Markov random fields for textures recognition with local invariant regions and their geometric relationships

Juliette Blanchet, Florence Forbes, Cordelia Schmid

Markov random fields for textures recognition with local invariant regions and their geometric relationships

Juliette Blanchet, Florence Forbes, Cordelia Schmid
Teams Lear and Mistis, Inria Rhône-Alpes
655 av. de l'Europe, Montbonnot, 38334 Saint Ismier Cedex, France
firstname.lastname@inrialpes.fr

Abstract

This paper describes a new probabilistic framework for recognizing textures in images. Images are described by local affine-invariant descriptors and their spatial relationships. We introduce a statistical parametric models of the dependence between descriptors. We use Hidden Markov Models (HMM) and estimate the parameters with a recent technique based on the mean field principle. Preliminary results for texture recognition are promising and outperform existing techniques.

1 Introduction

Local photometric descriptors computed for invariant interest regions have proved to be very successful in applications such as object recognition [14], texture classification [8] and texture recognition [9][1]. They are distinctive, robust to occlusions and invariant to image transformations. However, the geometric organization of these local invariant descriptors is very informative. Modeling their relative spatial organization is still an open issue. It is not clear which organizational model will prove to be the most useful, and statistical issues for estimating and selecting such models remain to be solved.

In this paper, we introduce a model based on Markov Random Fields and focus on texture recognition as an application. We use affine-invariant regions to create a sparse texture representation [8]. Such a representation has shown to perform well for texture classification, but it does not account for the organization of the detected regions within the image. In [9] neighborhood statistics are modeled by co-occurrence of descriptors and included into the recognition step based on relaxation [16]. This permits to refine the texture membership probabilities, but does not use an explicit organizational model of the data during learning. Similarly, other recent representations [13, 17] use features augmented with spatial information. They used a two-level scheme with intensity-based textons at the first level and histograms of textons distributions over local neighborhoods at the second level. However, beyond this augmentation, no spatial model is explicitly assumed so that the neighborhood information captured is somewhat weakened. Our claim

[1] Texture recognition identifies the texture class for an image location whereas texture classification determine the texture class of an image.

is that there is some gain in assuming that the feature vectors are dependent statistical variables and consequently in using parametric statistical models to account for this dependencies explicitly.

We show that recognition can be improved by using a Hidden Markov Model (HMM) as organizational model when learning the texture classes. The parameter estimation of such a model is in this context not trivial. We use recent estimation procedures based on the Expectation-Maximization (EM) algorithm and on the mean field principle of statistical physics [5].

2 Hidden Markov Models for textures

For the feature extraction stage, we follow the texture representation method described in [9] for its advantages over methods proposed in the recent literature [3, 7, 13, 17]. 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. More specifically, we use the affine-adapted Laplacian blob detector based on the *scale and shape selection* framework developed by Lindeberg et al. [10, 11]. Unlike most existing methods that use fixed-size windows to compute the descriptors, *shape selection* determines the regions over which the descriptors are computed automatically using an *affine adaptation process* [10]. The detector first finds locations in scale space where a normalized Laplacian measure reaches a local maximum. Informally, the spatial coordinates of the maximum define the center of a circular blob and the scale at which the maximum is achieved becomes the *characteristic scale* of the blob. Next, the *affine adaptation process* based on the second moment matrix turns the regions found by the detector into ellipses defined by $(i-i_0)^T M(i-i_0) \leq 1$, where i_0 is the center of the ellipse and M is a 2×2 symmetric *local shape matrix* (see [10, 14] for details). The neighborhood of a region represented by a given ellipse can be naturally computed by adding a constant amount (15 pixels in our implementation) to the major and minor axes and by letting the neighborhood consist of all the 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.

Each detected region is then described by a feature vector (descriptor). The descriptors we use are intensity domain *spin images* [8] rescaled to have a constant norm and flattened into 80-dimensional feature vectors. These rotation invariant descriptors are computed on the patches normalized by matrices M. In this work the spin image size is 5×16. Note that varying the size to 10×10 did not change the results significantly.

Our model assumes that descriptors are random variables with a specific probability distribution in each texture class. In [9], the distribution of descriptors in each texture class is modeled as a Gaussian mixture model where each component corresponds to a sub-class. This assumes that the descriptors are independent variables and does not take into account the strong neighborhood relationships between feature vectors. Here we model the distribution of the descriptors as a Hidden Markov Model (HMM) with K components and an appropriate parametrization 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 all model parameters *i.e.* $\Psi_m = (\theta_{mk}, \beta_m, k = 1 \ldots K)$. Our approach differs from [9] 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 $z = (z_1, \ldots, z_n)$ as the 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, i.e. the second sum 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. If α_m is 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 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 traditionally used for image segmentation. Note that this model is not 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, $\mathbb{B}_m$ can be very different from $\beta \times I$. Texture m is then represented by an HMM defined by parameters Ψ_m with $\Psi_m = (\mu_{mk}, \Sigma_{mk}, \alpha_m(k), b_m(k,l), k, l = 1, \ldots, K)$.

3 Learning the distribution of descriptors and their 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 [15] and [9] for more details on how to implement this generalization.

Here the training data consists 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 parameter estimation in problems with hidden data (here the sub-class assignments). In particular, it has been widely used for estimating independent mixture models. For such models, the independence assumption leads to an easy implementation of the algorithm (*cf.* [12]). 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 approximations based on the mean field principle [4]. The idea is to derive from the intractable Markov distribution a factorized model approximating the original model and for which implementing EM is easy. This allows to take the Markovian structure into account while preserving the good features of EM. More specifically, the factorized model is built as a product of marginal probabilities obtained by considering in turn each location i. For each i, it consists in neglecting the fluctuations in the neighborhood of i by setting the values at neighboring locations to constants. Doing this for all locations requires a set of constant values denoted by $\tilde{z}_1,\ldots\tilde{z}_n$ which are not arbitrary but satisfy some appropriate consistency conditions (see [4]). The mean field approximation consists in setting the $\tilde{z}_1,\ldots\tilde{z}_n$ to mean values. In this paper, we used the *simulated field* algorithm, based on simulated $\tilde{z}_1,\ldots\tilde{z}_n$, for it shows better performance in segmentation tasks (see [4]). Note that we had to extend these algorithms to incorporate the estimation of the matrix $\mathbb{B}_m$ and to include an irregular neighborhood structure coming from descriptors locations and not from regular pixel grids like in [4].

For comparison we also consider a different way to learn texture that does not use the HMM formalism. We used a penalized EM algorithm for spatial data called NEM for Neighborhood EM [1]. 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 [9] 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.

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. Then, each region can possibly be in one of $M \times K$ sub-classes. To identify these sub-classes, the model of the descriptor distribution has to incorporate the information learned for each texture. To do so, 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 during training. As regards the hidden field, the pair-wise potentials, are defined through a square matrix of size $MK \times MK$ 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 mentioned in [9] even when multi-texture images are used for learning, the estimation for these terms is 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 "smoothness 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 HMMs. As briefly explained in Section 3 this involves a set of constants $\tilde{z}_1, \ldots, \tilde{z}_n$ also refined at each iteration. More specifically, let $P_i^{(q)}(c)$ denote the current estimate (at iteration q) of the probability that the ith region has label c. At each iteration new estimates $P_i^{(q+1)}(c)$ are obtained by first simulating new $\tilde{z}_1^{(q)}, \ldots, \tilde{z}_n^{(q)}$ from the current probabilities $P_i^{(q)}(c)$ which are then updated using the equation

$$P_i^{(q)}(c) \quad \propto \quad \exp(\hat{\alpha}_c + \sum_{j \in \mathcal{V}(i)} B(c, \tilde{z}_j^{(q)}))\ f(x_i|\hat{\theta}_c). \tag{1}$$

In equation (1), each $\tilde{z}_i^{(q)}$ can be associated to a particular probability distribution on the $M \times K$ sub-classes which is 1 for sub-class c if $\tilde{z}_i^{(q)} = c$ and 0 for subclasses $c' \neq c$. Writing then $\tilde{z}_i^{(q)}(c)$ which is 1 if $\tilde{z}_i^{(q)} = c$ and 0 otherwise, equation (1) becomes

$$P_i^{(q)}(c) \propto \exp(\hat{\alpha}_c + \sum_{j \in \mathcal{V}(i)} \sum_{c'} B(c, c') \tilde{z}_j^{(q)}(c'))\ f(x_i|\hat{\theta}_c). \tag{2}$$

The normalising terms ensuring that for all i, $\sum_c P_i^{(q)}(c) = 1$ are not relevant in this presentation and are not written. Equation (1) shows that the updating of the probabilities is based on two terms. The exponential term in the right hand-side is a spatial regularizing term measuring through matrix B how "compatible" label c at location i and the label of its neighbors are. Note that in equation (1) the labels at the neighboring sites are unknown and are therefore represented by the current estimates $\tilde{z}_j^{(q)}$. The second term sometimes called a likelihood term involves the Gaussian density for sub-class c and the observed descriptor x_i at location i. It represents the likelihood that x_i belongs to sub-class c and is therefore measuring how consistent c is with the observation.

Each iteration is then a balance between spatial regularization and adequation to the observation model. We used such a formulation in order to make the comparison with the relaxation algorithm used in [9] and NEM algorithm clearer. The NEM algorithm uses the following updating equation

$$P_i^{(q)}(c) \propto p_c^{\text{NEM}}\ \exp(\sum_{j \in \mathcal{V}(i)} \sum_{c'} B(c, c')\ P_j^{(q-1)}(c')) f(x_i|\theta_c^{\text{NEM}}), \tag{3}$$

while relaxation uses

$$P_i^{(q)}(c) \propto P_i^{(q-1)}(c)\ (1 + \sum_{j \in \mathcal{V}(i)} \sum_{c'} B(c, c')\ P_j^{(q-1)}(c')). \tag{4}$$

NEM also involves a regularizing and an observation model term while relaxation only involves a regularizing term. However the observations are usually used to initialize the procedure. This differences will be further discussed in Section 5. Note that the standard EM algorithm for independent Gaussian mixtures with no spatial information does not involved any regularization and a similar formulation does not exist. Without spatial information, when all parameters are fixed, the algorithm reduces to a single iteration.

Using any of the algorithms mentioned above, membership probabilities are then obtained for each texture class. For each region located at i, we get an estimate of $P(Z_i = c_{mk}|x_i)$ 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 some preliminary experiments, the HMM in the test stage was only partly defined. All parameters were fixed 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. However this possibility is worth further investigation.

5 Experimental Results

5.1 Single texture images

Preliminary experiments are obtained with a data set containing 140 single texture images of seven texture classes. Images have been gathered over a wide range of viewpoints and scale changes (Figure 1). Each texture is represented by 20 images partitioned into a training and a test set of 10 single texture images each. For the mixture models, we set the number of sub-classes to $K = 10$ for each texture. We also selected K automatically using the Bayesian Information Criterion (BIC) of Schwarz [6], but we did not observe significantly better recognition results. The Gaussian distributions were restricted to diagonal covariance matrices. When dealing with high dimensional data, this reduces the number of parameters to be estimated significantly and tends to avoid numerical problems with singular matrices. For each texture class m, we selected with BIC the covariance model with $\Sigma_{mk} = \sigma_m^2 I$ for all $k = 1, \ldots K$. In terms of BIC values this simple model is slightly better than the more general model with $\Sigma_{mk} = \sigma_{mk}^2 I$ depending on k. Similar recognition results are obtained with less covariance parameters to estimate. Table 1 shows recognition 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 assumes that each texture class has the same probability to occur in the test image. A region is then classified as belonging to the texture class with the best mixture likelihood

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.

(learned parameters). "Relaxation" refers to the method used in [9]. 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 [16], using some additional spatial information deduced from the learning co-occurrence 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 more consistent and appropriate. Table 1 shows better rates for NEM when compared to relaxation. The method using HMMs is the only one where the descriptors are modeled as statistically dependent variables. It provides a way to analyse and control these dependencies through a number of parameters. "Simulated Field" in Table 1 refers 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. Compared 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 difficulties in recognizing textures 6 and 7. Both textures contain images with very strong illumination changes as well as blurred images; a possible reason is that our descriptors and/or our neighborhood structure may not be invariant enough. These preliminary experiments show 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.

Note the good performance of relaxation despite the fact that the observed data x are missing in iteration (4). This corresponds to a maximum a priori rather than to a maximum a posteriori principle which is more satisfying from a statistical point of view. A possible reason is that such an iteration is robust to deviations from the Gaussian mixture model. Statistical procedures to test the validity of the Gaussian mixture assumption are not available but we consider studying data transformations (extensions of standard Box-Cox transformations) to get closer to Gaussian data when necessary.

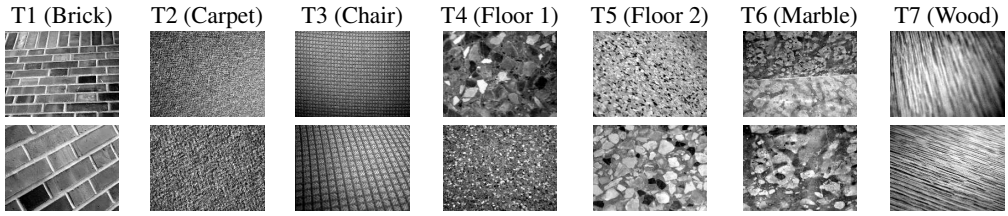

Figure 1: Samples of the texture classes used in the experiments.

5.2 Multi-texture images

The algorithms are then tested using 62 multi-texture images. Figure 2 shows a typical example of recognition results using maximum likelihood, relaxation, NEM and the simulated field algorithm for an image containing textures wood and chair. It illustrates that including spatial information via relaxation, NEM or simulated field algorithm, improves classification results. It also show that the best results are obtained when using the simulated field algorithm that is when statistical dependencies are modeled explicitly through a Markov model. More specifically the chair part is very well recognized while some wrong assignments are made for the wood part. This is consistent with results in Table 1 showing that texture wood was more difficult to learn.

Results of the simulated field algorithm are displayed in Figure 3. The marble texture in 3(a) is not well recognized (cf. Table 1), but is mainly mistaken for the neighboring textures wood and brick. This illustrates the characteristic behavior of the simulated field algorithm which tends to group neighboring regions in the same texture class. Figures 3 (b) and (c) show that classification results tend to decrease when image quality decreases. In Figure 3 (b), the upper wood part is blurred resulting in classification results worse than those of the sharper image 2. Similarly, the brick wall in Figure 3 (c) is badly lit so that the corresponding regions are not very well classified although the brick texture was well learned (see Table 1). These last examples suggest that bad classification results are at least partially due to the descriptor quality, rather than from limitations of the proposed algorithms. Also most classifications errors occur at texture boundaries suggesting that the neighborhood graph has a significant part to play and may required more care.

6 Conclusion

We based our work on recent techniques for image description going beyond regular grid of pixels to sets of irregularly spaced features. Our aim was to show that statistical parametric models could be introduced to account for spatial geometric relationships between feature vectors. We showed that Hidden Markov Models were natural candidates and focused on a texture recognition task for 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 adjusted to incorporate a priori

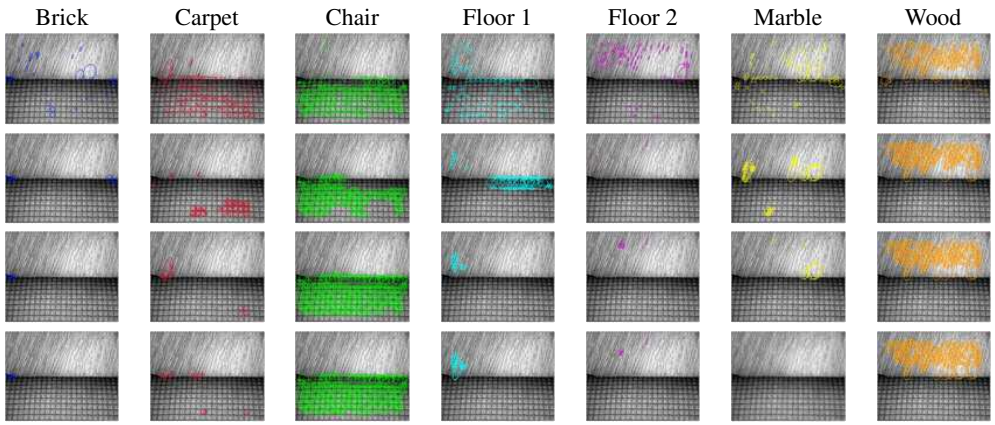

Figure 2: From top to bottom row: recognition results using maximum likelihood, relaxation, NEM algorithm and the simulated field algorithm for an image with chair and wood textures. Please print the figure in colour.

knowledge with respect to texture proportions or strength of interactions. Other methods such as relaxation are much less readable in that sense.

Preliminary results are promising and illustrate a general statistical formalism. Future work includes investigation in other contexts for example object recognition. A more specific analysis with respect to the choice of the neighborhood structure would be necessary. In particular, the use of stronger geometric neighborhood relationships that take into account affine shape while preserving the maximum amount of invariance would be worth investigation. Also the methodology presented here for feature vectors could be investigated with other image description techniques.

References

[1] C. Ambroise,V. Mo Dang and G. Govaert, "Clustering of spatial data by the EM algorithm", geoENV I- Geostatistics for Environmental Applications,Quantitative Geology and Geostatistics, Vol. 9, Dordrencht, Kluwer Academic Publishers,pp. 493-504, 1997.

[2] M. Akaike, "Information theory and an extension of the maximum likelihood principle", *Second Int. Symposium on Information Theory*, 1973.

[3] B. Bradshaw, B. Scholkopf and J. Platt, "Kernel Methods for Extracting Local Image Semantics", *Microsoft Research Technical Report*, MSR-TR-2001-99, 2001.

[4] G. Celeux, F. Forbes and N. Peyrard, " EM Procedures Using Mean Field-Like Approximations for Markov Model-Based Image Segmentation", *Pattern Recognition*, 36(1), p. 131-144, 2003.

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

[6] F. Forbes and N. Peyrard, "Hidden Markov Random Field Model Selection Criteria based on Mean Field-like Approximations", *IEEE trans. PAMI*, 25(8),p. 2003,

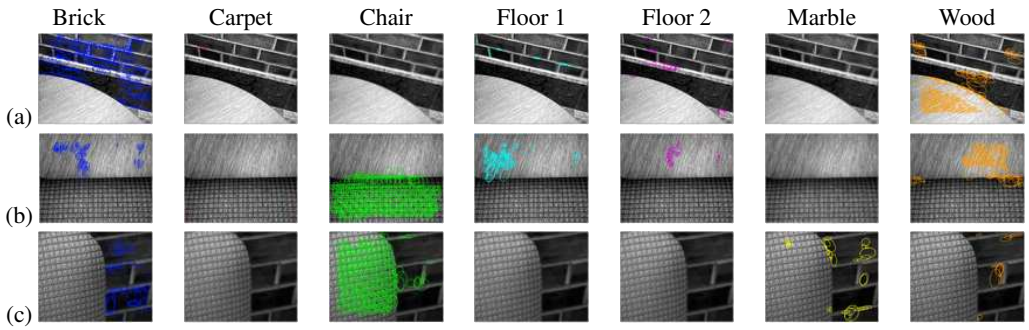

Figure 3: Recognition results using the simulated field algorithm: (a) for an image with textures brick, marble and wood; (b) for an image with textures chair and wood, the latter being slightly blurred; (c) for an image with bad lighting including textures chair and brick. Please print the figure in colour.

[7] S. Kumar and M. Hebert, "Man-Made Structure Detection in Natural Images Using a Causal Multiscale Random Field", *Proc. CVPR*, 2003.

[8] S. Lazebnik, C. Schmid and J. Ponce, "Sparse Texture Representation Using Affine-Invariant Regions", *Proc. CVPR*, 2003.

[9] S. Lazebnik, C. Schmid and J. Ponce, "Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition", *Proc. ICCV*, 2003.

[10] 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, pp. 415-434, 1997.

[11] T. Lindeberg, " Feature Detection with Automatic Scale Selection", *IJCV*, 30(2), pp. 77-116, 1998.

[12] G. J. McLachlan and D. Peel, "Finite Mixture Models", Wiley, 2000.

[13] J. Malik, S. Belongie, T. Leung and J. Shi, "Contour and Texture Analysis for Image Segmentation", *IJCV*, 43(1), pp. 7-27, 2001.

[14] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector", *Proc. ECCV*, 2002.

[15] K. Nigam, A. McCallum, S. Thrun and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents using EM", *Machine Learning,*39 (2/3), pp. 103-134, 2000.

[16] A. Rosenfel, R. Hummel and S. Zucker, "Scene Labeling by Relaxation operations", *IEEE Trans. Systems, Man, and Cybernetics,* 6(6), pp.420-433, 1976.

[17] C. Schmid, "Constructing Models for Content-Based Image Retrieval", *Proc. CVPR*, 2001.