



HAL
open science

Classification markovienne automatique d'images aériennes de haute résolution

Xavier Marsault, Matthieu Aubry, Jean Fortin, Lamiae Azizi

► **To cite this version:**

Xavier Marsault, Matthieu Aubry, Jean Fortin, Lamiae Azizi. Classification markovienne automatique d'images aériennes de haute résolution. SAGEO 2009, 2009, PARIS, France. hal-04379773

HAL Id: hal-04379773

<https://hal.science/hal-04379773>

Submitted on 15 Jan 2024

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.

Classification markovienne automatique d'images aériennes de haute résolution

Xavier Marsault* — Matthieu Aubry** — Jean Fortin*** — Lamiae Azizi****

* Laboratoire MAP-ARIA, ENSAL de Lyon
UMR MAP 694 du CNRS
3, rue Maurice Audin, F-69512 Vaulx en Velin cedex
xavier.marsault@aria.archi.fr

** INSA de Lyon
20, avenue Albert Einstein, F-69100 Villeurbanne
matthieu.aubry@gmail.com

*** LACL, Faculté des Sciences et Technologies, Université Paris XII
61, avenue du Général de Gaulle, F-94010 Créteil cedex
jean.fortin@ens-lyon.org

**** Laboratoire MISTIS, INRIA Rhône-Alpes
655, avenue de l'Europe, F-38330 Montbonnot-Saint-Martin
lamiae.azizi@inrialpes.fr

RÉSUMÉ. Dans le cadre du projet de la région Rhône-Alpes DEREVE II, le laboratoire MAP-ARIA de l'ENSAL a démarré en 2005/2006 une collaboration avec le laboratoire MISTIS de l'INRIA sur le thème de la classification détaillée d'orthophotographies numériques couleur (ONC), avec comme terrain d'étude le département de Haute-Savoie, avec lequel nous avons signé une convention de mise à disposition de données géomatiques (RGD73-74). Le présent article a pour but de présenter le contexte scientifique de cette étude et les résultats expérimentaux obtenus, qui s'appuient sur les logiciels SpaCEM³ (Spatial Clustering with EM and Markov Models) et IFE (Image Feature Extraction) développés respectivement à l'INRIA et à MAP-ARIA.

ABSTRACT. During the DEREVE II project, MAP-ARIA laboratory started in 2005/2006 a collaboration with INRIA MISTIS laboratory on the detailed classification of digital color orthophotos (ONC). The field study was the french department of Haute-Savoie, with whom we signed an agreement for providing GIS data (RGD73-74). This article aims to present the scientific background for this study and the experimental results that we obtained, based on software SpaCEM3 (Spatial Clustering with EM and Markov Models) and IFE (Image Feature Extraction) respectively developed at INRIA and MAP-ARIA.

MOTS-CLÉS : champs de Markov cachés, modèles de Markov triplets, extraction de mesures sur une image de haute résolution, paramètres d'apprentissage, classification automatique

KEYWORDS: hidden Markov fields, triplet Markov models, high resolution image feature extraction, learning parameters, automatic classification

1. Introduction

La reconnaissance automatique de motifs de texture dans une collection d'images est un sujet de recherche récurrent et passionnant, dont les applications sont multiples et variées (biologie, médecine, physique, géographie, urbanisme, géomatique, paysage), et les enjeux plus actuels à l'heure d'un Internet toujours plus orienté vers l'exploitation automatique et systématique de données picturales de toutes sortes. Pour nous, l'objectif est de classer une banque de photographies aériennes en Haute-Savoie, dont la résolution est de 50cm par pixel, en y reconnaissant différentes matières (neige, glace, herbe, forêt, route...).

Dans (Premoze *et al.*, 1999), les auteurs effectuent une classification probabiliste d'un terrain montagneux à partir d'images aériennes en niveau de gris, de résolution inférieure à celles que nous utilisons. Leur idée générale est de classer chaque pixel de l'image à l'aide d'un vecteur de huit propriétés discriminantes: intensité du pixel, intensité moyenne des pixels voisins, minimum et maximum des intensités des pixels voisins, altitude, pente, direction de la pente, angle par rapport à l'horizon. Dans une première phase, ils effectuent un apprentissage en donnant un certain nombre de points pour chaque classe. Cela permet de calculer les lois de probabilités des différentes valeurs suivant la classe. L'image est ensuite segmentée en utilisant un classificateur bayésien. Si cette méthode permettait d'avoir des résultats relativement corrects pour l'époque, les dépendances spatiales des pixels n'étaient décrites que par les propriétés 2 à 4 du vecteur descripteur et ne tenaient compte que du voisinage immédiat.

S'inspirant de ce travail, nous avons aussi testé l'apprentissage avec des paramètres liés à la géométrie du terrain (altitude et pente, estimées à partir du MNT à 16m). Cela semblait être une idée intéressante, puisque la présence ou l'absence de certaines matières peut être liée à ces données variables (on trouve plus facilement de la neige à haute altitude en été, et moins d'arbres, voir plus du tout). Pourtant, elle n'a pas favorisé l'obtention de meilleurs résultats (Fortin *et al.*, 2005). De plus, cette méthode requiert un catalogue d'échantillons à plusieurs altitudes, ce qui complique la création d'une base d'apprentissage, mais aussi l'étape de classification.

Pour classer des textures plus complexes, notamment en couleur, il est nécessaire de s'appuyer sur des voisinages plus larges, et d'améliorer le vecteur descripteur. Nous nous sommes alors tournés vers l'utilisation de modèles markoviens qui tiennent compte naturellement des dépendances spatiales au sein des données, et nous avons entrepris une étude systématique des paramètres d'apprentissage devant composer le vecteur descripteur (paramètres d'observation).

Une description des modèles de champ de Markov, même simplifiée, est toujours difficile à entreprendre, car les pré requis sont souvent peu connus, sauf des spécialistes en probabilités et statistiques. Néanmoins, il apparaît nécessaire de fournir au lecteur un minimum de connaissances (section 2). Nous renvoyons à (Blanchet, 2007) pour de plus amples détails, ou à divers travaux récents qui ont utilisé les champs de Markov en analyse ou en synthèse de signal. En ce qui concerne la classification des textures, on peut citer (Newsam *et al.*, 2002) et les techniques d'estimation de points d'intérêt dans les images (Blanchet *et al.*, 2005) et

(Mouret et al., 2008). Dans notre cas d'étude, nous nous plaçons résolument dans l'estimation dense de paramètres, car c'est en chaque pixel de l'image que l'on désire obtenir une information sur la nature du sol ou du sur-sol.

Durant la dernière décennie, les programmes markoviens d'apprentissage et de classification développés en langage C++ par MISTIS (SEMS, NEM, puis SpaCEM³) n'ont cessé de s'améliorer et de progresser en simplicité d'utilisation, grâce aux talents conjugués de ses chercheurs, doctorants et ingénieurs d'étude. Issus de travaux de recherche en classification non supervisée ou supervisée s'appuyant sur les champs de Markov cachés et triplets, ces programmes modélisent les interactions présentes dans des données uni ou multi-dimensionnelles (section 2), en particulier les dépendances spatiales dans les textures, ce qui nous intéresse directement pour obtenir une bonne reconnaissance / classification des motifs de sol. De notre côté, nous avons créé en langage C le programme IFE (Image Feature Extraction, section 5.1) pour calculer des paramètres d'observation (section 4) et lancer de nombreux tests (section 5) avec des configurations variées de paramètres et d'algorithmes. Pour cela, un générateur automatique de scripts de tests - écrit en PHP - rend totalement transparentes pour l'utilisateur les étapes des calculs.

2. Modèles de champs de Markov cachés et triplets

2.1. Définitions

Soit S un ensemble de points spatialement organisés, indexés dans $\{1, \dots, n\}$. Pour une image ces points seront en général disposés sur une grille régulière de dimension 2. On note $N(i)$ le voisinage du point i . Les voisinages utilisés le plus souvent sont les quatre ou les huit plus proches voisins: ils sont appelés respectivement voisinages d'ordre un et deux.

Définition 1: *un champ aléatoire Z est un champ de Markov si et seulement s'il vérifie la propriété:*

$$\forall z, P(z_i | Z_{S \setminus \{i\}} = z_{S \setminus \{i\}}) = P(z_i | Z_{N(i)} = z_{N(i)})$$

Cette propriété signifie que parmi les autres points, seul le voisinage du point i a une influence sur le champ au point i : il s'agit donc d'une généralisation de la notion de chaîne de Markov.

Définition 2: *un champ aléatoire Z suit une loi de Gibbs si la distribution de probabilité jointe est de la forme:*

$$P(z) = W^{-1} \exp(-H(z))$$

où la fonction énergie H peut se décomposer en somme des potentiels sur les cliques:

$$H(z) = \sum_{c \in C} V_c(z_c)$$

$W = \sum_Z \exp(-H(z))$ est la constante de normalisation.

Le théorème de Hammersley-Clifford énonce que Z est un champ de Markov si et seulement si $P(z)$ suit une loi de Gibbs. Ce théorème permet de décrire plus facilement les champs de Markov, en donnant la fonction énergie associée.

2.2. Champs de Markov discrets

Pour un champ de Markov discret, la variable Z_i représente une étiquette attribuée au point i dans $\{1, \dots, K\}$. Comme on l'a vu précédemment, on peut définir le champ de Markov à partir de sa fonction énergie. Dans le cas où l'on se limite aux cliques d'ordre inférieur à 3, l'énergie d'un champ de Markov discret s'écrit :

$$H(z|\beta) = \sum_{i=1}^n \left[V_i(z_i) - \frac{\beta}{2} \sum_{j \in N(i)} V_{ij}(z_i, z_j) \right]$$

Les potentiels sur les singletons (cliques d'ordre 1) donnent alors l'importance relative des différentes étiquettes, et les potentiels sur les paires (cliques d'ordre 2) donnent la corrélation spatiale entre les différentes étiquettes. Le paramètre β donne l'importance relative du terme spatial.

2.3. Exemple

Le modèle de Potts sans champ externe est un cas très simple de champ de Markov discret. Dans ce modèle, les potentiels sur les singletons sont nuls, et les potentiels sur les paires sont égaux à une unique valeur non nulle lorsque les étiquettes sont identiques, et nuls sinon.

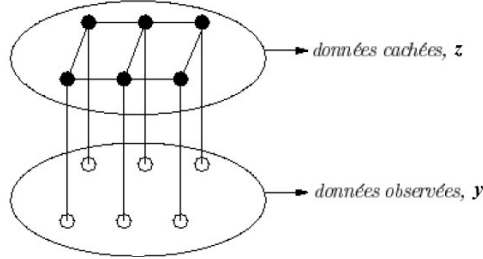
$$\forall i \in S, \forall z_i, V_i(z_i) = 0$$

$$\forall i, j \in S, \forall z_i, z_j, V_{ij}(z_i, z_j) = I_{Z_i=Z_j}(z_i, z_j)$$

Pour β positif, ce modèle donne une plus grande probabilité aux configurations où les points voisins ont la même étiquette. C'est donc un modèle adapté pour la segmentation d'images en zones homogènes. Pour le problème de la reconnaissance de texture, il est intéressant de généraliser ce modèle, ce qui se fait naturellement en autorisant des potentiels non nuls sur les singletons, et en faisant dépendre le potentiel sur les paires de z_i et z_j .

2.4. Champs de Markov cachés discrets

Dans le cadre de la segmentation d'image, on a deux types de données: d'une part celles observées qui peuvent être par exemple la couleur des pixels, et d'autre part la classification de ces pixels. On introduit alors la notion de champ de Markov caché: on place ces deux types de données sur deux niveaux différents, et on suppose que la loi de probabilité des données cachées (la segmentation) connaissant les données observées correspond à un champ de Markov. On note $y = \{y_i, i \in S\}$ l'observation, et $z = \{z_i, i \in S\}$ le résultat de la classification.



On a donc une distribution de la forme:

$$P(z|y) = W^{-1} \exp(-H(z|y))$$

De plus, on suppose que les données observées sont indépendantes conditionnellement à Z , c'est-à-dire que:

$$g(y|z) = \prod_{i \in S} g(y_i|z_i)$$

On suppose en général que ces distributions suivent des lois gaussiennes.

2.5. Champs de Markov triplets

Les modèles des champs de Markov triplets généralisent les champs de Markov cachés. Dans ces modèles, on introduit un champ auxiliaire $X = (X_i)_{i \in S}$ discret, avec $\forall i \in \mathcal{I}, Y_i \in \mathcal{L} = [1, L]$.

Les triplets considérés dans les segmentations effectuées dans la suite de notre étude sont de la forme:

$$P_G(\mathbf{y}, \mathbf{x}, \mathbf{z}) \propto \exp\left(-\sum_{i \sim j} V_{ij}(x_i, z_i, x_j, z_j) + \sum_{i \in \mathcal{I}} \log f(y_i | \theta_{x_i z_i})\right)$$

où les V_{ij} sont des potentiels sur les paires. De plus, les potentiels V_{ij} sont supposés être les mêmes sur l'ensemble des sites de sorte que nous pouvons écrire sans perte de généralité:

$$V_{ij}(x_i, z_i, x_j, z_j) = -B_{z_i z_j}(x_i, x_j) - C(z_i, z_j)$$

Un exemple simple de champ de Markov triplet est donné par :

$$P(\mathbf{y}, \mathbf{x}, \mathbf{z}) \propto \exp\left(b \sum_{i \sim j} \mathbb{1}_{x_i=x_j} \mathbb{1}_{z_i=z_j} + c \sum_{i \sim j} \mathbb{1}_{z_i=z_j} + \sum_{i \in \mathcal{I}} \log f(y_i | \theta_{x_i z_i})\right)$$

où b et c sont des réels et θ les paramètres des distributions gaussiennes.

2.6. Segmentation

Le problème qui nous intéresse est la segmentation d'une image en différentes zones de textures différentes. Pour cela, on utilise le modèle du champ de Markov caché discret. Il existe deux types de méthodes:

- on peut segmenter directement les images en un certain nombre de classes, sans effectuer d'apprentissage de motifs de texture au préalable. Les algorithmes utilisés tentent de trouver les paramètres des champs de Markov cachés et la segmentation décrivant au mieux l'image.

- mais lorsqu'on connaît à l'avance les motifs de textures à reconnaître, il est préférable d'effectuer au préalable leur apprentissage (on parle de « classification supervisée »). Lors d'une première phase, on apprend les paramètres du champ de Markov triplet correspondant aux relations entre les observations au cœur d'une texture ou entre textures différentes. Pour cela, on utilise des motifs d'apprentissage, qui ne contiennent qu'une seule texture connue, mais peuvent admettre plusieurs échantillons. Une fois que l'on connaît tous les paramètres du champ de Markov, on peut effectuer la segmentation elle-même, c'est-à-dire trouver la valeur de la segmentation Z qui maximise la probabilité conditionnellement aux données observées Y et aux paramètres appris.

2.7. Modélisation

On s'appuie sur le travail effectué dans (Blanchet *et al.*, 2005) pour la modélisation des textures. On suppose que les celles-ci peuvent se décomposer en un nombre K de sous-classes, et que pour une texture m , la distribution suit la loi:

$$P(y_i | \Psi_m) = \sum_{k=1}^K P(Z_i = c_{mk} | \Delta_m) f(y_i | \theta_{mk})$$

où $f(y_i | \theta_{mk})$ est la densité de la loi gaussienne multivariée de paramètre θ_{mk} (moyenne μ_{mk} et matrice de covariance Σ_{mk}), Z_i représente la sous-classe du point i , sa loi est paramétrée par Δ_m , et $\Psi_m = (\Delta_m, (\theta_{mk}))$ est l'ensemble des paramètres pour la texture m . On suppose que les Z_i constituent un champ de Markov comme précédemment, avec la fonction d'énergie:

$$H(z, \Delta_m) = \sum_i V_i(z_i, \alpha_m) + \sum_{j \in N(i)} V_{ij}(z_i, z_j, B_m)$$

et comme dans le modèle de Potts généralisé, on suppose que les singletons ne dépendent pas de i , soit:

$$V_i(z_i, \alpha_m) = -\alpha_m(k), \text{ pour } z_i = c_{mk}$$

et on suppose de même que:

$$V_{ij}(z_i, z_j, B_m) = -B_m(k, l), \text{ pour } z_i = c_{mk}, z_j = c_{ml}$$

2.8. Apprentissage

Pour ce modèle, l'apprentissage consiste à apprendre les différents paramètres des champs de Markov, c'est-à-dire les B_m et α_m pour chacune des textures, mais aussi les paramètres des lois gaussiennes: μ_{mk} et m_k . Pour simplifier, on suppose que les matrices de covariance sont diagonales, c'est-à-dire que les différentes

composantes des descripteurs sont indépendantes. Pour effectuer l'apprentissage, on sélectionne des images contenant une seule texture. On segmente ces images par des algorithmes non supervisés (de type NREM) en K sous-classes, ce qui donne les paramètres du champ de Markov, ainsi que les moyennes et variances des densités.

2.9. Classification

Pour améliorer le résultat, on initialise les valeurs du champ de Markov à l'aide d'une pré classification (ex: KMeans): on associe à chaque pixel la classe qui maximise la probabilité a priori que le pixel lui appartienne, sans tenir compte des pixels voisins (on connaît les densités de probabilité de l'observation pour chaque classe). On effectue ensuite une classification supervisée de l'image, c'est-à-dire en fixant les paramètres du champ de Markov et des densités gaussiennes à l'aide des valeurs trouvées précédemment, et en utilisant l'un des algorithmes: *ICM* (iterational conditional modes), *Mean Field* ou *Simulated Field* (Blanchet, 2007).

2.10. Le logiciel SpaCEM³

Développé au sein de l'équipe MISTIS de l'INRIA Rhône Alpes, (SpaCEM³, 2007) est l'aboutissement de plus de dix ans de recherches. Ecrit en C++, il propose une variété d'algorithmes pour la classification de données uni ou multidimensionnelles en interaction, potentiellement de très grande dimension. Les algorithmes implémentés permettent de classer des données multimodales et dépendantes du fait de leur localisation spatiale ou du fait d'autres types de relations décrites par des structures graphiques quelconques. Ceci inclut la segmentation d'images, avec comme structure de dépendance sous-jacente des grilles régulières de pixels. L'approche se fonde sur l'utilisation d'algorithmes de type EM (Celeux *et al.*, 2003) pour une classification floue et sur les modèles de champs de Markov pour la modélisation des dépendances. Le logiciel inclue les fonctionnalités suivantes:

- classification non supervisée d'individus, basée sur une description des dépendances à l'aide d'un graphe non nécessairement régulier et un traitement basé sur les champs de Markov cachés. Les modèles markoviens disponibles incluent diverses extensions du modèle de Potts standard.
- classification supervisée d'individus, basée sur la famille de modèles de Markov triplets, avec des phases d'apprentissage et de test.
- critères de sélection (BIC, ICL) permettant de sélectionner "le meilleur" modèle de champ de Markov caché en fonction des données.

3. Catalogue de matières à reconnaître

Pour les tests effectués, nous avons extrait un catalogue de types de sol avec 48 motifs de texture associés, de taille 128 x 128 pixels (Figure.1): neige (poudreuse, cristallisée, fondante, sale), glace (blanche, bleue, grise), eau (torrent, lac, rivière), forêts (plusieurs types), maquis (plusieurs types), arbres isolés, prairie (plusieurs

types), roche (plusieurs types), rochers et éboulis (plusieurs tailles), roche végétalisée (plusieurs types), mélange prairie / roche / caillou, culture (plusieurs types), chemin de terre, bitume (route, sol urbain, parking) et toitures (plusieurs types). Comme il est impossible de définir une « classe poubelle » pour des pixels ne correspondant à aucun des motifs, cela oblige à fournir un catalogue le plus complet possible, à partir de la banque de photographies. De plus, le fait d'apprendre ces motifs en les décomposant en sous-classes (principe fondamental des méthodes utilisées) est un avantage pour la classification des textures naturelles composites. Par exemple, dans une forêt, il y a la canopée, le sol et l'ombrage, qui vont être appris ensemble, mais comme trois sous classes d'un même matériau « forêt ».

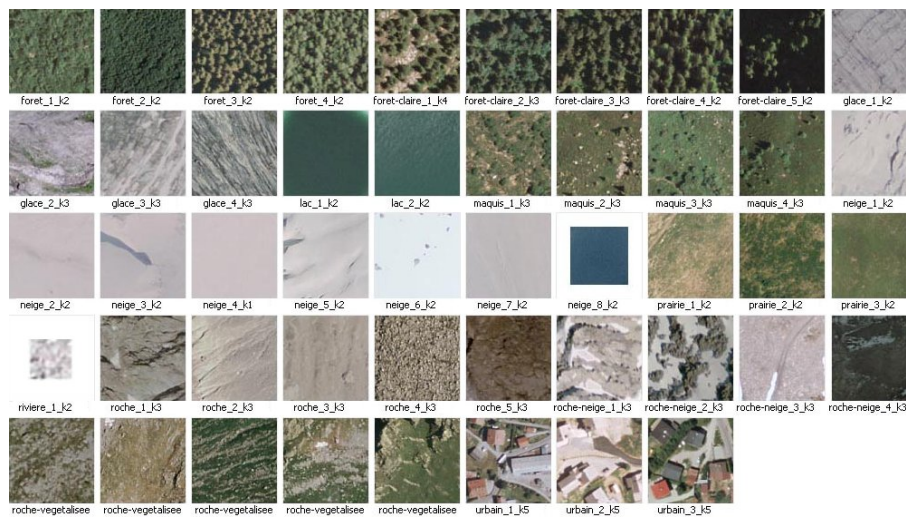


Figure 1. Les 48 motifs composant la base d'apprentissage.

4. Calcul des vecteurs descripteurs

4.1. Choix des paramètres

L'utilisation des champs de Markov offre une méthode pour caractériser un motif, en avoir un modèle implicite, à partir d'observations locales. Pour ce faire, un premier objectif est de construire un vecteur de propriétés (features) de texture qui sont les plus significatives pour la classification. Ces mesures effectuées en chaque pixel sont les seules informations dont disposent les algorithmes d'apprentissage et de classification. Par conséquent, elles doivent être judicieusement choisies, même si cela reste très expérimental, ce qui fait tout l'intérêt de ce type de recherches.

Avant l'apparition des modèles triplets, il était vivement souhaitable que ces features soient le plus possible décorréelées, pour ne pas nuire à la classification. On en avait donc un nombre limité, car il est difficile de trouver de nombreuses propriétés sur les pixels à la fois décorréelées et significatives. Par exemple dans (Premoze *et al.*, 1999), les 4 premières propriétés sont très fortement corrélées (3

composantes rgb et leur moyenne $(r + g + b) / 3$). On ne devait pas les stocker ensemble dans le vecteur descripteur, mais cela n'a plus lieu d'être avec les triplets.

Afin de permettre une segmentation efficace des différents motifs, nous avons recherché dans la littérature de l'analyse des textures des propriétés locales caractéristiques pour chaque pixel de l'image: colorimétriques (couleur, luminosité) et surfaciques (forme, rugosité, répétition de motifs).

4.2. Caractérisation colorimétrique

La question qui se pose naturellement est celle de l'espace de codage des couleurs. Nous avons de fait testé trois codages usuels: rgb (trichrome), yuv (luminance, chrominances) et hsv (teinte, saturation, luminosité). De manière prévisible, le codage rgb habituel, non décorrélé, donne de mauvais résultats. De meilleurs sont obtenus avec les composantes. Les tests effectués montrent que les composantes des codages yuv [1], mais surtout hsv (Figure.2) sont plus "indépendantes" que rgb , et plus discriminantes dans le cadre de la reconnaissance de textures.

$$\begin{aligned} y &= 0.299 r + 0.587 g + 0.114 b \\ u &= 0.492 (b - Y) \\ v &= 0.877 (r - Y) \end{aligned} \quad (1)$$

Après quelques tests très significatifs, il s'avère que l'espace de couleur hsv est aussi plus efficace que l'espace yuv . Avec $MAX = \max(r, g, b)$ et $MIN = \min(r, g, b)$:

$$h = \begin{cases} 0, & \text{si } MAX = MIN \\ 60 \times \frac{g-b}{MAX-MIN} + 0^\circ, & \text{si } MAX = r \\ & \text{et } g \geq b \\ 60 \times \frac{g-b}{MAX-MIN} + 360^\circ, & \text{si } MAX = r \\ & \text{et } g < b \\ 60 \times \frac{b-r}{MAX-MIN} + 120^\circ, & \text{si } MAX = g \\ 60 \times \frac{r-g}{MAX-MIN} + 240^\circ, & \text{si } MAX = b \end{cases} \quad s = \begin{cases} 0, & \text{si } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{sinon} \end{cases} \quad v = MAX$$

Figure 2. Formules de calcul des composantes hsv en fonction de rgb .

4.3. Caractérisation surfacique

Une texture de neige est très différente en surface d'une texture de forêt, même sans tenir compte de la couleur. Nous avons d'abord expérimenté plusieurs mesures provenant du traitement du signal:

- les filtres de Gabor (Recio *et al.*, 2005),
- les critères d'Haralick (Miyamoto *et al.*, 2005),
- les coefficients de la Karhunen-Loeve Transform KLT (Devaux *et al.*, 2000),

puis des mesures de la rugosité de surface:

- gradients des différentes composantes de couleur (yuv , hsv),
- mesure de contraste LIP (Jourlin *et al.*, 2001) pour tenir compte des ombres,

- mesures de la dimension fractale, avec ou sans filtrage.

Les différents paramètres ont été testés seuls ou en combinaison. Il n'est pas possible de montrer ici l'ensemble des résultats issus de toutes les expérimentations. Nous donnons simplement les conclusions auxquelles nous sommes parvenus. Après une série de tests qui se sont révélés peu probants sur l'utilisation de l'espace KLT, des filtres de Gabor et des critères d'Haralick, nous avons expérimenté avec la dimension fractale, les gradients des composantes hsv , et une mesure de contraste. Le gradient de h caractérisant la différence de teinte en un point donné est trop sensible, et gêne d'obtention de bons résultats. Les gradients de s et de v s'avèrent beaucoup mieux représenter la notion de « grain » ou de « rugosité » (Figure.5).

4.4. Dimension fractale

Elle nous a semblé être utile comme mesure du grain local de la texture autour de chaque pixel. Plusieurs méthodes existent pour calculer la dimension fractale d'une image, et nous en avons essayé deux, décrites dans (Chaudhuri *et al.*, 2005): celle basée sur le spectre de puissance d'un signal fractal bidimensionnel, et la méthode classique par comptage, basée sur l'autosimilarité. La première ne fonctionne pas très bien, car, en pratique, pour un signal bruité, le résultat est souvent aberrant. Il peut être nécessaire de ne travailler que sur une partie du spectre, mais il est difficile de savoir laquelle à l'avance. La seconde est plus souple à utiliser: si un ensemble est constitué de N_r copies de lui-même avec un facteur d'échelle r , alors la dimension fractale D de l'objet est donnée par: $D = \log(N_r) / \log(1/r)$. Le calcul classique se fait dans un espace en 3 dimensions, pour différentes valeurs de r , et il suffit ensuite d'utiliser une méthode de moindres carrés pour avoir une estimation de D . On applique enfin un filtre lisseur afin d'améliorer la qualité de la segmentation.

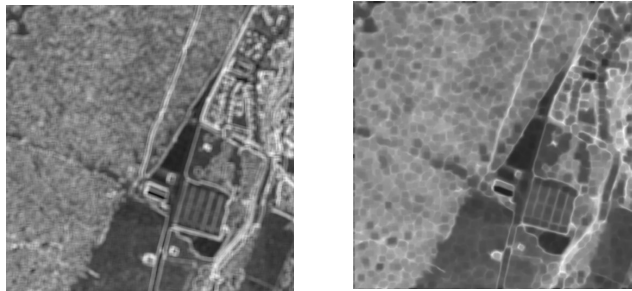


Figure 3. Images de la dimension fractale avant et après lissage.

5. Expérimentations et résultats

5.1. Le programme IFE

IFE a été conçu avec trois fonctionnalités majeures: l'écriture automatique de batchs (qui simplifient le lancement de tests multiples et permettent de commander

des calculs à SpaCEM³), le calcul des vecteurs descripteurs des images avec écriture dans un fichier .dat, la consolidation et la conversion des données issues de la classification en planches de résultats et en fichiers interprétables. Dans la version actuelle, seuls les paramètres (*h, s, v, grad_s, grad_v*) ont été conservés (section 4).

Un test complet sur un jeu d'images comprend les étapes suivantes:

- extraction des features pour chaque image du jeu (IFE),
- apprentissage d'un catalogue de motifs (SpaCEM³),
- préclassification des images du jeu (SpaCEM³),
- classification des images du jeu (SpaCEM³),
- consolidation des résultats de la classification (IFE),
- conversion des résultats en planches d'images facilement interprétables (IFE).

Le répertoire PATTERNS_LEARNED contient la liste des motifs appris, dont le nombre de sous classes est estimé visuellement par l'utilisateur. A chaque sous-classe identifiée est associée la couleur moyenne de ses pixels pour permettre de légendier automatiquement les planches de résultats. Chaque test est entièrement spécifié dans un fichier texte (ou XML), dont la structure est la suivante:

- ligne 1 : nom de la texture à analyser (ex: texture-foret-maquis),
- ligne 2 : liste de noms de patterns à prendre en compte pour l'apprentissage (ex: foret_clairière_3_k3 foret_2_k2 maquis_1_k3 prairie_2_k2),
- ligne 3 : liste des features à prendre en compte pour ce test. Les features sont codées dans le programme IFE et lui seront par la suite passées en paramètre (ex: *hsv* avec 3 paramètres et les gradients de *s* et *v*: *hsv grad_s 1 grad_v 1*),
- ligne 4 : algorithme utilisé (Simulated Field, Mean Field ou ICM),
- ligne 5 : nombre maximum d'itérations de l'algorithme (ex: 150).

Il est sauvé dans un répertoire unique TEST_n contenant 11 fichiers de résultats: descriptif du test, texture originale, descripteur d'image généré par le metascript PHP, vecteurs descripteurs calculés en chaque pixel, probabilités d'appartenance de chaque pixel à chaque sous classe, en chaque pixel la sous classe ayant la probabilité la plus forte, les paramètres du modèle markovien après classification, les résultats numériques de la consolidation, les résultats en couleurs de la consolidation, la précision de la classification, égale au rapport des deux probabilités des sous classes majoritaires, et la même information, mais en niveaux de gris.

5.2. Résultats et commentaires

On a d'abord observé que les résolutions des motifs d'apprentissage et des images à classifier doivent être identiques. Les algorithmes utilisables dans SpaCEM³ en classification supervisée sont au nombre de 4: *ICM, Mean Field, Modal Field* et *Simulated Field*, et plusieurs configurations de paramètres ont été testées pour chacun. La précision des résultats est obtenue par un indicateur simple consistant à faire la division de la plus grande probabilité d'appartenance à une classe par la 2e plus grande probabilité. Il est majoritairement bien supérieur à 1 dans nos essais.

La Figure.4 montre les résultats obtenus avec une image typique contenant principalement des textures d'herbe, de forêt et de route. Pour (b), on a utilisé comme vecteur descripteur $(y, u, v, \text{grad}_y, \text{grad}_u, \text{grad}_v, D)$, et pour (c) le vecteur $(h, s, v, \text{grad}_s, \text{grad}_h)$. Les motifs étaient décomposés en 3 sous-classes au cours de l'apprentissage, et c'est l'algorithme ICM qui a servi pour la classification. On constate que celle-ci est globalement satisfaisante pour (c), où les zones de forêt, d'herbe et de route sont bien détectées, et les frontières suffisamment précises. Il n'en est pas de même pour (b). Comme la dimension fractale est calculée sur une fenêtre d'une dizaine de pixels de largeur, les détails de moins de 10 pixels ont tendance à être perdus, ce qui va à l'encontre de l'objectif escompté. Certains chemins ne sont pas détectés correctement, ainsi que les zones herbeuses en forêt. Il est intéressant de remarquer que la Figure.3 fait nettement ressortir les contours des objets qui composent l'image, et qu'en ce sens, on est en droit de se demander ce que le calcul de la dimension fractale apporte de plus par rapport à un opérateur classique de détection de contour (gradient par exemple).

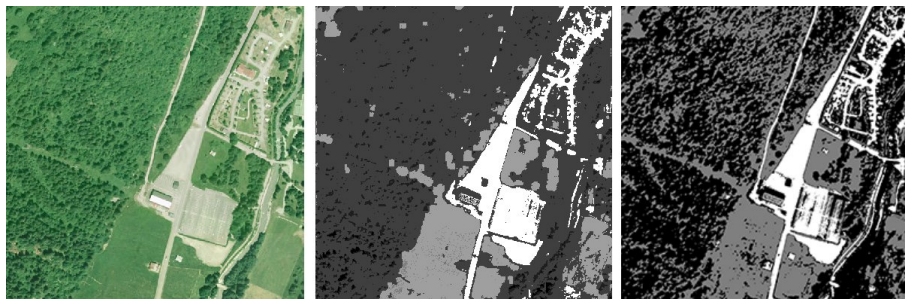


Figure 4. Classification en 3 classes avec l'algorithme ICM. (a) image native. (b) classification médiocre. (c) bonne classification.

5.3. Temps de calcul

Alors que la phase d'apprentissage de l'ensemble des motifs du catalogue est assez rapide, les temps de calcul pour la classification sont très lents (de 0,5h à 4h pour une image de 800 x 800 pixels avec 7 motifs et 15 sous-classes, selon qu'on utilise ICM, ou *Field). Le choix du nombre de motifs à utiliser influe fortement sur les temps de calcul et sur la mémoire occupée (prévoir 1 Go de RAM pour l'exemple décrit ci-dessus), ce qui rend long le processus de classification avec un catalogue exhaustif, et par conséquent difficile le traitement systématique d'un jeu d'images de vastes terrains. La complexité du traitement (temps de calcul et mémoire) en fonction de la largeur de l'image est quadratique, ce qui impose de travailler avec des images de taille moyenne (moins de 1000 x 1000 pixels). De plus, il est difficile de trouver un nombre optimal d'itérations qui minimise le temps de calcul tout en conservant la qualité des résultats, d'autant plus qu'ICM converge, ce qui est rarement le cas des algorithmes de type *Field.

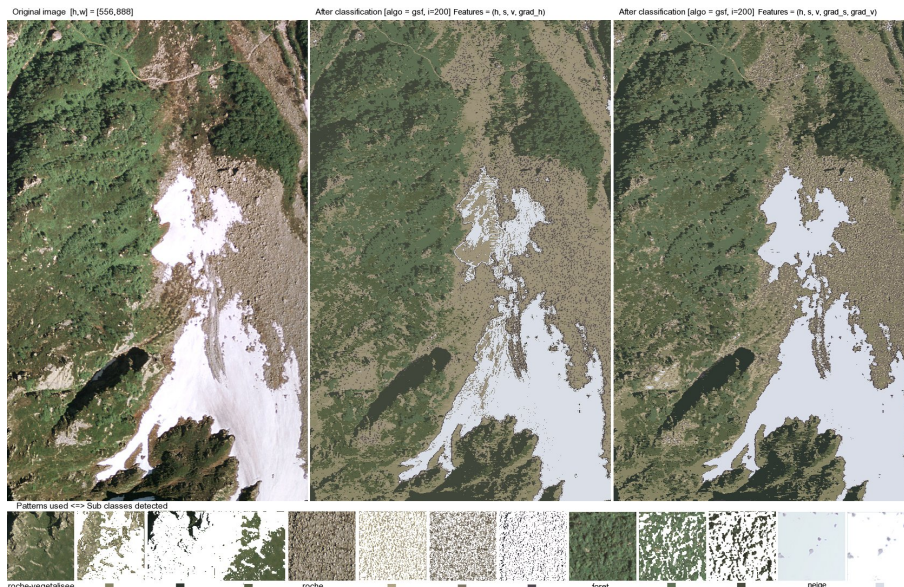


Figure 5. Segmentation d'une image en 4 classes (maquis, roche, forêt, neige) avec l'algorithme Mean Field. (a) image native. (b) classification médiocre avec *grad_h*. (c) bonne classification avec *grad_s* et *grad_v*. (d) motifs et leurs sous-classes.

6. Applications

Dans le cadre du projet DEREVE II, nous avons développé un navigateur de terrains de grandes dimensions incluant l'amplification végétale (Marsault *et al*, 2008). Cette technique a ensuite été transférée dans la plateforme PROLAND du laboratoire EVASION de l'INRIA. Dans un tel environnement, on fait apparaître en temps réel, lorsqu'on se rapproche suffisamment du sol, la végétation qui doit être visible. Celle-ci n'est générée qu'en fonction des contraintes du point de vue, et il est donc nécessaire de connaître localement la nature du sol, pour pouvoir synthétiser à la volée les éléments de végétation. Cette information (land cover classification) nous est donnée (en amont bien sûr) par la classification des images aériennes décrite plus haut. Les premières expérimentations ont donné d'excellents résultats (Figure 6) suite à la reconnaissance des massifs de forêts et d'arbres de plusieurs essences sur une zone de 16 km² autour de Megève. Ces résultats sont un peu meilleurs (en particulier moins de perte de petits bosquets ou d'arbres isolés) que ceux que nous a fournis en 2007 le laboratoire MATIS de l'IGN, obtenus avec le logiciel d'extraction de forêts décrit dans (Trias-Sanz *et al*, 2005).

7. Conclusions et perspectives

Durant les phases de tests, on a montré que l'espace de couleur *hsv* aide à bien reconnaître des motifs distincts dont les couleurs sont visuellement proches, et que la rugosité est assez bien décrite par le gradient des composantes *s* et *v*. Mais il est

difficile de déterminer le set optimal de features. Malgré les bons résultats obtenus dans IFE, cela reste donc, à notre avis, un problème ouvert. Les résultats obtenus avec la dernière version de SpaCEM³ sont très bons, mais on a noté, au niveau de la classification, une légère différence entre les algorithmes **Field* et *ICM* (moins de 5% des pixels sont classés différemment), ce qui méritera des tests plus approfondis.



Figure 6. *Amplification végétale de terrains naturels autour de Megève (avec l'aimable autorisation d'Eric Bruneton, INRIA-EVASION).*

Il est difficile d'obtenir des orthophotographies avec un soleil parfaitement zénithal lors d'une longue campagne aérienne d'acquisition. De fait, la présence d'ombres portées, même minime, ajoute un handicap pour la reconnaissance des textures. Le problème est alors de parvenir à reconnaître que deux textures, l'une dans l'ombre et l'autre éclairée, représentent la même matière. Certains facteurs, comme la mesure de contraste LIP, ont été rajoutés pour tenter de résoudre le problème, mais les résultats n'ont pas été satisfaisants. Sauf à apprendre les matières à l'ombre et à la lumière, le problème de la reconnaissance des textures dans les zones d'ombre n'est donc pas complètement résolu à ce jour.

Les extensions récentes des codes de calcul SpaCEM³ et leur interface graphique conviviale sous Linux et Windows permettent désormais la classification automatique et « à la chaîne » (en mode batch) d'une banque d'images à partir d'un catalogue de motifs appris, moyennant l'utilisation d'une station de calcul puissante équipée de nombreux Go de RAM. Depuis 2005, les expérimentations réalisées avec IFE ont cependant permis de diminuer les temps de calcul, et de rendre les données en sortie d'analyse plus facilement exploitables pour les applications. Si les temps de calcul restent encore élevés, il nous semble important de les mettre en balance avec la bonne qualité des résultats obtenus.

Remerciements

MAP-ARIA tient à remercier le laboratoire MISTIS pour son soutien, sa compétence, et particulièrement Juliette Blanchet et Florence Forbes, toujours disponibles et patientes, sans qui ce travail n'aurait pu voir le jour.

8. Bibliographie

- Amara Y., Marsault X., A GPU Tile-Load-Map Architecture for Terrain Rendering: Theory and Applications, *The Visual Computer*, Int. Journal on Graphics, Springer, 2008.
- Aubry M., Marsault X., Rapport de stage: « Analyse d'images aériennes de haute-résolution du département de Haute Savoie », MAP-ARIA, 2006, <http://giik.net/photos-satellite-segmentation>.
- Blanchet J., Modèles markoviens et extensions pour la classification de données complexes. Thèse de l'Université Joseph-Fourier, Grenoble, 2007.
- Blanchet J., Forbes F., Triplet Markov fields for the supervised classification of complex structure data, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(6): 1055-1067, 2008.
- Blanchet J., Forbes F., Schmid, C., Markov Random Fields for Recognizing Textures Modeled by Feature Vectors, *International Conference on Applied Stochastic Models and Data Analysis*, France, 2005.
- Celeux G., Forbes F., Peyrard N., EM procedures using mean field-like approximation for Markov model-based image segmentation, *Pattern Recognition.*, 36(1):131-144, 2003.
- Chaudhuri B., Sarkar N., Texture Segmentation Using Fractal Dimension, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 17, n°1, 1995.
- Devaux J.C., Gouton P., Truchetet F., Aerial Colour Image Segmentation by Karhunen-Loeve Transform, *Proceedings of the 15th International Conference on Pattern Recognition*, vol 1, p309-312, 2000.
- Fortin J., Marsault X., Rapport de Stage: « Analyse d'images aériennes de haute résolution de Haute Savoie », MAP-ARIA, http://www.aria.archi.fr/equipe/Analyse_textures_2005.pdf
- Jourlin M., Pinoli J.C., Logarithmic Image Processing : The mathematical and physical framework for the representation and processing of transmitted images, *Advances in Imaging and Electron Physics*, vol 115, 2001.
- Miyamoto E., Merryman T., Fast Calculation of Haralick Texture Features: www.ece.cmu.edu/~pueschel/teaching/18-799B-CMU-spring05/material/eizan-tad.pdf.
- Mouret M., Solnon C., Wolf C., Classification of images based on Hidden Markov Models, *IEEE Workshop on Content Based Multimedia Indexing*, p169-174, 2008.
- Newsam S., Bhagavathy S., Manjunath B.S., Modeling object classes in aerial images using hidden markov models, *Proc. of the International Conference on Image Processing*, 2002.
- Premoze S., Thompson, W.B., Shirley, P., Geospecific Rendering of Alpine Terrain. Department of Computer Science, University of Utah, 1999.
- Recio J.A., Ruiz L.A., Fernandez A., Use of Gabor filters for texture classification of digital images. Department of Computer Science, University of Utah, 2005.
- RGD73-74, Régie de Gestion des Données des deux Savoie: <http://www.rgd73-74.fr>.
- Trias-Sanz, R., Boldo, D., A High-Reliability, High-Resolution Method for Land Cover Classification Into Forest and Non-forest, *14th Conf. on Image Analysis*, Finland, 2005.
- SpacCEM³ (Spatial Clustering with EM and Markov Models), programme à télécharger sur spacem3.gforge.inria.fr ou mistis.inrialpes.fr/software/SpaCEM3.tgz, 2007.