Markov Random Fields for Recognizing Textures modeled by Feature Vectors

Florence Forbes and Juliette Blanchet

Inria Rhône-Alpes, 655 avenue de l'Europe,
Montbonnot, 38334 Saint Ismier Cedex, France
(e-mail: florence.forbes@inrialpes.fr, juliette.blanchet@inrialpes.fr)

Abstract. This paper decribes a new probabilistic framework for recognizing textures in images. Images are described by local affine-invariant descriptors and by spatial relationships between these descriptors. We propose to introduce the use of statistical parametric models of the dependence between descriptors. Hidden Markov Models (HMM) are investigated for such a task using recent estimation procedures based on the mean field principle to perform the non trivial parameter estimation they require. Preliminary experiments obtained with 140 images of seven different natural textures show promising results.
Keywords: Hidden Markov Models, Mean Field approximation, Statistical learning, Texture recognition.

1 Introduction

Image descriptors is a key notion in computer vision. Descriptors are local characteristics whose geometric organization can be very informative when carrying out pattern recognition tasks. The most important characteristics for efficient image descriptors are good discrimination, locality (for resistance to occlusions), and sufficient invariance to various image transformations. Local descriptors that meet these requirements exist, but incorporating information about the relative spatial organization of such descriptors is still an open issue. It is not yet clear which organizational models will prove to be the most useful, and many statistical issues relating to the estimation and selection of such models remain to be resolved. In this paper, we propose organizational models based on Markov Random Fields and we focus on a texture recognition task as a first investigation of these models. We show that recognition can be improved by using Hidden Markov Models (HMM) as organizational models when learning the texture classes. Estimating the parameters of such models in this context is not trivial. We use recent estimation procedures (EM-like algorithms) based on the Expectation-Maximization (EM) algorithm and on the mean field principle of statistical physics [Chandler, 1987].

2 Hidden Markov Models for textures

For the feature extraction stage, we follow the texture representation method described in [Lazebnik *et al.*, 2003a] for its advantages over methods proposed in recent literature. It is based on an interest point detector that leads to a sparse representation selecting the most perceptually salient regions in an image and on a *shape selection* process that provides affine invariance. Informally (see [Lindeberg and Garding, 1997] for details), regions are represented by ellipses of various volume and shape and centered at various locations (points found by the detector). The neighborhood of a region represented by a given ellipse can then be naturally computed by adding a constant amount (15 pixels in our implementation) to the major and minor axes and to let the neighborhood consists of all points that fall inside this enlarged ellipse. We can then think of an image as a graph with edges emanating from the center of each region to other centers within its neighborhood. To each detected region is then associated a feature vector (descriptor). The descriptors we use are intensity domain *spin images* [Lazebnik *et al.*, 2003b] rescaled to have a constant norm and flattened into 80-dimensional feature vectors. The basic assumption is that descriptors are random variables with a specific probability distribution in each texture class. In [Lazebnik *et al.*, 2003a], the distribution of descriptors in each texture class is modeled as a Gaussian mixture model where each component corresponds to a sub-class. This is assuming that the descriptors are independent variables although it naturally exists strong neighborhood relationships between feature vectors within the same image. To take that into account, we propose to improve on the Gaussian mixture model by assuming that for each image from a single texture, the distribution of descriptors is that of a Hidden Markov Model (HMM) with K components and appropriate parametrization to be specified below.

Let $x_1, \ldots, x_n$ denote the n descriptors (80-dimensional vectors) extracted at locations denoted by $\{1, \ldots, n\}$ from an image. Let m denotes the texture class of this image. For $i = 1, \ldots, n$, we model the probability of observing descriptor x_i when the image is from texture m as

$$P(x_i|\Psi_m) \;=\; \sum_{k=1}^{K} P(Z_i = c_{mk}|\beta_m)\; f(x_i|\theta_{mk}),$$

where $f(x_i|\theta_{mk})$ denotes the multivariate Gaussian distribution with parameters θ_{mk} namely the mean μ_{mk} and covariance matrix Σ_{mk}. Notation Z_i denotes the random variable representing the sub-class of descriptor x_i. It can take values in $\{c_{mk}, k = 1 \ldots K\}$ denoting the K possible sub-classes for texture m. Note that for simplicity we assume K being the same for each texture but this can be generalized (see section 5). Notation β_m denotes additional parameters defining the distribution of the Z_i's and Ψ_m denotes the whole model parameters *i.e.* $\Psi_m = (\theta_{mk}, \beta_m, k = 1 \ldots K)$. Our approach differs from [Lazebnik *et al.*, 2003a] in that our aim is to account for spatially dependent descriptors. More specifically, the dependencies between neighboring descriptors are modeled by further assuming that the joint distribution

of $Z_1, \ldots, Z_n$ is a discrete Markov Random Field on the graph defined above. Denoting $z = (z_1, \ldots, z_n)$ specified values of the Z_i's, we define

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

where $W(\beta_m)$ is a normalizing constant and H is a function assumed to be of the following form (we restrict to pair-wise interactions),

$$H(z, \beta_m) = \sum_{i=1}^{n} V_i(z_i, \beta_m) + \sum_{\substack{i,j \\ i \sim j}} V_{ij}(z_i, z_j, \beta_m),$$

where the V_i's and V_{ij}'s are respectively referred to as singleton and pair-wise potentials. We write $i \sim j$ when locations i and j are neighbors on the graph, so that the second sum above is over neighboring locations. The spatial parameters β_m consist of two sets $\beta_m = (\alpha_m, \mathbb{B}_m)$ where α_m and $\mathbb{B}_m$ are defined as follows. We consider pair-wise potentials V_{ij} that only depend on z_i and z_j (not on i and j). Since the z_i's can only take a finite number of values, we can define a $K \times K$ matrix $\mathbb{B}_m = (b_m(k,l))_{1 \leq k,l \leq K}$ and write without lost of generality

$$V_{ij}(z_i, z_j, \beta_m) = -b_m(k,l) \text{ if } z_i = c_{mk} \text{ and } z_j = c_{ml}.$$

Similarly we consider singleton potentials V_i that only depend on z_i so that denoting by α_m a $K-$dimensional vector, we can write

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

where $\alpha_m(k)$ is the k^{th} component of α_m. This vector α_m acts as weights for the different values of z_i. When α_m is zero, no sub-class is favored, *i.e.* at a given location i, if no information on the neighboring locations is available, then all sub-classes appear with the same probability at location i. When $\mathbb{B}_m$ is zero, there is no interaction between the locations and the Z_i's are independent. When $\mathbb{B}_m$ is zero, β_m reduces to α_m and it comes that for $i = 1, \ldots, n$ and $k = 1, \ldots, K$,

$$P(Z_i = c_{mk}|\alpha_m) = \frac{\exp(\alpha_m(k))}{\sum_{l=1}^{K} \exp(\alpha_m(l))},$$

which clearly shows that α_m acts as weights for the different possible values of z_i. Conversely, when α_m is zero and $\mathbb{B}_m = \beta \times I$ where β is a scalar, the spatial parameters β_m reduce to a single scalar interaction parameter β and we get the Potts model traditionnaly used for image segmentation. Note that this model is not necessarily appropriate for textures since it tends to favor neighbors that are in the same sub-class. In practice we observed in our experiments that when learning texture classes, $\mathbb{B}_m$ could be far from $\beta \times I$. Texture m is then represented by an HMM defined by parameters Ψ_m being $\Psi_m = (\mu_{mk}, \Sigma_{mk}, \alpha_m(k), \mathbb{B}_m, k = 1, \ldots, K)$.

3 Learning the descriptors distribution and organization

In a supervised framework, we first learn the distribution for each texture class based on a training data set. Our learning step is based on an EM-

like algorithm and this framework allows to incorporate unsegmented multi-texture images. However, we refer to the work of [Nigam *et al.*, 2000] and [Lazebnik *et al.*, 2003a] for more details on how to implement this generalization. In this presentation the training data consists then of single-texture images from each texture class $m = 1, \ldots, M$. Using all the feature vectors and neighborhood relationships extracted from the images belonging to class m, we estimate an HMM as described in section 2. The EM algorithm is a commonly used algorithm for parameters estimation in problems with hidden data (here the sub-class assignments). For Hidden Markov Random Fields, due to the dependence structure, the exact EM is not tractable and approximations are required to make the algorithm tractable. In this paper, we use some of the approximations based on the mean field principle presented in [Celeux *et al.*, 2003]. This allows to take the Markovian structure into account while preserving the good features of EM. The procedures in [Celeux *et al.*, 2003] are based on mean field approximation. More specifically, we used the so-called *simulated field* algorithm for it shows better performance in some segmentation tasks (see [Celeux *et al.*, 2003]). Note that in practice, we had to extend these algorithms to incorporate the estimation of matrix $\mathbb{B}_m$ and to include irregular neighborhood structure coming from descriptors locations and not from regular pixel grids like in [Celeux *et al.*, 2003]. For comparison we also consider a different way to learn texture that do not use the HMM formalism. We used a penalized EM algorithm for spatial data called NEM for Neighborhood EM [Ambroise *et al.*, 1997]. It provides a way to add spatial information when dealing with data represented as independent mixture models. It leads to a simple procedure but is not as flexible as the HMM approach which includes spatial information directly in the model. NEM can be seen as intermediate between the use of independent mixture models as in [Lazebnik *et al.*, 2003a] and our approach. To use it in our experiments we had to generalize its Potts-like penalization to a penalization term appropriate for textures. We used a matrix $\mathbb{B}$ as in Section 2.

A set of parameters is then associated to each texture class and used to classify regions in test images in one of the learned textures as specified in the next section.

4 Classification and retrieval

Images in the test set are not labeled and may contain several texture classes. Our aim is first to classify each region individually in one of the M texture classes under consideration. Then, each region can possibly be in one of $M \times K$ sub-classes. To identify these sub-classes, the model for the descriptor distribution has to incorporate the information learned from each texture in the learning stage. To do so, at recognition time, the descriptors distribution is assumed to be that of a Gaussian HMM as presented in Section 2 but with

a discrete hidden field taking values in $\{c_{mk}, m = 1, \ldots, M, k = 1, \ldots, K\}$ *i.e.* with $M \times K$ components instead of K in the learning stage. In addition, the parameters of this HMM are given: for $m = 1, \ldots, M$ and $k = 1, \ldots, K$, the conditional distributions $f(x_i|\theta_{mk})$ are assumed to be Gaussian with means and covariance matrices learned at learning time. As regards, the hidden field, the pair-wise potentials are defined through a square matrix of size $M \times K$ denoted by $\mathbb{B}$ and constructed from the learned $\mathbb{B}_m$ matrices as follows: we first construct a bloc diagonal matrix using the learned $\mathbb{B}_m$ as blocs. The other terms correspond to pairs of sub-classes belonging to different classes. When only single-texture images are used in the learning stage, these terms are not available. As mentionned in [Lazebnik *et al.*, 2003a] even when multi-texture images are used for learning, the estimations for such terms are not reliable due to the fact that only a few such pairs are present in the training data. Unless the number of texture classes is very small, it is quite difficult to create a training set that would include samples of every possible boundary. In practice the missing values in $\mathbb{B}$ are set to a constant value chosen as a "smootheness constraint". The potentials on singletons, which are related to the proportions of the different sub-classes as mentioned in Section 2 are fixed to the values learned for each texture. Then the EM-like algorithm of Section 3 can be used with all parameters fixed to estimate the membership probability for each of the $M \times K$ sub-classes. The algorithm can be seen as iterations refining initial membership probabilities by taking into account the learned HMM's. This is not possible with standard EM for Gaussian mixtures since without spatial information, when all parameters are fixed, the algorithm reduces to a single iteration.

Membership probabilities are then also obtained for each texture class. For each region located at i, we get $P(Z_i = c_{mk}|x_i)$ for $m = 1, \ldots M$ and $k = 1, \ldots K$ and $P(Y_i = m|x_i)$ if Y_i denotes the unknown texture class. We have $P(Y_i = m|x_i) = \sum_{k=1}^{K} P(Z_i = c_{mk}|x_i)$. Determining the texture class of the region located at i consists then in assigning it to the class m that maximizes $P(Y_i = m|x_i)$. At the image level, a global score can be defined for each texture class. For instance, the score for class m can be computed by summing over all n regions found in the image, *i.e.* $\sum_{i=1}^{n} P(Y_i = m|x_i)$, and the image assigned to the class with the highest score.

Note that in a previous study, the HMM in the test stage was only partly defined. All parameters were fixed as above except the potentials on singletons which were estimated using the EM-like algorithm as in Section 3. This required much more computation and did not lead to better recognition rates in our experiments, except for some rare cases. However this possibility would worth further investigation.

5 Experimental Results

Preliminary experiments are made on a data set containing seven different textures (Figure 1). The data set is partitionned into a training and a test set containing 10 single texture images each. For simplicity, we set $K = 10$ for each texture. In some preliminary study we selected varying K using the Bayesian Information Criterion (BIC) of Schwarz [Forbes and Peyrard, 2003] but we did not observe significantly better recognition results. For the Gaussian distributions we restrict to diagonal covariance models. For each texture class m, using BIC we select among these models, the ones with $\Sigma_{mk} = \sigma_m^2 I$ for all $k = 1, \ldots K$. Table 1 shows classification results for individual regions that is the fraction of all individual regions in the test images that were correctly classified. The "Max likelihood" column refers to the method that consists in assuming that all texture class has the same probability to occur in the test image independently of the image. A region is then classified as belonging to the texture class with the best mixture likelihood (learned parameters). The "Relaxation" column refers to the method used in [Lazebnik *et al.*, 2003a]. The procedure uses as initial probabilities the ones that can be computed from the learned mixture models. These probabilities are then modified, through a relaxation step [Rosenfel *et al.*, 1976], using some additional spatial information deduced from the learning stage using co-occurence statistics. The results in Table 1 show that the rates improve significantly on the Maximum Likelihood rates for textures 1 to 5 but much less for textures 6 and 7. This points out one drawback of Relaxation which is sensitive to the quality of the initial probability estimates. The following columns refer to methods investigated in this paper. When all parameters are fixed, as this is the case in the test stage, NEM iterations can be reduced to update equations for the membership probabilities. These equations can be compared to Relaxation equations which similarly consist in updating membership probabilities. However, a main difference is that NEM is originally made for mixture models and therefore the mixture model is taken into account at each iteration. In the Relaxation algorithm, no model assumption is made and iterations are independent of the model used for the data. In a context where learning is made by assuming mixture models, using NEM seems then more consistent and appropriate. Table 1 shows better rates for NEM when compared to Relaxation. The method using HMM's is the only one where the descriptors are modeled as statistically dependent variables. It provides a way to analyse and control theses dependencies through a number of parameters. The "simulated Field" columns refer to our HMM model. When all parameters are fixed, the Simulated Field algorithm also reduces to update equations comparable to Relaxation but with the advantage of including the Markov model explicitly. The rates increase when compared to Relaxation. When comparing to NEM, rates increase for textures 5 to 7 and decrease for textures 1 to 4 but on average the Simulated Field algorithm performs better. As a global comment, one can observe that all methods have

more trouble in recognizing textures 6 and 7. The corresponding data sets both contain images with very strong luminosity changes and some fuzzy images suggesting that the descriptors and/or the neighborhood structure we used may not be invariant enough. These preliminary experiments show however that there is significant gain in incorporating spatial relationships between descriptors. It appears that there is some gain in doing that using statistical parametric models, such as mixture models (NEM) or their extension HMM's (Simulated Field Algorithm), in the learning stage as well as in the test stage.

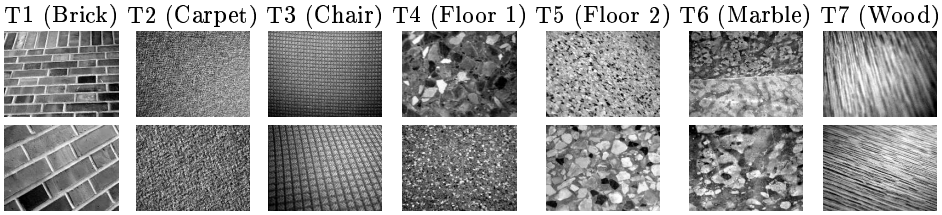

Fig. 1. Samples of the texture classes used in the experiments.

Class	T1	T2	T3	T4	T5	T6	T7
Max. Likelihood	48	77	52	56	50	17	30
Relaxation	78	96	72	86	80	19	42
NEM	82	98	78	88	80	20	43
Simulated Field	81	97	77	80	86	26	46

Table 1. Classification rates in % for individual regions of single-texture images.

6 Conclusions

We based our work on recent techniques for image description going further regular grid of pixels to sets of irregularly spaced feature vectors. Our aim was to show that statistical parametric models could be introduced to account for spatial or geometric relationships between feature vectors. We show that Hidden Markov Models were natural candidates and focused on a texture recognition task as an illustration. For such a task Markov Models have been used to model grey-level values on regular pixel grids but their introduction in the context of feature vectors at irregular locations is new. In this context, they provide parametric models where the parameters have a natural interpretation. Some of them (the α_{mk}'s) can be related to texture

proportions while others (matrix $\mathbb{B}$) to pair-wise interactions (see Section 2). In our method, parameters can be estimated or tuned, for instance, to incorporate a priori knowledge regarding texture proportions or strenght of interactions. Other methods such as Relaxation are much less readable in that sense.

Preliminary results are promising and illustrate a general methodology. It provides a statistical formalism to be investigated in other contexts. Future work would be to study its application for object recognition or more complex classes recognition. Before that, more specific analysis would be necessary as regards the choice of the neighborhood structure. In particular, the use of stronger geometric neighborhood relationships that take into account affine shape while preserving the maximum amount of invariance would worth additional investigation. Also the methodology presented here for feature vectors derived from interest points and spin images, could be investigated with other image description techniques.

References

[Ambroise *et al.*, 1997]C. Ambroise, V. Mo Dang, and G. Govaert. Clustering of spatial data by the EM algorithm. In Kluwer Academic Publishers Dordrencht, editor, *geoENV I- Geostatistics for Environmental Applications,Quantitative Geology and Geostatistics*, volume 9, pages 493–504, 1997.

[Celeux *et al.*, 2003]G Celeux, F. Forbes, and N. Peyrard. EM procedures using mean field-like approximations for Markov model-based image segmentation. *Pattern Recognition*, 36(1):131–144, 2003.

[Chandler, 1987]D. Chandler. *Introduction to Modern Statistical Mechanics*. Oxford University Press, 1987.

[Forbes and Peyrard, 2003]F. Forbes and N. Peyrard. Hidden markov random field model selection criteria based on mean field-like approximations. *IEEE trans. PAMI*, 25(8), 2003.

[Lazebnik *et al.*, 2003a]S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proc. ICCV*, 2003.

[Lazebnik *et al.*, 2003b]S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant regions. In *Proc. CVPR*, 2003.

[Lindeberg and Garding, 1997]T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-d depth cues from affine distorsions of local 2-d brightness structure. *Image and Vision Computing*, 15:415–434, 1997.

[Nigam *et al.*, 2000]K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[Rosenfel *et al.*, 1976]A. Rosenfel, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Systems, Man, and Cybernetics*, 6(6):420–433, 1976.