Unsupervised Texture Segmentation Using 2-D AR Modeling and a Stochastic Version of the EM Procedure

Claude Cariou, Kacem Chehdi

LASTI - Groupe Image
Ecole Nationale Supérieure de Sciences Appliquées et Technologie
BP 47 - 6, rue de Kerampont 22305 Lannion Cedex - France
Tel: (+33) 02 96 46 50 30 ; Fax: (+33) 02 96 37 01 99
e-mail: claude.cariou@enssat.fr, kacem.chehdi@enssat.fr

ABSTRACT

The problem of textured image segmentation upon an unsupervised scheme is addressed. Until recently, there has been few interest in segmenting images involving possible complex random texture patterns. It is also a fact that most unsupervised segmentation techniques generally suffer from the lack of information about the correct number of texture classes. Therefore, this number is often assumed known *a priori*. On the basis of the so-called SEM (Stochastic Expectation Maximisation) algorithm, we try to perform a reliable segmentation without such prior information, starting from an upper bound for the number of texture classes. The image model first assumes an autoregressive (AR) structure for the class-conditional random field, and in a further step, a Markovian structure of the region process. The application of this method on a textured mosaic is presented.

1 INTRODUCTION

Image segmentation is one of the important tasks in computer vision systems, and many fields of application are concerned with it, including robotics, remote sensing, medical imaging, etc. There exist various way of segmenting images, and the main approaches are region-based, contour-based and mixed (region-contour) techniques. In this correspondence, we deal with the region-based image segmentation problem. Therefore, the objective is to separate one image into disjoint regions, within which the presence of homogeneous random fields is assumed. These regions may possibly have the same average grey level and variance, but not the same spatial distribution (second or higher order correlations), i.e. the texture.

During the last two decades, there has been much interest in image segmentation using Markov random fields (MRF's), leading either to supervised or unsupervised techniques. However, most techniques assume region conditional i-i-d random fields, which obviously may not correctly reflect the correlation between adjacent pixels that may occur in most practical situations.

Another fact is that most unsupervised techniques need some prior knowledge about the number of texture classes (be they correlated or not). Also, the general problem of unsupervised techniques is known as a cluster validation problem [1].

A stochastic model-based technique is presented herein for the segmentation of textured images which are processed at two levels of resolution, following the method developed by Cohen and Fan [2]. The first resolution level consists in regularly spaced small-sized windows taken from the original image and from which can be computed *i*) the correct number of texture classes, *ii*) features that will reveal the textural information and *iii*) a rough but rather accurate 'block-like' segmentation. The second image resolution level concerns the pixel itself, using the above information to refine the segmentation in an iterative scheme.

This method proceeds in three steps. First a causal non-symmetric half-plane (NSHP) 2-D AR modeling of the textured image is performed within possibly overlapping windows covering the entire image. The second step uses a stochastic version of the Expectation-Maximisation algorithm, called SEM [3], the purpose of which is firstly (classically) to estimate the parameters of identifiable mixed distributions, i.e. the prior probabilities and the parameters of the distribution of each texture class, and secondly (less classically) to estimate the correct number of classes. This estimation scheme is performed on the basis of previously computed AR features and is followed by a rough image pre-segmentation. The third step starts from both the original image and the preceding analysis with the aim to refine the segmentation, assuming a Markovian structure of the underlying region process and trying to recursively maximize the posterior marginals.

2 DESCRIPTION OF THE METHOD

2.1 First Step : Computation of AR features vectors

The objective of this step is to produce a set of reliable features descriptive of the textured nature of the image, within regions as small as possible and throughout the entire image. Various ways exist for texture feature extraction, including non-parametric (Fourier transform - based), semi-parametric (co-occurrence matrix - based), and parametric (Gauss-Markov or autoregressive random fields) techniques. The latter provide an interesting way due to the reduced number of parameters used to describe the texture.

In this work we use 2-D causal non-symmetric half-plane (NSHP) AR modeling (see for example [4, Chapter 15]) mainly because of the linearity of the maximum likelihood (ML) estimation of the parameters. Another important issue of image linear modeling for the following lies in the asymptotic multivariate gaussianity of AR parameters ML estimates [5]. Thus the textured image model within a small window indexed by j is :

$$\begin{aligned} y^{(j)}(m,n) &= \sum\sum_{(k,l)\ \in\pi} \theta^{(j)}(k,l)\, y^{(j)}(m-k,n-l) \\ &\quad + \sqrt{\sigma^{2\,(j)}}w^{(j)}(m,n) \;+\; \theta_0^{(j)} \end{aligned} \tag{1}$$

where $\{\theta^{(j)}(k,l);\ (k,l)\in\pi\}$ are AR parameters, π is the 2-D NSHP support of linear prediction, $\{w^{(j)}(m,n)\}$ is the i-i-d driving noise sequence with unit variance, $\sigma^{2\,(j)}$ represents the variance of the driving noise, and $\theta_0^{(j)}$ is a non-zero constant related to the positive mean of the window. The computation of AR parameters ML estimates from the observation $\{y^{(j)}(m,n), 0\le m\le M-1,\ 0\le n\le M-1\}$ within window j is straightforward and simply requires the inversion of a covariance matrix. At the end of this step is available a set of AR parameters N-dimensional vectors $\{\boldsymbol{\theta}^{(j)} = [\{\theta^{(j)}(k,l)\},\theta_0^{(j)},\sigma^{2\,(j)}]^T;\ 1\le j\le NW\}$, where NW is the number of possibly overlapping windows.

2.2 Second Step : Application of the SEM Procedure

The Stochastic Estimation-Maximisation (SEM) algorithm, just as the well known EM, is aimed at providing the parameters of identifiable mixed distributions, say the prior probabilities and the parameters of each distribution. It also may be readily used in classifying data, and in this way has been applied for the segmentation of satellite images [6]. However, in this latter work, the image model was rather simple, and consisted in regions made of realisations of i-i-d random fields the unknown parameters of which were estimated by the SEM procedure.

In our work, the entries of the SEM procedure are not pixel values, but the AR parameter vectors obtained previously. The SEM algorithm applied to this data is then the following :

Step 0 : Define an *a priori* upper bound K for the number of clusters and affect the same initial posterior probability p_{ij} for each vector $\boldsymbol{\theta}^{(j)}$ to belong to each cluster i ;

Step 1 : Affect randomly each vector $\boldsymbol{\theta}^{(j)}$ to cluster i according to p_{ij} and estimate the prior probability of each cluster p_i by frequencies. If p_i is below a defined threshold (for example, one percent of NW), suppress cluster i and redistribute the corresponding vectors into other clusters ;

Step 2 : Compute the mean vector $\boldsymbol{\mu}_\theta^{(i)}$ and the covariance matrix $\boldsymbol{\Lambda}_\theta^{(i)}$ of each cluster ;

Step 3 : For each vector $\boldsymbol{\theta}^{(j)}$, estimate the posterior probability as :

$$\begin{aligned} p_{ij} &= p_i\, \frac{1}{(2\pi)^{N/2}\sqrt{det\,\boldsymbol{\Lambda}_\theta^{(i)}}}\cdot \\ &exp\left(-\frac{1}{2}\left[\boldsymbol{\theta}^{(j)}-\boldsymbol{\mu}_\theta^{(i)}\right]^T(\boldsymbol{\Lambda}_\theta^{(i)})^{-1}\left[\boldsymbol{\theta}^{(j)}-\boldsymbol{\mu}_\theta^{(i)}\right]\right) \end{aligned} \tag{2}$$

and then return to *Step* 1.

Jointly and at each iteration of the SEM procedure, it is possible to produce an image pre-segmentation in form of a 'block-like' labelled image. This pre-segmentation is obtained following the marginal *a posteriori* mode (MPM) strategy [7], i.e. one choses for each $\boldsymbol{\theta}^{(j)}$ the cluster i^* that verifies

$$i^* = arg\left(\max_i\ p_{ij}\right) \tag{3}$$

The SEM procedure is ran out until convergence of parameter estimates. At the end of this second step, many information is available about the textured image, i.e. the correct number of classes, an estimate of texture class priors, estimates of the texture class parameters, and a rough image pre-segmentation. We are then in position to refine the segmentation on a pixel-based approach.

2.3 Third Step : Final Segmentation

In a way to refine the segmentation from such available information, many approaches may be used, such as the maximum *a posteriori* (MAP), the marginal *a posteriori* modes (MPM), the iterative conditional modes (ICM), or contextual methods. Once again, we chose the MPM technique. To insure a regularisation of the segmentation result, we assume a simple Markovian structure for the underlying region process. In this model, prior conditional probabilities are given by

$$Pr\,(label=i\mid .\,) = \frac{exp\,(\beta\ u_j(i))}{\sum_{l=1}^{K} exp\,(\beta\ u_j(l))} \tag{4}$$

where $u_j(l)$ represents the number of neighbours of site j having label l, and β is a given parameter that governs the grouping of pixels with similar labels within a defined neighbourhood. The objective being to maximise the conditional *a posteriori* distribution $f\,(label(j)\mid label(k), k\neq j, y)$ which in the Bayesian framework is proportional to the conditional density $f\,(y(j)\mid label(j))\ .\ Pr\,(label(j)\mid .\,)$, it can be shown that for each label (or texture class), the posterior probability for pixel j to belong to texture class i is :

$$p_{ij} \propto \frac{1}{\sigma_i} \, exp \left(-\frac{e_i^2(j)}{2\sigma_i^2} + \beta \, . \, u_j(i) \right) \tag{5}$$

where $e_i(j)$ is the difference between the observation and its linear prediction by AR model i over site j, and σ_i^2 is the variance of the AR model i. Thus the algorithm is the following :

Step 0 : Starting from *i*) AR parameter estimates for each texture class, *ii*) the pre-segmentation and *iii*) a given β ;

Step 1 : For each $i = 1, \cdots, K$ and $j = 1, \cdots, M$ (M is the number of pixels or sites), estimate p_{ij} from (5) ;

Step 2 : Chose randomly, for each site j, a possible new label i following p_{ij} ;

Step 3 : Repeat from Step 1 until convergence, then from Step 0 with and increasing value of β.

Starting from a low β parameter from one pass of the second step to another allows mainly to compensate for possible window misclassification derived from the first step while using rather correct estimates of conditional model parameters. Note that the estimation of β is made a non-issue herein and that it may be chosen reasonably within the range $[0.5, 2.0]$
From a more general point of view, one can see that the use of a regularisation model only in the final step avoids the method to converge towards a local maximum of the likelihood function as is a well-known drawback of the ICM method, for example.

3 RESULTS

In this Section, we present a segmentation result obtained with this technique. An experiment is performed with a Brodatz texture mosaic with known regions [8]. Figure 1 shows a 256 by 256 pixels image made of six regions, each having one among four Brodatz textures. During the first step, a texture feature extraction within 50 percent overlapping 16 by 16 pixels windows is performed using a NSHP AR model with four parameters ($N = card(\pi) = 4$). In this experiment, these four parameters only are used for pre-segmentation. Thus we run the SEM procedure with data consisting of 1024 4-D vectors, starting with an upper bound of $K = 10$ texture classes. After 1000 iterations of the SEM procedure, the correct number of texture classes (i.e. $K = 4$) is found, and the pre-segmentation in Figure 2 is obtained. Note that the correct classification rate is better than 80 percent at this level of the method, which is yet satisfactory. Then the final segmentation step is applied using the results previously obtained, i.e. the pre-segmentation and the parameter vector of each texture class. The neighbourhood of the underlying Markovian region process is taken to be a second order one, and as such, for a given site, is made of its eight nearest neighbours. The parameter β was taken to 0.5 at the first pass and then increased to 2.0 by steps of 0.5. Each pass took 20 iterations of Step 3. The final result is shown in Figure 3. A correct classification rate of 97 percent was obtained for this example. Figure 4 shows the error map. It can be pointed out that most misclassified pixels are closely related with region edges due to the breaking of the adequate models and to the high prediction error that occurs in this situation. This suggest the use of some edge information for a further refinement of the final segmentation step. Moreover, one should note that suppressing the two last parameters in each $\boldsymbol{\theta}^{(j)}$ makes the pre-segmentation insensitive both to grey level shifts and expands which may be useful in some applications. On the other hand, the dimensionality of texture features is a key problem, due to the large amount of computations required to reach a stable solution.

4 CONCLUSION

The unsupervised segmentation technique for textured images presented here works on two levels for statistically homogeneous regions retrieval. The first level attempts to extract texture features upon a window-based 2-D AR modeling of the random field and as such may be seen as a kind of pre-attentive vision procedure. The second level allows to classify each pixel of the image on the basis of the previous parameter estimates. This method offers some advantages in comparison with others available in the literature. Firstly it allows the estimation of the correct number of texture classes due to the use of the SEM algorithm which is found to provide other reliable information about the textured image. Secondly, the textured nature of homogeneous regions is taken into account via a modeling which is simple in essence and in the few number of parameters that are needed to describe them. It is important to note that other stochastic models for region-conditional texture may be used within this framework, such as Gauss-Markov random fields (GMRFs) or more general ones such as non-causal autoregressive-moving average models (NCARMA).

References

[1] J. Zhang, J. W. Modestino, 'A model-fitting approach to cluster validation with application to stochastic model-based image segmentation,' *IEEE Trans. PAMI*, **12**, no 10, pp. 1009-1017, 1990.

[2] F. S. Cohen, Z. Fan, 'Maximum likelihood unsupervised textured image segmentation,' *CVGIP : Graphical Models and Image Processing*, **54**, no 3, pp. 239-251, 1992.

[3] G. Celeux, J. Diebolt, 'L'algorithme SEM : un algorithme d'apprentissage probabiliste pour la reconnaissance de mélange de densités,' *Revue de Statistique Appliquée*, **34**, no 2, pp. 30-33, 1994, *in French.*

[4] S. M. Kay, *Modern Spectral Analysis - Theory and Application*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[5] G. Box, G. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Fransisco, 1970.

[6] P. Masson, W. Pieczynski, 'SEM algorithm and unsupervised statistical segmentation of satellite images,' *IEEE Trans. Geosc. Remote Sens.*, **31**, no 3, pp. 618-633, 1993.

[7] J. Marroquin, S. Mitter, T. Poggio, 'Probabilistic solution of ill-posed problems in computational vision,' *J. Amer. Statis. Ass.*, **82**, no 397, pp. 76-89, 1987.

[8] K. Chehdi, C. Cariou, C. Kermad, 'Image segmentation and texture classification using local thresholds and 2-D AR modelling,' *Proc. EUSIPCO-94*, pp. 30-33, 1994.

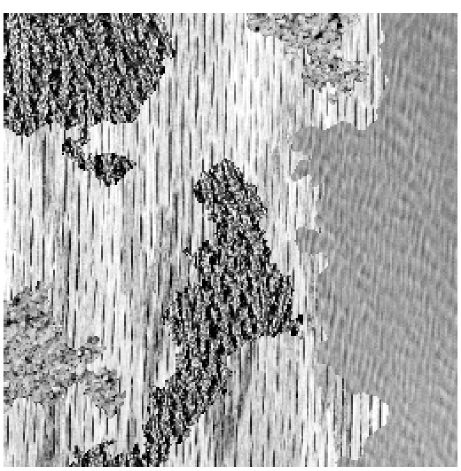

Figure 1: Original image

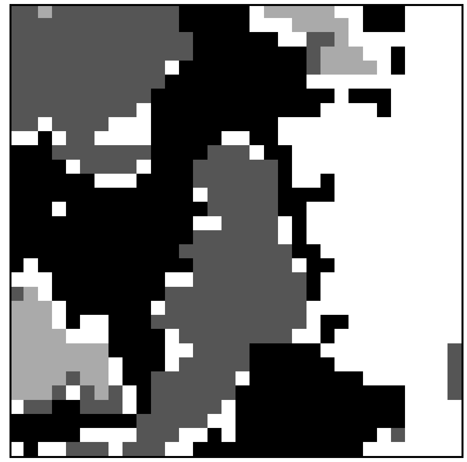

Figure 2: Pre-segmentation by SEM algorithm

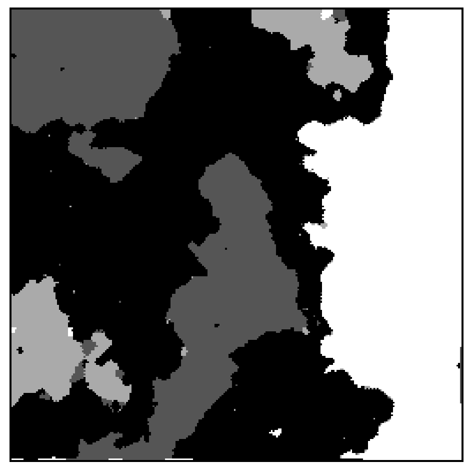

Figure 3: Final segmentation

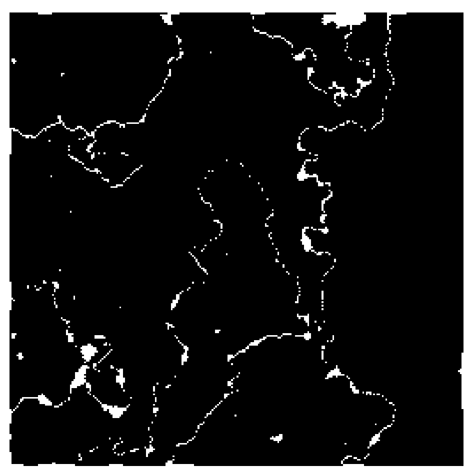

Figure 4: Error map