GAUSSIAN MIXTURE MODELS OF TEXTURE AND COLOUR FOR IMAGE DATABASE RETRIEVAL

H. Permuter, J. Francos[*]

Ben-Gurion University
Electrical and Computer Engineering Department
Israel

I. H. Jermyn[*]

INRIA Sophia Antipolis
Ariana Project (CNRS/INRIA/UNSA)
Sophia Antipolis, France.

ABSTRACT

We introduce Gaussian mixture models of 'structure' and colour features in order to classify coloured textures in images, with a view to the retrieval of textured colour images from databases. Classifications are performed separately using structure and colour and then combined using a confidence criterion. We apply the models to the VisTex database and to the classification of man-made and natural areas in aerial images. We compare these models with others in the literature, and show an overall improvement in performance.

1. INTRODUCTION

In many domains of image processing, there is a strong correspondence between entities in the scene and textures (by texture, we mean both what we will later call 'structure' information, and colour information) in the image. This implies that the ability to classify these textures can furnish important semantic information about the scene. Consequently, the problems of texture description and classification, and the closely related problem of segmentation, have received considerable attention, with numerous approaches being proposed (see, *e.g.*, [1] and references therein). In particular, in the field of content-based image retrieval, the ability to answer the question: "Is there a significant amount of such-and-such texture in this image?", can be the basis for many types of query.

One approach to characterizing textures is to use statistical models. Many kinds of statistical models have been applied to texture classification, but the closest to the models proposed in this paper are those based on various Markov models. Motivated by the desire to incorporate contextual information, Li and Gray [2] proposed a 2D HMM for image classification. A somewhat different model is the non-causal HMM described in [3]. Another recently popular class of models uses hidden Markov trees to model the joint statistics of wavelet coefficients [4, 5, 6]. In [4, 5, 7], an independent mixture model (IMM) in the wavelet domain is introduced. This model bears some similarities to the GMM proposed in this paper, since it employs a mixture of Gaussians, but in the IMM each wavelet feature is modelled separately by a mixture of models, and it is assumed that the features are independent. In the GMM framework adopted in this paper, no independence assumption about the features is made. The performance of the IMM-based classification was evaluated in [4, 5, 7], using the wavelet coefficients at each pixel as features.

2. GMMS FOR TEXTURE CLASSIFICATION

We introduce our classification model, describe Gaussian mixture models and motivate their use. We assume that we are dealing with N texture classes, labelled by $n \in N \cong \{1, \ldots, N\}$, corresponding to different entities.

2.1. Classification model

Since texture is not a local phenomenon, in order to classify a pixel one must take into account a neighbourhood of that pixel. We will compute features from, and assign classes to, $S \times S$ subimages called 'blocks'. The set of blocks is denoted B. We define the neighbourhood $P(b)$ of a block b, called a 'patch', to be the set of blocks in a larger $T \times T$ subimage with b at its centre. We denote by D_b the data associated to block b, and by $\nu_b \in N$ the classification of b. Given the likelihood of the data in a block given its class, $\Pr(D_b|\nu_b)$, we use the following classification rule:

$$\nu_b = \arg\max_{n \in N} \prod_{b' \in P(b)} \Pr(D_{b'}|\nu_{b'} = n) \qquad (1)$$

This says: "Assign to a block b that class n which, if all the blocks in $P(b)$ had class n, would maximize the probability of the data in $P(b)$" (we assume conditional independence of the data in the blocks in a patch given the classification).

[*] This work was supported by European project MOUMIR, HP-99-108 (`www.moumir.org`). The authors would also like to thank Professors R. Gray and J. Li for permission to use the aerial images contained in this paper.

The effect of this classification rule is similar to that of a Potts prior, in that it encourages spatial homogeneity of the classification. Its advantage is that it is not necessary to consider the classifications of neighbouring blocks in making a classification decision. This reduces computation time considerably.

2.2. Gaussian mixture models

The data D_b associated to each block will be a vector of features, denoted $\vec{x}$. We must choose, for each texture class, a probability distribution that represents the feature statistics of a block of that class. We will use Gaussian mixture models for this purpose. Thus, for a given texture class, the probability that $\vec{x}$ be observed is a convex combination of M Gaussian densities:

$$P(\vec{x}|\{p_i, \vec{\mu}_i, \Sigma_i\}) = \sum_{i=1}^{M} p_i b(\vec{x}, \vec{\mu}_i, \Sigma_i) \tag{2}$$

where $b(\vec{x}, \vec{\mu}, \Sigma)$ is a Gaussian of mean $\vec{\mu}$ and covariance Σ. The parameters for a given class are thus $\{p_i, \vec{\mu}_i, \Sigma_i | i \in M\}$.

It is clear that modelling a texture class with a GMM rather than a single Gaussian gives a great deal of added flexibility to the model. Indeed, if one is allowed an arbitrary number of components, any continuous density function can be approximated to any desired accuracy. A GMM is also the natural model to use if a texture class contains a number of distinct subclasses, as is often the case (for example, forest texture in an aerial image).

2.3. Parameter estimation for GMMs

To apply the above classification procedure, we must learn the parameters of the GMM models. Given a training set consisting of the data from T blocks of a particular texture class, $X = \{\vec{x}_t | t \in T\}$, we would like to estimate the parameters of the Gaussian mixture density using a maximum likelihood estimator. Fortunately, maximum likelihood parameter estimation for a GMM can be solved using the EM algorithm [8], or by using k-means. Lack of space prevents an exposition of these algorithms here, but note that the update steps for GMMs are expressible in closed form.

2.4. The BKG model

In addition to the texture classes that we wish to classify, we introduce also the background (‘BKG’) class. Its parameters are learned from the blocks in the union of the training sets of each class, the k-means algorithm being used because of its faster convergence properties on large amounts of data. The BKG model has two roles. First, it is used to initialise the training of the individual texture models, thus ensuring that the initialisation is the same for all classes and not biased towards any one. Second, the BKG model is used as a ‘no decision’ class. If the BKG model is more likely than any of the individual classes, then no decision is made.

3. FEATURE EXTRACTION

We must choose sizes for a block and a patch. For segmentation, there is a trade-off between our ability to discriminate classes and the accuracy of boundary estimation. However, for retrieval purposes, the accuracy of texture boundaries is not such a big issue. We choose a block size of $S = 16$, since this seems large enough to capture a reasonable sample of the largest structures in the textures in the images with which we are dealing. Choosing patch size is equivalent to choosing a degree of smoothing for the classifications: there is a tendency for blocks near the centre of a given patch to be assigned the same class, since their corresponding patches have many blocks in common. We choose square patches containing nine blocks.

3.1. Structure Features

Structure features are designed to capture spatial regularity of the texture over the block. We extract structure information from the intensity images alone. We compared several sets of features for this purpose: the energies in different wavelet subbands for both Haar and Daubechies wavelets; AR models of different orders; and the energies of DCT coefficients in regions of frequency space corresponding to a wavelet decomposition. We found in practice that the wavelet-like DCT and the Haar wavelet features performed best, although only the AR models did significantly worse. The DCT is computationally the most efficient however, and we chose these energies as our structure features.

3.2. Colour features

Colour provides an extremely powerful cue for the distinguishing of different entities in the scene. As colour features, we used the mean RGB values over a block and the data covariance of the RGB values over a block. Since the covariance matrix is symmetric, only half of it, including the diagonal, is included in the feature vector. The colour feature vector is thus a 9-dimensional vector, 3 coming from the mean and 6 from the covariance.

3.3. Combining colour and structure

We combine the colour and texture models in the following way. For each block, and for both structure and colour features, we compute the right hand side of equation 1 for all classes. Let $n_1(b)$ and $n_2(b)$ be the maximizing and next-to-maximizing classes for block b. We define the ‘confidence’

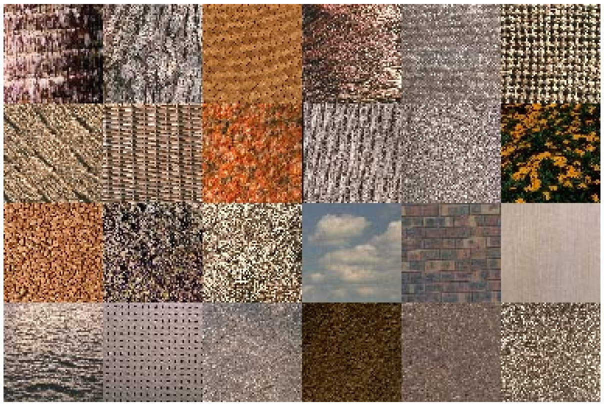

Fig. 1. Textures from the Vistex Database: Bark0; Bark12; Fabric0; Fabric4; Fabric7; Fabric8; Fabric11; Fabric13; Fabric15; Fabric17; Fabric19; flowers0; Food.0000; Leaves12; Grass1; Clouds0; Brick0; Wood2; water0; Tile7; Stone4; Sand.0000; Misc2; Metal0.

of the classification decision taken as:

$$C(b) = \Pr(D_b|\nu_b = n_1(b)) / \Pr(D_b|\nu_b = n_2(b)) \qquad (3)$$

If the classifications resulting from using the structure features and the colour features conflict, we choose the decision with the highest confidence.

4. EXPERIMENTAL RESULTS

The experiments in this section were conducted on the MIT Vision Texture (VisTex) database, and on the aerial images of the San Francisco Bay area that were used in [9, 2, 10, 11].

4.1. Vistex texture database

We chose randomly 24, 512×512 textured colour images from the Vistex database. The textures are displayed in figure 1. Each image was divided into subimages of size 32×32 pixels. All blocks extracted from the first 96 subimages of each texture were used for training, while the remaining 160 subimages were used for testing.

For each class we trained a GMM with five components using 30 iterations of the EM algorithm. We chose five components because increasing the number of components did not improve the results significantly. We used 30 iterations for a similar reason: the EM algorithm appeared to have converged after this number of iterations. We used the same initialization for each texture class: the BKG model. The results of the classification using the colour features, the structure features and the combined decision are shown in table 1.

Texture	Colour	Structure	Both
Bark0	100	99	100
Bark12	93	73	92
Fabric0	100	96	100
Fabric4	82	100	100
Fabric7	96	100	100
Fabric8	99	94	98
Fabric11	100	86	100
Fabric13	94	98	100
Fabric15	100	98	100
Fabric17	92	98	99
Fabric19	99	99	100
Flowers0	100	96	99
Food0	100	96	100
Leaves12	19	99	94
Grass1	81	88	92
Clouds0	100	100	100
Brick0	100	99	100
Wood2	88	98	99
Water0	36	43	44
Tile7	56	94	86
Stone4	87	96	94
Sand0	100	96	100
Misc2	99	96	100
Metal0	97	94	96
	Avg. Colour	Avg. Texture	Avg. Both
	88.3	93.2	95.6

Table 1. VisTex results

4.2. Aerial images

This database includes six 512×512 grey-scale images. There exist also manual segmentations of the images into man-made and natural areas. We use these segmentations as ground truth. The images are displayed in Figure 2.

We used this database for evaluation exactly as it was used in [9, 2]. For each iteration, one image was used as test data, and the other five were used as training data. Performance is evaluated by averaging over all iterations. Each class ('man-made' and 'natural') was modelled by a five-component GMM of the structure features. For initialization we used the BKG model.

The results from the GMM algorithm were compared to the results from other statistical models reported in [2, 9]: the 2D HMM (two-dimensional hidden Markov model) [2]; the 2D MHMM (two-dimensional multi-resolution hidden Markov model) [9]; CART (a decision tree algorithm) [12]; and LVQ1 (version 1 of Kohonen's learning vector quantization) [13]. The classification error rates for each test image in the six-fold cross-validation and the average error rates are listed in table 2.

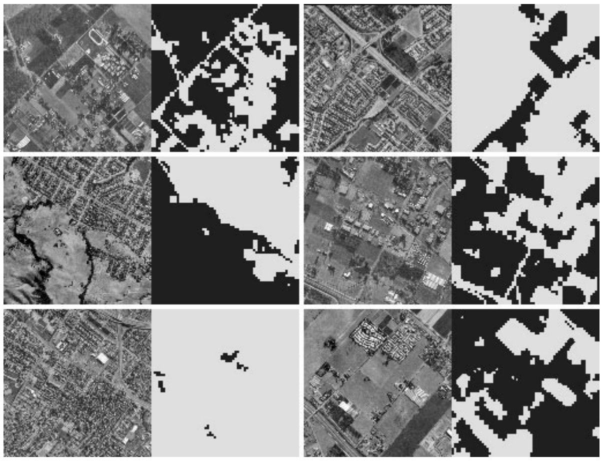

Fig. 2. Aerial images. On the left of each pair, the original images. On the right, the manual segmentations. The dark areas are natural, the lighter areas, man-made.

Iteration	CART	LVQ1	HMM	MHMM	GMM
1	22.6	21.6	19.0	17.3	16.4
2	18.0	19.1	17.6	16.3	14.0
3	28.9	28.4	20.3	17.8	19.6
4	25.2	24.9	24.0	20.5	19.1
5	14.2	18.7	18.3	12.5	4.2
6	20.2	18.1	13.3	11.5	15.4
Ave.	21.5	21.8	18.8	16.0	14.8

Table 2. Classification error rates (percentage) by algorithm

5. CONCLUSION

We have described Gaussian mixture models of texture and colour features, and used them for the classification of textures in the VisTex database and for classifying ‘man-made’ and ‘natural’ areas in aerial images. We have compared these models with others in the literature, and shown an overall improvement in performance.

6. REFERENCES

[1] T. Reed and J. M. H. du Buf, “A recent review of texture segmentation and feature extraction techniques,” *CVGIP Image Understanding*, vol. 57, pp. 359–372, 1993.

[2] J. Li, A. Najmi, and R. M. Gray, “Image classification by a two-dimensional hidden markov model,” *IEEE Trans. Sig. Proc.*, vol. 48, no. 2, pp. 517–533, 2000.

[3] B. R. Povlow and S. M. Dunn, “Texture classification using noncausal hidden markov models,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, no. 10, pp. 1010–1014, 1995.

[4] M. Crouse, R. Nowak, and R. Baraniuk, “Wavelet-based statistical signal processing using hidden markov models,” *IEEE Trans. Sig. Proc.*, vol. 46, no. 4, pp. 886–902, 1998.

[5] G. Fan and X.-G. Xia, “Maximum likelihood texture analysis and classification using wavelet-domain hidden markov models,” in *Proc.* 34^{th} *Asilomar Conf. Signals, Systems and Computers*, October 2000.

[6] H. Choi and R. G. Baraniuk, “Image segmentation using wavelet-domain classification,” *Proc. SPIE Tech. Conf. Mathematical Modeling, Bayesian Estimation, and Inverse Problems*, pp. 306–320, July 1999.

[7] H. A. Chipman, E. D. Kolaczyk, and R. E. McCulloch, “Adaptive bayesian wavelet shrinkage,” *J. Am. Statist. Ass.*, vol. 92, no. 440, pp. 1413–1421, 1997.

[8] A. Dempster, N. Larid, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. R. Statist. Soc.*, vol. 39, pp. 1–38, 1977.

[9] J. Li, R. Gray, and R. Olshen, “Multiresolution image classification by hierarchical modeling with two dimensional hidden markov models,” *IEEE Trans. Inform. Th.*, pp. 1826–1841, 2000.

[10] K. L. Oehler and R. Gray, “Combining image compression and classification using vector quantization,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, no. 5, pp. 461–473, May 1995.

[11] K. Oehler, *Image Compression and Classification using Vector Quantization*, Ph.D. thesis, Stanford University, 1993.

[12] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Chapman & Hall, 1984.

[13] T. Kohonen, *Self-Organization and Associative Memory.*, Spring-Verlag, Berlin, 1989.