Modèles markoviens pour l'organisation spatiale de descripteurs d'images.

J.Blanchet, F.Forbes, C.Schmid

INRIA Rhône-Alpes
ZIRST-655 avenue de l'Europe
38330 Montbonnot Saint Martin

Résumé : Ce papier décrit une nouvelle approche probabiliste pour la reconnaissance de textures. Une image est décrite à l'aide de descripteurs locaux, ainsi que par des relations spatiales entre ces descripteurs. On peut alors associer une image à un graphe : les nœuds sont les points d'intérêt de l'image correspondant à des régions caractéristiques et les arêtes relient des régions voisines. Ajouter une telle information de voisinage permet d'améliorer les résultats de reconnaissance. Les approches actuelles consistent à modéliser les descripteurs comme des variables indépendantes, puis à rajouter l'information spatiale par le biais de poids, sans modéliser explicitement ces dépendances. Nous proposons d'introduire un modèle statistique rendant compte directement de cette dépendance entre descripteurs, par l'utilisation de champs de Markov cachés. L'estimation des paramètres de tels modèles étant en pratique difficile, nous utilisons des procédures d'estimation récentes basées sur le principe du champ moyen de la physique statistique. Nous illustrons notre méthode sur la reconnaissance d'images uni et multi-textures. Les résultats obtenus sont prometteurs.

Mots-clés : Champ de Markov, Algorithme de type EM, Relaxation, Classification, Apprentissage statistique, Reconnaissance de textures.

1 Introduction

Une notion clé en vision par ordinateur est celle de descripteurs, caractérisations locales d'une image. De manière générale, un bon descripteur se doit d'être résistant aux occlusions, ainsi qu'invariant à diverses transformations géométriques de l'image. La recherche de "bons" descripteurs a déjà fait l'objet de nombreuses études, alors que la prise en compte de leur organisation spatiale reste un problème très ouvert. Nous proposons donc de coupler l'utilisation d'outils performants de vision et de statistique, dans le but de modéliser l'organisation spatiale de tels descripteurs.
Une tentative de prise en compte du caractère spatial des données pour la reconnaissance de textures a déjà été effectuée dans (Lazebnik *et al.*, 2003a) : dans ce travail, lors de la phase de reconnaissance, les probabilités *a posteriori* d'appartenance aux différentes classes de textures sont raffinées par l'algorithme de relaxation proposé dans (Rosenfeld *et al.*, 1976). Cependant, le voisinage n'y est pris en compte que par un

terme de poids, sans modèle explicite. Nous proposons de modéliser les descripteurs comme des variables statistiques liées, et par conséquent d'utiliser un modèle statistique paramétrique rendant compte explicitement de ces dépendances. Le modèle que nous avons choisi est celui du champ de Markov caché. L'estimation des paramètres d'un tel modèle étant difficile, nous utilisons des procédures d'estimation récentes (algorithme de type EM), basées sur l'algorithme d'*Expectation-Maximisation* (EM) et sur le principe du champ moyen issu de la physique statistique (Chandler, 1987).
La base d'apprentissage considérée dans notre présentation n'est composée que d'images uni-texturées. Cependant, les algorithmes de type EM utilisés pour l'apprentissage pourraient tout à fait être généralisés à des images multi-textures pourvu que soit déclaré l'ensemble des textures qu'elles contiennent. On pourra se référer à (Lazebnik *et al.*, 2003a) ou (Nigam *et al.*, 2000) pour plus de détails. Pour ce qui est de la reconnaissance, la méthode ne se restreint pas aux images uni-textures : chaque vecteur caractéristique est classé individuellement, si bien que les images test peuvent être multi-textures (voir Section 6.2).
Dans la Section 2, nous présentons brièvement la procédure d'extraction de caractéristiques, et la façon dont est construit le graphe de voisinage. Le modèle probabiliste utilisé (champs de Markov cachés) pour les textures est explicité dans la Section 3. Les différentes phases d'apprentissage et de reconnaissance sont décrits dans la Section 4. Nous présentons pour comparaison en Section 5 deux autres algorithmes de classification de textures : la relaxation et l'algorithme NEM. Des expériences effectuées sur des images de scènes d'intérieur ainsi qu'une discussion achèvent ce papier.

2 Extraction de caractéristiques et graphe de voisinage

Pour la phase d'extraction de caractéristiques, nous suivons la méthode décrite dans (Lazebnik *et al.*, 2003a) pour ses performances en comparaison à d'autres méthodes récentes (Bradshaw *et al.*, 2001; Kumar & Hebert, 2003; Malik *et al.*, 2001; Schmid, 2001). Le détecteur utilisé est le *Laplacien avec adaptation affine*. Brièvement, les points d'interêt détectés correspondent à des maximum locaux dans l'espace échelle du laplacien normalisé de l'intensité. A ces points sont associés des cercles de rayon l'échelle correspondante, cercles transformés en ellipse par le processus d'*adaptation affine* (Lindeberg & Garding, 1997). Outre ses performances (notamment l'invariance aux transformations affines), l'intérêt d'un tel détecteur est de permettre la définition naturelle d'un graphe de voisinage (Lazebnik *et al.*, 2003a). Un point i sera dit voisin de j s'il appartient à l'ellipse centrée sur j, grossie d'un certain nombre de pixels (15 dans notre présentation) le long de chacun de ses axes (ceci pour éviter que les points associés à de petites régions aient trop peu de voisins). Notons qu'un tel voisinage est non symétrique. On peut alors voir une image comme un graphe orienté, chaque arcs reliant un point d'interêt détecté à un autre point voisin. Dans l'optique d'utiliser les champs de Markov, définis sur un graphe non orienté, nous avons symétrisé ce voisinage, en remplaçant les arcs par des arêtes (voir figure 1). A chacune des régions (ellipses) détectées, est associé un vecteur caractéristique (descripteur). Les descripteurs utilisés sont des vecteurs de dimension 80 obtenus à partir de *spin images* (Lazebnik *et al.*, 2003b) , obtenues de la manière suivante : chaque ellipse détectée est transformée

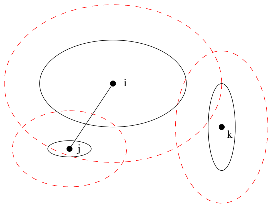

FIG. 1 – Graphe de voisinage symétrique : les points i et j sont voisins

en cercle unité, sur lequel on calcule un histogramme à 2 dimensions : une tranche de la spin image correspondant à la distance d est l'histogramme de l'intensité des pixels situés à la distance d du centre du cercle. Pour notre expérimentation, nous avons pris des spin images de taille 5×16. D'autres descripteurs sont envisageables et la méthode que nous proposons s'applique de même. A chaque point d'intérêt détecté est donc associé un vecteur caractéristique de dimension 80.

3 Modélisation des textures

L'hypothèse sur laquelle se fonde nos travaux est que les descripteurs sont des variables aléatoires dépendantes, de loi de probabilité spécifique pour chaque texture. Dans (Lazebnik *et al.*, 2003a), la distribution des descripteurs issus d'une texture donnée est modélisée par un mélange de gaussiennes, ce qui suppose que les descripteurs sont des variables indépendantes. Il existe pourtant de manière évidente une forte dépendance spatiale entre les vecteurs caractéristiques d'une même image. Afin de prendre en compte cette dépendance, nous proposons de modéliser leur distribution par un champ de Markov caché, de paramètres inconnus.

Soit $\mathbf{x} = (x_1, \ldots, x_n)$ les n descripteurs (vecteurs de dimension 80) extraits d'une image (ou partie d'image) de la texture m ($1 \leq m \leq M$). On suppose que chaque texture est composée de K sous-classes $c_{m1} \ldots c_{mK}$. Nous prendrons $K = 10$ dans nos expérimentations, indépendemment de la texture, mais la méthode resterait tout à fait identique pour un $K(m)$ dépendant de m. Pour $i = 1, \ldots, n$, on modélise la probabilité d'observer le descripteur x_i pour une image de la texture m par :

$$P(x_i|\mathbf{\Psi}_m) \;=\; \sum_{k=1}^{K} P(Z_i = c_{mk}|\mathbf{\Delta}_m)\; f(x_i|\mathbf{\Theta}_{mk}), \tag{1}$$

où $f(x_i|\mathbf{\Theta}_{mk})$ dénote la distribution gaussienne multivariée de paramètres $\mathbf{\Theta}_{mk}$ (la moyenne μ_{mk} et la matrice de covariance Σ_{mk}). La variable aléatoire Z_i représente la sous-classe à laquelle appartient le descripteur x_i ; elle peut prendre les valeurs $\{c_{m1}, \ldots, c_{mK}\}$, et sa loi est paramétrée par $\mathbf{\Delta}_m$. $\mathbf{\Psi}_m = (\mathbf{\Delta}_m, (\mathbf{\Theta}_{mk})_{1 \leq k \leq K})$ dénote l'ensemble des paramètres du modèle pour la texture m.

Dans (Lazebnik *et al.*, 2003a), le modèle utilisé pour $P(x_i|\mathbf{\Psi}_m)$ est le mélange gaus-

sien, ce qui revient à supposer que les Z_i sont des variables *indépendantes* et que $P(Z_i = c_{mk}|\mathbf{\Delta}_m) = p_{mk}$ indépendemment du site i. $\mathbf{\Delta}_m$ est alors l'ensemble des proportions $(p_{mk})_{1\leq k\leq K}$ du mélange pour la texture m. Cela implique que les decripteurs sont également des variables indépendantes.
C'est cette limitation que nous nous proposons de surmonter, en supposant que les descripteurs sont des variables *dépendantes*. Plus précisément, les dépendances entre descripteurs voisins sont modélisées en considérant que la loi jointe des variables $Z_1, \ldots, Z_n$ est un champ de Markov discret sur le graphe défini à la section 2. Soit $\mathbf{z} = (z_1, \ldots, z_n)$ des réalisations des Z_i. On définit :

$$P(\mathbf{z}|\mathbf{\Delta}_m) \quad = \quad W(\mathbf{\Delta}_m)^{-1} \exp(-H(\mathbf{z}, \mathbf{\Delta}_m)) \tag{2}$$

où $W(\mathbf{\Delta}_m)$ est une constante de normalisation et H une fonction énergie supposée être de la forme (nous nous limitons aux interactions entre paires) :

$$H(\mathbf{z}, \mathbf{\Delta}_m) = \sum_i V_i(z_i, \boldsymbol{\alpha}_m) + \sum_{i\sim j} V_{ij}(z_i, z_j, \mathbb{B}_m)$$

(la notation $i \sim j$ signifie que les sites i et j sont voisins ; la somme de droite ne porte donc que sur des sites voisins).

Les fonctions V_i et V_{ij} se rapportent respectivement aux potentiels sur les singletons et sur les paires, de paramètres respectifs $\boldsymbol{\alpha}_m$ et $\mathbb{B}_m$. Il s'en suit $\mathbf{\Delta}_m = (\boldsymbol{\alpha}_m, \mathbb{B}_m)$.
Nous supposons que les potentiels sur les singletons ne dépendent que de la valeur z_i (et non de i), c'est à dire :

$$V_i(z_i, \boldsymbol{\alpha}_m) \quad = \quad -\alpha_m(k) \quad \text{si } z_i = c_{mk}$$

Les potentiels sur les singletons sont donc caractérisés par K poids $\boldsymbol{\alpha}_m = (\alpha_m(k))_{1\leq k\leq K}$ associés aux K sous-classes de la texture m. Dans le cas particulier où les V_{ij} sont nuls (ce qui revient à supposer qu'il n'y a pas d'interactions entre les points), $\mathbf{\Delta}_m$ se résume à $\boldsymbol{\alpha}_m$ et d'après (2),

$$P(Z_i = c_{mk}|\boldsymbol{\alpha}_m) = \frac{e^{\alpha_m(k)}}{\sum_{l=1}^{K} e^{\alpha_m(l)}}$$

si bien que $\boldsymbol{\alpha}_m$ pondère l'importance relative des différentes sous-classes de la texture m.
De même, les potentiels sur les paires V_{ij} sont supposés ne dépendre que de z_i et z_j, soit

$$V_{ij}(z_i, z_j, \mathbb{B}_m) \quad = \quad -B_m(k,l) \quad \text{si } z_i = c_{mk}, z_j = c_{ml}$$

Ils sont donc caractérisés par la matrice $\mathbb{B}_m = (B_m(k,l))_{1\leq k,l\leq K}$. Notons que si $\mathbb{B}_m = \beta_m \times Id$, le paramètre spatial $\mathbb{B}_m$ se réduit à un scalaire β_m et nous retombons sur le modèle de Potts traditionnel utilisé en segmentation d'images.
La texture m est donc représentée par un champ de Markov caché paramétré par $\mathbf{\Psi}_m = (\boldsymbol{\alpha}_m, \mathbb{B}_m, (\mathbf{\Theta}_{mk})_{1\leq k\leq K})$.

4 Apprentissage et classification

Dans la modélisation précédente, les paramètres $\mathbf{\Psi}_m = (\mathbf{\Delta}_m, (\mathbf{\Theta}_{mk})_{1\leq k\leq K})$ sont inconnus et doivent être estimés pour chaque texture m.

4.1 Apprentissage des paramètres inconnus

Pour apprendre le modèle associé à chacune des textures, on suppose que l'on dispose d'une base d'apprentissage constituée d'images identifiées comme appartenant à l'une des textures. Chaque texture va alors être apprise successivement sur les images correspondantes. Pour la texture m, doivent être estimés les paramètres $(\mu_{mk}, \Sigma_{mk})_{1\leq k\leq K}$ des lois gaussiennes ainsi que les paramètres du champ spatial, *ie* le vecteur $\boldsymbol{\alpha}_m = (\alpha_m(k))_{1\leq k\leq K}$ de K poids et la matrice d'interaction spatiale $\mathbb{B}_m$ de dimension $K \times K$. L'algorithme EM (Dempster *et al.*, 1977) est couramment utilisé pour l'estimation de paramètres dans le cas de données cachées, et en particulier pour l'estimation de mélanges indépendants. Pour de tels modèles, l'hypothèse d'indépendance mène à une implémentation facile de l'algorithme. Pour les champs de Markov cachés, du fait de la dépendance des données, l'algorithme n'est pas utilisable en pratique, et des approximations sont donc nécessaires. Dans ce papier, nous utilisons une des approximations de (Celeux *et al.*, 2003), basée sur le principe du champ moyen. Les procédures qui en découlent ont la particularité de prendre en compte la structure markovienne tout en préservant les bonnes qualités de EM. L'algorithme EM variationnel de (Zhang, 1992) en est un cas particulier. Nous utiliserons cependant plutôt les algorithmes de *type champ moyen* (et en particulier l'algorithme en *champ simulé*) pour leurs performances dans le cadre de la segmentation (Celeux *et al.*, 2003). Notons cependant qu'il est nécessaire de généraliser ces algorithmes pour permettre l'estimation de la matrice $\mathbb{B}_m$ (et non seulement d'un scalaire β_m comme dans le modèle de Potts originel).
Dans le cas du champ de Markov caché de paramètre $\mathbf{\Psi}_m$, deux difficultés apparaissent lors de la mise en œuvre de l'algorithme :

- la constante de normalisation $W(\mathbf{\Delta}_m)$ de (2)
- les probabilités conditionnelles $P(z_i|\mathbf{x}, \mathbf{\Psi}_m)$ et $P(z_i, z_j, j \in \mathcal{V}(i)|\mathbf{x}, \mathbf{\Psi}_m)$, où $\mathcal{V}(i)$ désigne l'ensemble des voisins de i

ne peuvent être calculées de manière exacte. Le principe du champ moyen consiste à se ramener à un système de particules indépendantes (sur lequel l'algorithme EM pourra être appliqué) en négligeant, pour un site i, les fluctuations de ses voisins autour de leur moyenne (*ie* en fixant $\forall j \in \mathcal{V}(i),\ z_j = \mathbb{E}(Z_j)$). Plus généralement, on parle d'approximation de type champ moyen lorsque, pour un site i, ses voisins sont fixés à des constantes. Ce champ de constantes $\tilde{z}_1, \ldots, \tilde{z}_n$ n'est pas arbitraire, il doit satisfaire certaines conditions (Celeux *et al.*, 2003). La distribution markovienne (2) peut alors être approximée par :

$$P(\mathbf{z}|\mathbf{\Delta}_m) \simeq \prod_{i=1}^{n} P(z_i|\tilde{z}_j, j \in \mathcal{V}(i), \mathbf{\Psi}_m)$$

De même, la loi jointe $P(\mathbf{x}, \mathbf{z}|\mathbf{\Psi}_m)$ et la loi markovienne $P(\mathbf{z}|\mathbf{x}, \mathbf{\Psi}_m)$ se trouvent approximées par des distributions factorisées. L'utilisation de telles approximations mène à des algorithmes itératifs à deux étapes **(a)** et **(b)** : à l'itération (q),

(a) Créer, à partir des observations $\mathbf{x}$ et de l'estimation courante des paramètres $\mathbf{\Psi}_m^{(q)}$, une configuration $\tilde{z}_1^{(q)}, \ldots \tilde{z}_n^{(q)}$, *i.e.* des valeurs pour les Z_i.

De part les approximations précédentes, les deux problèmes rencontrés lors de la mise en œuvre de EM disparaissent, et par suite,

(b) Appliquer l'algorithme EM sur ce modèle factorisé, afin d'obtenir, à partir du paramètre courant $\mathbf{\Psi}_m^{(q-1)}$, une nouvelle estimation $\mathbf{\Psi}_m^{(q)}$.

En particulier, l'algorithme en *champ moyen* consiste, à partir de la distribution conditionnelle $P(\mathbf{z}|\mathbf{x}, \mathbf{\Psi}_m^{(q-1)})$, à fixer les $\tilde{z}_i^{(q)}$ à l'estimation de leur moyenne, l'algorithme en *champ modal* à l'estimation de leur mode et l'algorithme en *champ simulé* à les simuler. En pratique, pour l'étape **(b)**, une seule itération de l'algorithme EM est suffisante. Dans ce cas, l'algorithme en champ moyen est l'algorithme *EM variationnel* de (Zhang, 1992).

L'utilisation d'un de ces algorithmes nous permet, pour chaque texture m, d'obtenir des estimateurs $(\hat{\mu}_{mk}, \hat{\Sigma}_{mk})_{1 \leq k \leq K}$ des lois gaussiennes, ainsi que des estimateurs $\hat{\mathbb{B}}_m$ et $\hat{\boldsymbol{\alpha}}_m$ des paramètres du champ caché. Cet ensemble de paramètres $\hat{\mathbf{\Psi}}_m$ va ensuite être utilisé pour classer les régions d'une image test dans une des textures apprises.

4.2 Classification d'une image test

L'objectif est de classer individuellement chacune des régions dans une des M textures. Notons encore $\mathbf{x} = (x_1, \ldots, x_n)$ les n descripteurs extraits d'une image (ou partie d'image) de texture inconnue (image test). Chaque descripteur est susceptible d'être issu d'une des M textures possibles, et donc d'une des MK sous-classes possibles. Il est alors naturel de modéliser le champ caché par un champ de Markov discret, pouvant prendre les valeurs $\{c_{mk}, m \in [1, M], k \in [1, K]\}$. Pour $i = 1, \ldots, n$, on modélise donc la probabilité d'observer le descripteur x_i par :

$$P(x_i|\mathbf{\Psi}) \;=\; \sum_{m=1}^{M} \sum_{k=1}^{K} P(Z_i = c_{mk}|\mathbf{\Delta}) \; f(x_i|\mathbf{\Theta}_{mk}), \tag{3}$$

Comme dans la section 3, la fonction d'énergie du champ spatial Z est supposée se décomposer en fonctions potentielles sur les singletons et sur les paires. Son paramètre noté $\mathbf{\Delta}$ s'écrit alors sous la forme $\mathbf{\Delta} = (\boldsymbol{\alpha}, \mathbb{B})$, où $\boldsymbol{\alpha}$ est un vecteur de dimension MK pondérant l'importance relative des différentes sous-classes, et $\mathbb{B}$ est une matrice $MK \times MK$ modélisant les interactions entre sous-classes associées à des sites voisins. Il est alors naturel de fixer $\boldsymbol{\alpha}$ égal aux potentiels $(\hat{\boldsymbol{\alpha}}_m)_{1 \leq m \leq M}$ appris pendant la phase d'apprentissage. Notons que cela revient à supposer que les différentes textures sont *a priori* équiprobables. De même, les termes de la matrice $\mathbb{B}$ correspondant aux potentiels d'interaction entre sous-classes c_{mk} et c_{ml} d'une même texture m sont fixés aux $\hat{B}_m(k, l)$ appris. Les autres termes, au contraire, concernent les interactions entre sous-classes associées à des textures différentes. Une possibilité pour les estimer serait d'effectuer l'apprentissage sur des images multi-textures, mais comme souligné dans (Lazebnik *et al.*, 2003a), l'estimation de ces termes serait mauvaise du fait du faible nombre d'arcs entre sous-classes de textures différentes. En pratique ils sont fixés à

une valeur constante qui peut varier selon le degré d'interaction que l'on souhaite (-10 pour nos expérimentations). La matrice $\hat{\mathbb{B}}$ est donc construite de la manière suivante : sur la diagonale, les matrices $\hat{\mathbb{B}}_m$, et en dehors, la constante choisie. Enfin, il est logique de fixer les $\mathbf{\Theta}_{mk}$ aux $\hat{\mathbf{\Theta}}_{mk}$ obtenus par l'apprentissage. Au final, on obtient donc une valeur du paramètre $\hat{\mathbf{\Psi}} = (\hat{\boldsymbol{\alpha}}, \hat{\mathbb{B}}, \hat{\mathbf{\Theta}})$ de la loi d'une image quelconque.
Une texture m étant composée des K sous-classes $c_{m1}, \ldots, c_{mK}$, il est naturel de classer un descripteur x_i dans la texture m maximisant $\sum_{k=1}^{K} P(Z_i = c_{mk}|x_i, \hat{\mathbf{\Psi}})$ et donc maximisant $\sum_{k=1}^{K} P(Z_i = c_{mk}|\hat{\mathbf{\Delta}})\ f(x_i|\hat{\mathbf{\Theta}}_{mk})$. Cependant, la loi markovienne de Z_i fait intervenir la classification inconnue z_j des sites j voisins de i, et n'est donc pas calculable directement. On peut cependant réappliquer le principe du champ moyen sur les paramètres $\hat{\mathbf{\Psi}}$ laissés fixes. Pour toute sous-classe c, notons $t_{i,c}^{(q-1)}$ une approximation à l'itération $(q-1)$ de $t_{i,c} \equiv P(Z_i = c|x_i, \hat{\mathbf{\Psi}})$. A l'itération (q) pour le site i, un nouveau champ de voisins $\tilde{z}_1^{(q)}, \ldots, \tilde{z}_n^{(q)}$ est créé à l' étape **(a)** à partir des paramètres $\hat{\mathbf{\Psi}}$ et des $t_{i,c}^{(q-1)}$. Le principe du champ moyen conduit à réestimer à l'étape **(b)** les $t_{i,c}^{(q)}$ par :

$$t_{i,c}^{(q)} \propto P(Z_i = c|(\tilde{z}_j^{(q)})_{j\in\mathcal{V}(i)}, \hat{\mathbf{\Delta}})\ f(x_i|\hat{\mathbf{\Theta}}_c)$$

c'est à dire par :

$$t_{i,c}^{(q)} \propto \exp(\hat{\boldsymbol{\alpha}}_c + \sum_{j\in\mathcal{V}(i)} B(c, \tilde{z}_j^{(q)}))\ f(x_i|\hat{\mathbf{\Theta}}_c).$$

Pour rendre la comparaison avec d'autres algorithmes (voir Section 5) plus claire, on utilisera également la notation $\tilde{z}_{j,c'}^{(q)}$ qui vaut 1 si $\tilde{z}_j$ appartient à la sous-classe c', et 0 sinon. La formule précédente s'écrit alors :

$$t_{i,c}^{(q)} \propto \exp(\hat{\boldsymbol{\alpha}}_c)\ f(x_i|\hat{\mathbf{\Theta}}_c)\ \exp(\sum_{j\in\mathcal{V}(i)} \sum_{c'} B(c, c')\tilde{z}_{j,c'}^{(q)}). \quad (4)$$

5 Autres algorithmes de classification de textures

Pour comparaison, nous considérons également d'autres algorithmes pour l'apprentissage de textures : la relaxation (Rosenfeld *et al.*, 1976), et l'algorithme Neighborhood EM (Ambroise *et al.*, 1997).

5.1 Relaxation

Il s'agit d'une méthode courante en vision, notamment utilisée dans (Lazebnik *et al.*, 2003a), pour ajouter simplement du spatial, par un terme de poids. Le principe consiste à raffiner les probabilités $t_{i,c} = P(Z_i = c|x_i, \mathbf{\Psi})$ utilisées dans la règle de classification. A l'itération (q), les nouvelles probabilités $t_{i,c}^{(q)}$ sont réestimées à partir des $t_{i,c}^{(q-1)}$ par :

$$t_{i,c}^{(q)} \propto t_{i,c}^{(q-1)}(1 + \sum_{j\in\mathcal{V}(i)} \sum_{c'} B(c, c')\ t_{j,c'}^{(q-1)}) \quad (5)$$

où le terme $B(c, c')$ exprime à quel point les sous-classes c et c' sont compatibles. Dans (Lazebnik *et al.*, 2003a), ce terme est calculé comme une co-occurence pour deux sous-classes c et c' associées à une même texture, et fixé à une constante négative sinon. Notons que la matrice $B(c, c')$ ainsi définie joue un rôle comparable à la matrice $\mathbb{B}$ du champ de Markov : elle traduit la force d'interaction entre les différentes sous-classes.
Dans (Lazebnik *et al.*, 2003a), la méthode de classification de textures est la suivante : l'algorithme est initialisé par

$$t_{i,c}^{(0)} = P(Z_i = c | x_i, \mathbf{\Psi}^{\text{EM}})$$

où $\mathbf{\Psi}^{\text{EM}}$ désigne une estimateur de $\mathbf{\Psi}$ obtenu par application de l'algorithme EM pour chacune des textures (ce qui suppose que les descripteurs sont indépendants). Notons que dans ce cas, $\mathbf{\Psi}$ correpond aux proportions $\{p_c\} = \{p_{mk}, m \in [1, M], k \in [1, K]\}$ du mélange, ainsi qu'aux paramètres $\{\mathbf{\Theta}_c\} = \{\mathbf{\Theta}_{mk}, m \in [1, M], k \in [1, K]\}$ des gaussiennes. L'algorithme de relaxation ne convergeant pas, un certain nombre d'itérations sont effectuées (N=200 en pratique). Un descripteur x_i est enfin classé en fonction des $t_{i,c}^{(N)}$, selon la même règle que dans la Section 4.2.
Notons qu'une classification de base obtenue à l'aide des $t_{i,c}^{(0)}$ consiste à classer un descripteur x_i dans la texture de plus grande vraisemblance relativement à x_i pour un modèle de mélange indépendant (c'est la méthode que nous appellerons "Maximum de vraisemblance" dans la Section 6).

5.2 Algorithme Neighborhood EM

L'algorithme *Neighborhood EM* ou NEM (Ambroise *et al.*, 1997) est une pénalisation de l'algorithme EM, permettant d'ajouter de l'information spatiale pour des données modélisées par un mélange indépendant. S'inspirant de la Section 4, une méthode de classification de textures basées sur cet algorithme est la suivante : les paramètres $\{p_c^{\text{NEM}}\}$ (les proportions du mélange) et $\{\mathbf{\Theta}_c^{\text{NEM}}\}$ (les paramètres des gaussiennes) des différentes textures sont appris en appliquant NEM. Dans la phase de test, comme souligné en Section 4.2, la classification pourrait consiter à classer un descripteur dans la texture m maximisant $\sum_k P(Z_i = c_{mk} | x_i, \hat{\mathbf{\Psi}})$, mais en pratique les résultats sont grandement améliorés en réappliquant NEM au mélange avec paramètres fixés $\hat{\mathbf{\Psi}}$. Dans ce cas, les probabilités $t_{i,c}$ sont réestimées à l'itération (q) par :

$$t_{i,c}^{(q)} \propto p_c^{\text{NEM}} \; f(x_i | \mathbf{\Theta}_c^{\text{NEM}}) \; \exp(\sum_{j \in \mathcal{V}(i)} \sum_{c'} B(c, c') \; t_{j,c'}^{(q-1)}) \qquad (6)$$

où le terme $B(c, c')$ a la même fonction et est estimé de la même manière que pour la relaxation.
Remarquons que dans l'algorithme NEM, à chaque itération, un modèle de mélange est sous-jacent, ce qui se note dans la formule (6) par la présence du terme $p_c^{\text{NEM}} \; f(x_i | \mathbf{\Theta}_c^{\text{NEM}})$. Dans la relaxation au contraire, la formule (5) nous montre qu'aucun modèle sous-jacent n'est supposé. Dans la formule (4) des algorithmes de type champ moyen, comme souligné en Section 3, le terme $\exp(\hat{\boldsymbol{\alpha}_c})$ peut être comparé aux proportions p_c^{NEM} du mélange. En fait, dans le cas particulier où les paramètres sont fixés, l'algorithme en

champ moyen est équivalent à l'algorithme NEM. Pour les algorithmes en champs modal et simulé, le terme dans l'exponentielle diffère : $t_{j,c'}^{(q-1)}$ est un réel estimant $P(Z_i = c'|x_i, \hat{\mathbf{\Psi}})$, alors que $\tilde{z}_{j,c'}^{(q)}$ prend la valeur 0 ou 1 selon que Z_j ait été affecté à la classe c' ou simulé dans c' par la loi $P(.|x_i, (\tilde{z}_j)_{j \in \mathcal{V}(i)}, \hat{\mathbf{\Psi}})$.

6 Résultats expérimentaux

Les expériences sont effectuées sur des images de 7 textures différentes dont un échantillon se trouve dans la Figure 2.

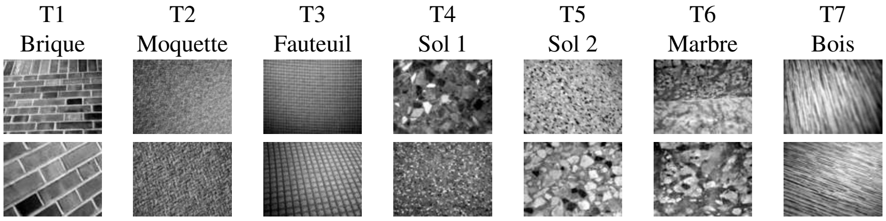

FIG. 2 – Echantillon des 7 textures utilisées pour les expériences.

La base d'apprentissage est composée de 10 images uni-texture pour chacune des 7 textures (soit un total de 70 images). Pour simplifier, nous supposons que $K = 10$ pour chacune des textures (en fait, la selection de K par l'application du Critère d'Information Bayésien (BIC) de Schwartz (Celeux *et al.*, 2003) ne semble pas apporter d'amélioration significative). De même, nous nous sommes restreints à des matrices de covariances diagonales. Parmi ces modèles, pour chaque texture m, BIC sélectionne celles de la forme $\Sigma_{mk} = \sigma_m^2 \ I$ pour tout $k = 1 \ldots K$ (encore une fois, un modèle de la forme $\Sigma_{mk} = \sigma_{mk}^2 \ I$ n'améliore que très peu les résultats, tout en augmentant très significativement les temps de calcul).

6.1 Reconnaissance d'images uni-texture

Nous testons notre méthode sur une base de test comprenant 10 images de chaque texture non utilisées dans la phase d'apprentissage. Le Tableau 1 donne les taux de bonne classification de l'ensemble des régions, c'est à dire, pour chaque texture m, le pourcentage de descripteurs de la texture m effectivement classés dans la texture m. Les différentes lignes se réfèrent aux algorithmes vus précédemment (les résultats correspondant aux champs moyen et modal ne sont pas reportés, car au mieux équivalents à ceux du champ simulé).

Les résultats de la Table 1 confirment l'intérêt de prendre en compte l'organisation spatiale des descripteurs : la classification par maximum de vraisemblance donne des taux de classification significativement inférieure pour les textures 1 à 5 (de -20 à -34%). Pour les textures 6 et 7, l'amélioration est moins notable. De manière générale, toutes les méthodes semblent avoir du mal à apprendre les textures 6 et 7. Ceci s'explique par

Texture	T1	T2	T3	T4	T5	T6	T7
Max. vraisemblance	48	77	52	56	51	17	30
Relaxation	78	96	72	86	80	19	42
NEM	82	98	78	88	80	20	43
Champ simulé	81	97	77	80	86	26	46

TAB. 1 – % de bonne classification des régions sur des images uni-textures.

le fait que ces deux textures contiennent des images avec de très brusques changements de luminosité, rendant l'apprentissage d'autant plus difficile.
NEM et l'algorithme en champ simulé améliorent les taux de classification pour chacune des textures par rapport à la relaxation. En effet, dans l'algorithme de relaxation, aucun modèle n'est supposé et les itérations sont indépendantes du modèle utilisé pour les données. Au contraire, NEM est originellement fait pour un modèle de mélange, si bien que ce modèle de mélange est pris en compte à chaque itération. La méthode utilisant les champs de Markov est quant à elle la seule où les descripteurs sont modélisés comme des variables statistiques dépendantes.
Le taux de bonne classification de NEM est plus élevé que celui du champ simulé pour les textures 1 à 4, mais plus faible pour les textures 5 à 7. En moyenne, c'est l'algorithme en champ simulé qui apparait le plus performant (70.43% de reconnaissance pour l'algorithme en champ simulé contre 69.86% pour NEM).
Ces premières expériences montrent qu'il y a un apport significatif à ajouter de l'information spatiale entre les descripteurs. Il apparait en outre qu'il est plus judicieux d'utiliser un modèle paramétrique, comme le modèle de mélange (NEM) ou son extension aux champs de Markov (algorithme de type champ moyen), pour l'apprentissage comme pour le test.
Notons toutefois les bonnes performances de l'algorithme de relaxation malgré l'absence des données dans l'itération (5). Ces dernières ne sont prises en compte que dans l'initialisation, ce qui correspond à un principe de maximum a priori et non a posteriori comme le recommande la théorie statistique. Une explication de ces bonnes performances est sans doute la robustesse d'un tel algorithme vis à vis de la mauvaise adéquation du modèle de mélange aux données. Il n'existe pas à ce jour de test statistique pour tester l'hypothèse de mélange mais une voie d'approche est envisageable, basée sur un prétraitement des données visant à les rendre plus proches du modèle choisi.

6.2 Sur des images multi-textures

Les différents algorithmes sont également testés sur 62 images multi-textures, dont 5 artificiellement créées. Des exemples de classifications obtenues sont présentés dans les Figures 3 à 8.
La Figure 3 est un exemple de segmentation obtenue par les 4 algorithmes étudiés précédemment : le maximum de vraisemblance, la relaxation, NEM et le champ simulé.

Il en ressort clairement que la prise en compte de l'information spatiale (par la re-

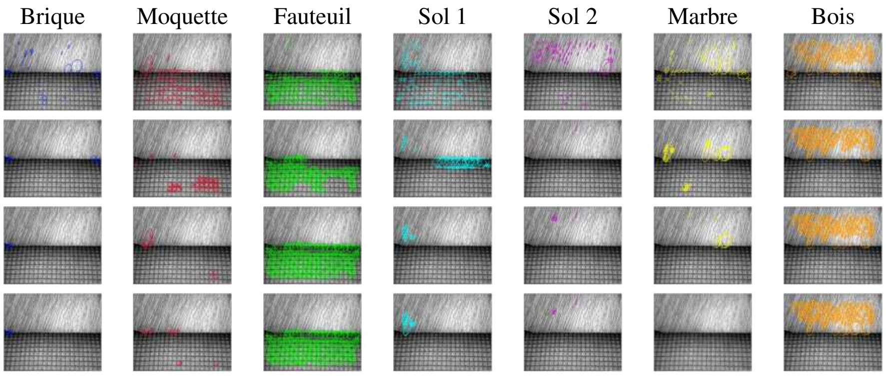

FIG. 3 – De haut en bas : classification obtenue par le maximum de vraisemblance, la relaxation, NEM et l’algorithme en champ simulé, sur une image composée de fauteuil et de bois

laxation, NEM, ou le champ simulé), améliore la classification. De plus on observe que, parmi ces algorithmes spatiaux, c’est celui utilisant une modélisation explicite des dépendances (champ de Markov caché estimé par l’algorithme en champ simulé) qui donne les meilleurs résultats. En outre, on observe que la moquette est très bien reconnue, alors que quelques erreurs subsistent avec le bois. Ces résultats vont dans le sens de ceux obtenus sur des images uni-textures, le bois ayant été mal appris.

Un autre exemple illustrant l’intérêt de la prise en compte de l’organisation spatiale des descripteurs est donné Figure 4 : l’algorithme en champ simulé permet une bien meilleure segmentation que le maximum de vraisemblance.

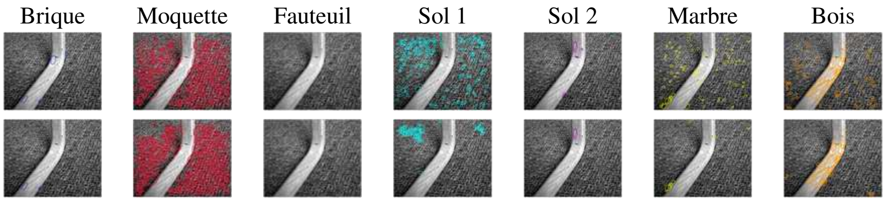

FIG. 4 – Classification obtenue par le maximum de vraisemblance (en haut) et l’algorithme en champ simulé (en bas), sur une image composée de moquette et de bois

Les Figures 5 à 8 sont des exemples de classifications obtenues par l’algorithme en champ simulé. On notera (Figure 5) les performances de cet algorithme spatial sur une image artificielle, constituée de 4 bouts d’images.

En Figure 6, on observe clairement le comportement de l’algorithme en champ simulé : le marbre est mal reconnu (ce qui était un résultat prévisible au vu du Tableau 1), mais c’est surtout avec la brique et le bois qu’il est confondu. En effet, cet algorithme

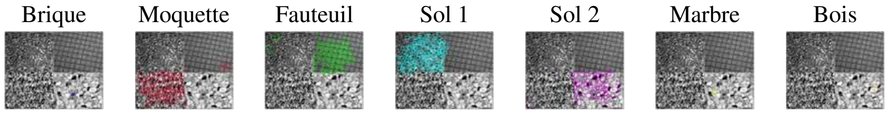

FIG. 5 – Classification obtenue par l'algorithme en champ simulé, sur une image artificielle composée de sol 1, de fauteuil, de moquette et de sol 2.

spatial tend à regrouper les sites voisins dans un même classe, et va donc, de proche en proche, classer une partie des sites correspondant au marbre dans l'une des deux textures présentes aux alentours, la brique et le bois.

FIG. 6 – Classification obtenue par l'algorithme en champ simulé, sur une image composée de briques, de marbre et de bois.

Enfin, les Figures 7 et 8 mettent en relief les problèmes de classification lorsque la qualité de l'image diminue. En effet, en Figure 7, l'arrière plan (le bois) est flou, et les résultats obtenus sont médiocres (noter que sur l'image nette de la Figure 3, ils sont bien meilleurs). En Figure 8, une partie de l'image (en l'occurence les briques) est mal éclairée, si bien que les sites correspondants sont relativement mal classés, alors que la texture brique a été bien apprise (81% de bonne classification sur les images uni-textures). Plus qu'une limitation de nos algorithmes, c'est plutôt la qualité du descripteur que doit être mis en défaut. La possibilité d'utiliser d'autres descripteurs plus invariants au problèmes de luminosité est à envisager.
De manière générale, notons que la plupart de erreurs ont lieu près des bords, ou aux frontières entre textures différentes. Ceci suggère que le graphe de voisinage pourrait être repensé pour limiter ce phénomène.

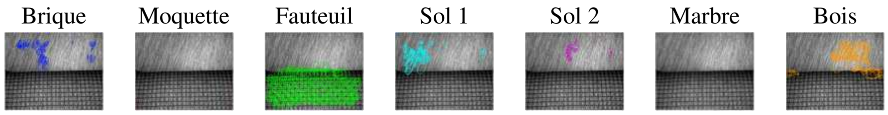

FIG. 7 – Classification obtenue par l'algorithme en champ simulé, sur une image de qualité moindre composée de fauteuil et de bois (partie bois floue).

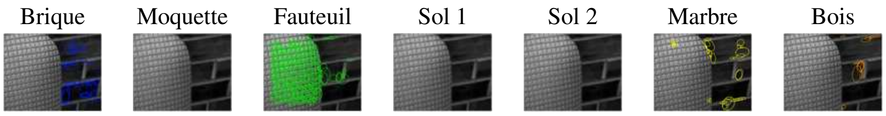

FIG. 8 – Classification obtenue par l'algorithme en champ simulé, sur une image avec mauvais éclairage composée de fauteuil et de briques.

7 Conclusion et perspectives

Notre travail s'est basé sur des techniques récentes de description d'image à l'aide de descripteurs calculés en un certain nombre de points caractéristiques de l'image. Notre objectif était de montrer que des modèles statistiques paramétriques pouvaient être introduits pour rendre compte de l'organisation spatiale et géométrique de ces descripteurs. Les champs de Markov cachés étaient des candidats naturels, que nous avons expérimentés dans le cadre de la reconnaissance de textures. L'utilisation de modèles markoviens pour la ségmentation d'images caractérisées par une grille régulière de pixels est standard, mais leur introduction en reconnaissance pour modéliser des vecteurs caractéristiques irrégulièrement espacés est nouvelle. Dans ce contexte, les paramètres de ces modèles ont une interprétation naturelle : certains (les α_{mk}) peuvent être assimilés à des proportions sur les textures, alors que d'autres (la matrice $\mathbb{B}$) à des intéractions spatiales. Dans notre méthode, ces paramètres sont estimés, ou bien fixés pour incorporer de la connaissance *a priori* sur les textures.
Des résultats obtenus sur des images uni ou multi textures sont prometteurs. Ils mettent notamment à jour l'intérêt de prendre en compte l'organisation spatiale des descripteurs. De plus, l'utilisation d'un modèle rendant explicitement compte de ces dépendances (champ markovien) améliore les taux de reconnaissance. Notons que les résultats pourraient sans doute encore être améliorés en effectuant sur les données des transformations préliminaires visant à les rendre plus proche du modèle utilisé (un mélange gaussien). Plus précisément, on pourra envisager de coupler des transformations de type Box-Cox avec des tests de gaussianité en grande dimension.
Par ailleurs, le formalisme général présenté dans ce papier pourrait être envisagé dans d'autres contextes, en reconnaissance d'objets par exemple. Mais avant cela, une étude plus spécifique du choix de la structure de voisinage serait nécessaire, en particulier concernant la définition d'un graphe de voisinage préservant au maximum l'invariance affine. Enfin, la méthodologie pourrait être testée en utilisant d'autres techniques de description d'images.

Références

AMBROISE C., DANG V. M. & GOVAERT G. (1997). Clustering of spatial data by the EM algorithm. In K. A. P. DORDRENCHT, Ed., *geoENV I- Geostatistics for Environmental Applications,Quantitative Geology and Geostatistics*, volume 9.

BRADSHAW B., SCHOLKOPF B. & PLATT J. (2001). Kernel methods for extracting local image semantics. In *Microsoft Research Technical Report, MSR-TR-2001-99*.

CELEUX G., FORBES F. & PEYRARD N. (2003). EM procedures using mean field-like approximations for Markov model-based image segmentation. In *Pattern Recognition*, volume 36(1), p. 131–144.

CHANDLER D. (1987). Introduction to modern statistical mechanics.

DEMPSTER A., LAIRD N. & RUBIN D. (1977). Maximum Likelihood from incomplete data via the EM algorithm.

KUMAR S. & HEBERT M. (2003). Man-made structure detection in natural images using a causal multiscale random field. In *Proc. CVPR*.

LAZEBNIK S., SCHMID C. & PONCE J. (2003a). Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proc. ICCV*.

LAZEBNIK S., SCHMID C. & PONCE J. (2003b). Sparse texture representation using affine-invariant regions. In *Proc. CVPR*.

LINDEBERG T. & GARDING J. (1997). Shape-adapted smoothing in estimation of 3-d depth cues from affine distorsions of local 2-d brightness structure. In *Image and Vision Computing*, volume 15.

MALIK J., BELONGIE S., LEUNG T. & SHI J. (2001). Contour and texture analysis for image segmentation. In *IJCV*, volume 43(1).

NIGAM K., MCCALLUM A., THRUN S. & MITCHELL T. (2000). Text classification from labeled and unlabeled documents using EM. In *Machine Learning*, volume 39 (2/3).

ROSENFELD A., HUMMEL R. & ZUCKER S. (1976). Scene labeling by relaxation operations. In *IEEE Trans. Systems, Man, and Cybernetics*, volume 6(6).

SCHMID C. (2001). Constructing models for content-based image retrieval. In *Proc. CVPR*.

ZHANG J. (1992). The Mean Field Theory in EM Procedures for Markov Random Fields. In *IEEE trans. Signal Proc.*, volume 40(10).