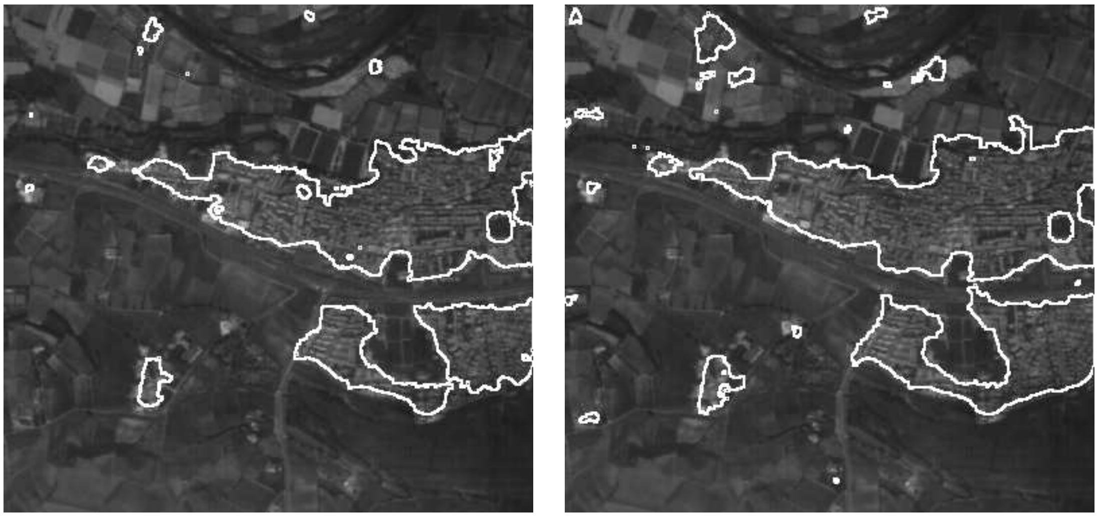


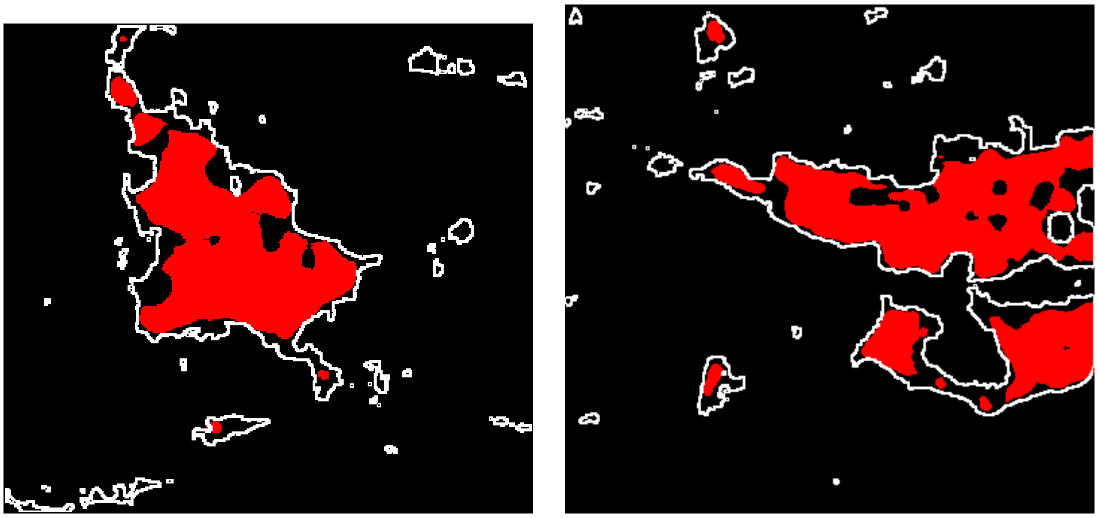
FIG. 15 – *Images segmentées avec l'attache aux données 1 ($\beta = 0.3$) a) puis 2 ($\beta = 0.5$) b). Il s'agit d'une zone proche de Perpignan: le village de Toreilles*

FIG. 16 – *Image initiale: village de Canet en Roussillon*



a b

FIG. 17 – *Images segmentées avec l'attache aux données 1 ($\beta = 0.3$) a) puis 2 ($\beta = 0.5$) b). Il s'agit d'une zone proche de Perpignan: le village de Canet en Roussillon*



a b

FIG. 18 – *Superposition du résultat du filtre médian en gris et du résultat de la segmentation (attache aux données 2) en blanc pour les villages de Toreilles a) et de Canet en Roussillon b)*

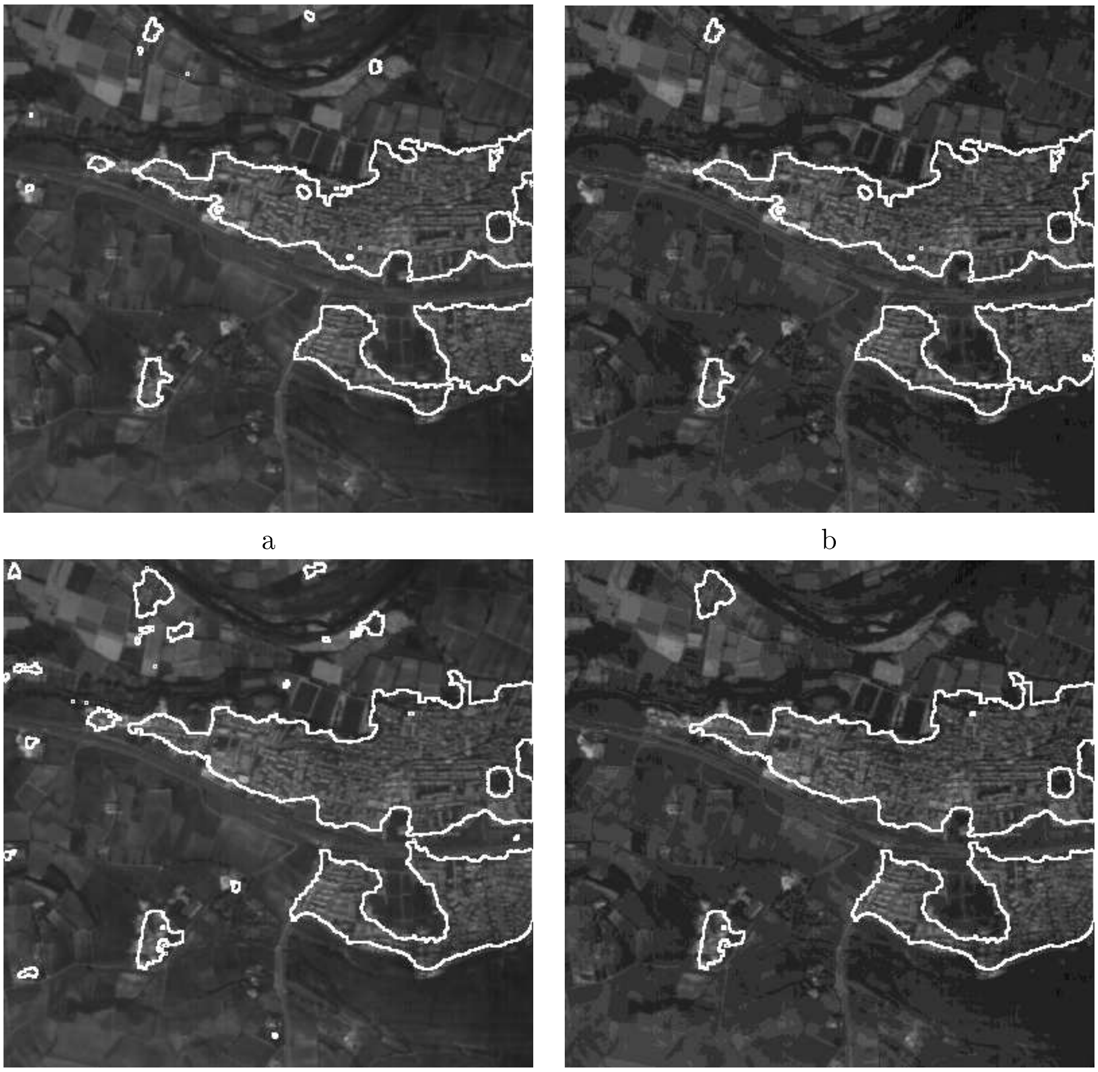


FIG. 19 – *Attache aux données 1 (a): Segmentation avant élimination des fausses alarmes et (b): Segmentation après élimination des fausses alarmes /Attache aux données 2 (c): Segmentation avant élimination des fausses alarmes et (d): Segmentation après élimination des fausses alarmes*

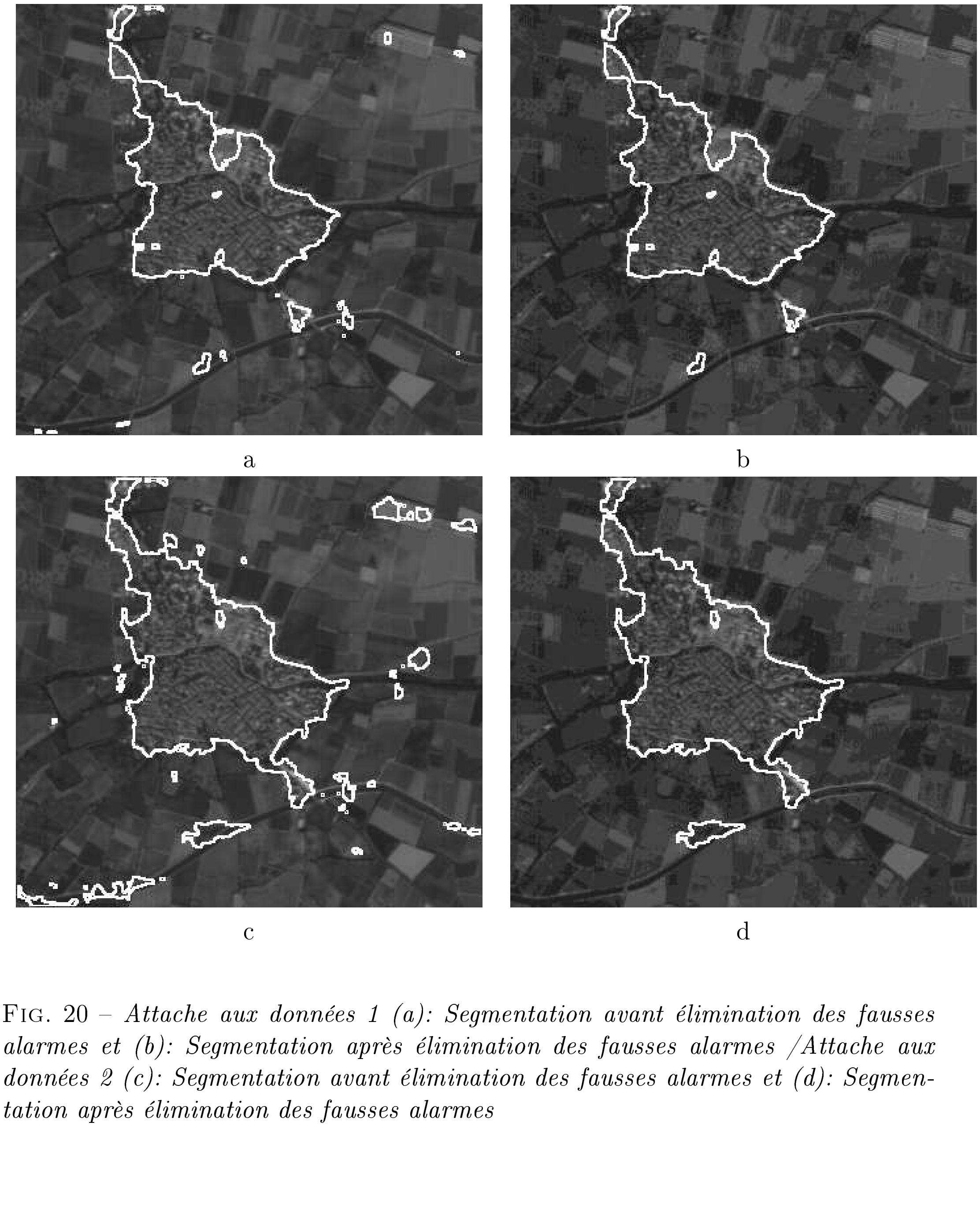
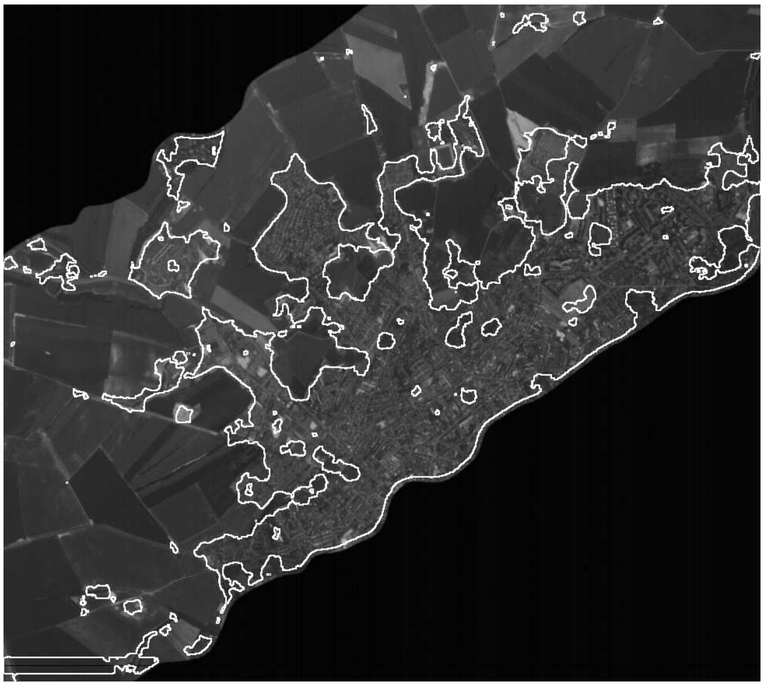
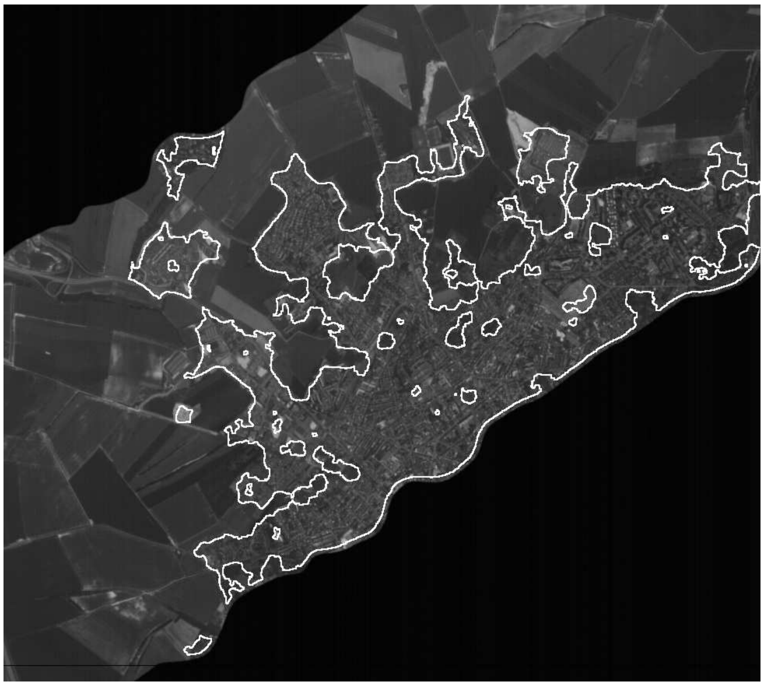


FIG. 20 – *Attache aux données 1 (a): Segmentation avant élimination des fausses alarmes et (b): Segmentation après élimination des fausses alarmes /Attache aux données 2 (c): Segmentation avant élimination des fausses alarmes et (d): Segmentation après élimination des fausses alarmes*



a



b

FIG. 21 – *Résultats pour Eppeville. Attache aux données 2 (a): Segmentation avant élimination des fausses alarmes et (b): Segmentation après élimination des fausses alarmes*

classique ce qui permet de ne pas avoir à fixer un nombre de classes précis et ce qui permet surtout de ne pas détecter plusieurs classes lorsqu'il n'y en a qu'une.

Comme nous l'avons signalé dans l'introduction la zone urbaine peut être analysée de deux manières. Dans les travaux présentés dans ce rapport, la ville est considérée comme un seul et même objet que nous avons extrait. Dans la suite de nos travaux nous allons chercher à distinguer dans la ville plusieurs régions suivant des critères qui restent à définir.

Annexe A

Le but est ici de calculer les paramètres du réseau décimé en fonction des paramètres du réseau complet (FIG. 5).
Pour cela, il convient d'intégrer la loi jointe du modèle par rapport aux paramètres que l'on souhaite voir disparaître.
Les calculs sont effectués ici en décimant un site sur deux. Ceci est un cas particulier. En effet, pour traiter le problème présenté dans le chapitre 2.3 il convient de décimer davantage le réseau et d'intégrer sur 11, 16 et 27 variables.
Dans ce qui suit $E[.]$ est la fonction partie entière.
La loi jointe est la suivante :

$$P(x_1, x_2, \ldots, x_N) \quad \alpha \quad \exp\left(-\beta\left(\sum_{k=1}^{N}\lambda(x_k-\mu)^2 + \sum_{k=1}^{N-1}(x_k - x_{k+1})^2\right)\right)$$

Nous supposons par ailleurs des conditions aux bords toroïdales. Dès lors, x_1 a deux voisins x_0 et x_2, avec $x_0 = x_N$. De même, x_N a deux voisins x_{N-1} et x_{N+1}, avec $x_{N+1} = x_0$.

$$P(x_2, x_4, \ldots, x_{2E[\frac{N}{2}]}) \quad \alpha \quad \int_{x_1, x_3, \cdots, x_{2E[\frac{N-1}{2}]+1}} \exp\left(-\beta\left(\sum_{k=1}^{N}\lambda(x_k-\mu)^2 + \sum_{k=1}^{N-1}(x_k - x_{k+1})^2\right)\right) dx_1 \cdots$$

$$P(x_2, x_4, \ldots, x_{2E[\frac{N}{2}]}) \quad \alpha \quad \exp\left(-\beta\sum_{k=1}^{E[\frac{N}{2}]}\lambda(x_{2k}-\mu)^2\right) * I$$

Avec :

$$I = \prod_{k=0}^{E[\frac{N-1}{2}]}\left(\int_{x_{2k+1}} \exp\left(-\beta(\lambda(x_{2k+1}-\mu)^2 + (x_{2k+1}-x_{2k})^2 + (x_{2k+1}-x_{2k+2})^2)\right) dx_{2k+1}\right)$$

On pose :

$$I_{x_{2k+1}} = \int_{x_{2k+1}} \exp\left(-\beta\big(\lambda(x_{2k+1}-\mu)^2 + (x_{2k+1}-x_{2k})^2 + (x_{2k+1}-x_{2k+2})^2\big)\right) \quad dx_{2k+1}$$

Le calcul de $I_{x_{2k+1}}$ nous donne :

$$I_{x_{2k+1}} = \sqrt{\frac{\pi}{\beta(2+\lambda)}} \exp\left(-\frac{\beta}{2+\lambda}\left((x_{2k+2}-x_{2k})^2 + \lambda(x_{2k+2}-\mu)^2 + \lambda(x_{2k}-\mu)^2\right)\right)$$

Nous rappelons que x_0 et x_{N+1} sont définis par les conditions aux bords précisées ci-dessus.

On en déduit donc que :

$$P(x_2, x_4, \ldots)\ \alpha\ \exp\left(-\frac{\beta}{2+\lambda}\left(\sum_{k=1}^{E[\frac{N-2}{2}]}(x_{2k}-x_{2k+2})^2 + \sum_{k=1}^{E[\frac{N}{2}]}(\lambda(2+\lambda)+2\lambda)(x_{2k}-\mu)^2\right)\right)$$

$$P(x_2, x_4, \ldots)\ \alpha\ \exp\left(-\beta_1\left(\sum_{k=1}^{E[\frac{N-2}{2}]}(x_{2k}-x_{2k+2})^2 + \sum_{k=1}^{E[\frac{N}{2}]}\lambda_1(x_{2k}-\mu)^2\right)\right)$$

On obtient donc les 2 relations suivantes :

$$\beta_1 = \frac{\beta}{2+\lambda}$$
$$\lambda_1 = \lambda(4+\lambda)$$

Un calcul analogue permet d'obtenir les résultats en décimant davantage le réseau et en intégrant sur 11, 16 et 27 variables . Ces calculs ont été effectués avec le logiciel Maple.

Annexe B

La mise à jour des paramètres de l'algorithme de *fuzzy Cmeans* s'effectue en minimisant le critère suivant :

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^2 d^2(xj, ci) - \alpha \sum_{i=1}^{C} p_i \log(p_i) \tag{20}$$

sous la contrainte :

$$\sum_{i=1}^{C} u_{ij} = 1 \qquad \forall j \in [0, N]$$

La minimisation se fait alternativement par rapport aux ci puis par rapport aux u_{ij}. La prise en compte de la première contrainte s'effectue en utilisant les multiplicateurs de Lagrange. Nous minimisons alors le critère défini comme suit :

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^2 d^2(xj, ci) - \alpha \sum_{i=1}^{C} p_i \log(p_i) - \sum_{j=1}^{N} \lambda_j (\sum_{i=1}^{C} u_{ij} - 1) \tag{21}$$

On résoud alors les deux équations suivantes :
1) On minimise (21) par rapport aux centroïdes c_i :

$$\frac{\partial J}{\partial ci} = \sum_{j=1}^{N} (u_{ij})^2 (-2(xj - ci)) = 0$$

2) Les centroïdes étant connus on minimise par rapport aux u_{ij} :

$$\frac{\partial J}{\partial u_{ij}} = 2u_{ij} d^2(xj, ci) - \alpha(1 + \log(p_i)) \frac{\partial p_i}{\partial u_{ij}} - \lambda_j = 0$$

$$\frac{\partial p_k}{\partial u_{ij}} = \frac{\delta(k, i)}{N} \quad \text{où} \quad \delta \quad \text{est le symbole de kronecker}$$

$$ci = \frac{\sum_{j=1}^{N} u_{ij}^2 xj}{\sum_{j=1}^{N} u_{ij}^2} \quad et$$

$$u_{ij} = \frac{\dfrac{1}{d^2(xj,ci)}}{\displaystyle\sum_{k=1}^{C} \frac{1}{d^2(xj,ck)}} + \frac{\alpha}{2Nd^2(xj,ci)} \left(1 + \log(p_i) - \frac{\displaystyle\sum_{k=1}^{C} \frac{1+\log(p_k)}{d^2(xj,ck)}}{\displaystyle\sum_{k=1}^{C} \frac{1}{d^2(xj,ck)}} \right)$$

Références

[1] Baraldi A. et Parmiggiani F. – Urban area classification by multispectral spot images. *IEEE Trans. on geoscience and remote sensing*, vol. 28, n4, juillet 1990, pp. 674–680.

[2] Besag J. E. – Spatial interaction and the statistical analysis of lattice system. *Journal of the Royal Statistical Society, Series B*, no36, 1974, pp. 192–236.

[3] Cross G. R. et Jain A. K. – Markov random field texture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, n1, janvier 1983, pp. 25–39.

[4] Descombes X. – *Champs markovien en analyse d'images.* – Thèse de Doctorat, ENST, 1993.

[5] Descombes X., Morris R. et Zerubia J. – Quelques améliorations à la segmentation d'images bayésienne. première partie: modélisation. *Traitement du signal*, vol. 14, n4, 1997, pp. 373–382.

[6] Descombes X. et Préteux F. – Topology and parameter estimation in mrf modeling. *In : SPIE, Neural and Stochastic Methods in Image and Signal Processing II.* – San Diego, juillet 1993.

[7] Descombes X., Sigelle M. et Préteux F. – Estimating gaussian markov random field parametres in a non-stationary framework: Application to remote sensing imaging. *IEEE Trans. on Image Processing*, à paraitre en 1998.

[8] Dubes R. C. et Jain A. K. – Random fields models in image analysis. *Journal of applied satistics*, vol. 16, n2, 1989, pp. 131–164.

[9] Frigui H. et Krishnapuram R. – A robust clustering algorithm based on competitive agglomeration and soft rejection of outliers. *In : CVPR*, pp. 550–555. – San Francisco, juin 1996.

[10] Geman D. et Reynolds G. – Constrained restoration and the recovery of discontinuities. *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 14, n 3, mars 1992, pp. 367–383.

[11] Geman S. et Geman D. – Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, n6, novembre 1984, pp. 721–741.

[12] Gidas B. – A renormalization group approach to image processing problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, n2, février 1989, pp. 164–180.

[13] Gouinaud C. – *Traitement d'images satellitaires pour la détection d'agglomérations.* – Thèse de Doctorat, ENST, 1996.

[14] Haralick R. M. – Statistical and structural approaches to texture. *proc. of the IEEE*, vol. 67, n5, mai 1979, pp. 786–804.

[15] Herland E. A. – Sar interferometry with ers-1 in forested areas. *In: IGARSS*, pp. 202–204. – Italie, 1995.

[16] Herland E. A. et Berglund R. – Demonstration of operational sea ice monitoring in the baltic sea with ers-1 sar. *Int. Journal of remote sensing*, vol. 16, n 17, novembre 1995.

[17] Houzelle S. et Giraudon G. – Data fusion using spot and sar images for bridge and urban area extraction. *In: IGARSS.* – Helsinki, juin 1991.

[18] Krishnapuram R. et Keller J. M. – *Fuzzy and possibilistic clustering methods for computer vision.* – 1994, *SPIE Institute series*, vol. 12. pp. 133-159.

[19] LeMen H. – Cartographie semi-automatique de l'évolution de l'occupation des sols par télédétection. – Habilitation à diriger des recherches, Université René Descartes, décembre 1993.

[20] Li S. Z. – *Markov Random Field Modeling in Computer Vision.* – Springer-Verlag, 1995.

[21] Marroquin J., Miter S. et Poggio T. – Probabilistic solution of ill-posed problems in computational vision. *Journal of the american statistical association*, no82, 1987, pp. 76–89.

[22] Palubinskas G., Descombes X. et Kruggel F. – An unsupervised clustering method using the entropy minimization. *In: ICPR.* – Australie, août 1998.

[23] Paquerault S. – Traitement d'images aériennes: reconnaissance de zones urbaines sur des critères de texture. – Rapport de DEA, ENST, 1994.

[24] Pearson Carl E. (édité par). – *Handbook of Applied Mathematics.* – Van Nostrand Reinhold, 1990, second édition.

[25] Richard F., Falzon F., Zerubia J. et Giraudon G. – Segmentation of spot images using markov random fields. *In: EUSIPCO.* – Rhodes, septembre 1998.

[26] Schroeder M. – *Fractals, chaos, power laws : minutes from an infinite paradise.* – W. H. Freeman and Company, 1990.

[27] Serendero M. – *Extraction d'informations symboliques en imagerie SPOT: réseaux de communication et agglomérations.* – Thèse de Doctorat, UNSA, décembre 1989.

[28] Sigelle M. et Ronfard R. – Modèles de potts et relaxation d'images de labels par champs de markov. *Traitement du Signal*, vol. 6, n9, 1992, pp. 449–458.

[29] Treitz P. M., Filho O. R., Howarth P. J. et Soulis E. D. – Textural processing of multi-polarization sar for agricultural crop classification. *In: IGARSS*, pp. 1986–1988. – Lincoln (USA), 1996.

[30] Weber C. – *Images satellitaires et milieu urbain.* – Collection géomatique, 1995, hermès édition.

[31] Weigl K. et Yu S. – A hybrid approach for image classification using neural network and markov random field. *In: Int. Conf. on Artificial Neural Networks*, pp. 321–326. – Paris, octobre 1995.

[32] Winkler G. – *Image analysis, random fields and dynamic Monte Carlo methods: a mathematical introduction.* – Springer-Verlag, 1995.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 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 - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399