Correspondence

Unsupervised Texture Segmentation Using Markov Random Field Models

B. S. Manjunath and R. Chellappa

Abstract—**We consider the problem of unsupervised segmentation of textured images. The only explicit assumption made is that the intensity data can be modeled by a Gauss Markov random field (GMRF). The image is divided into number of nonoverlapping regions and the GMRF parameters are computed from each of these regions. A simple clustering method is used to merge these regions. The parameters of the model estimated from the clustered segments are then used in two different schemes, one being an approximation to the maximum *a posteriori* estimate of the labels and the other minimizing the percentage misclassification error. Our approach is contrasted with a recently published algorithm [1] which detailed an interesting simultaneous parameter estimation and segmentation scheme. We compare the results of the adaptive segmentation algorithm in [1] with a simple nearest neighbor classification scheme to show that if enough information is available, simple techniques could be used as alternatives to computationally expensive schemes.**

Index Terms—**Estimation, Gibbs distribution, Markov random field, relaxation, segmentation, simulated annealing.**

I. Introduction

Segmenting a textured scene into different classes in the absence of *a priori* information is still an unsolved issue in computer vision. The main difficulty is that the model and its parameters are unknown and need to be computed from the given image before segmentation. To compute the parameters effectively the segmented image itself is needed! Simultaneous parameter estimation and segmentation is often computationally prohibitive. An alternate approach to this problem is to have a two step process, first estimating the parameters in small regions and getting a crude segmentation. Then estimate the parameters again from this segmented image and use pixel based segmentation schemes [2], [3]. In this correspondence we assume that the texture intensity distribution can be modeled by a second order GMRF. Hence the problem is in estimating these GMRF parameters and segmenting the textures based on the estimated values.

Some of the recent work on unsupervised segmentation is reported in [1], [4], and [5]. Lakshmanan and Derin [1] in a recent paper address the problem of simultaneous estimation and segmentation of Gibbs random fields (GRF). They obtained an interesting convergence result for the maximum likelihood estimates (MLE) of the parameters and maximum *a posteriori* probability (MAP) solution for the segmentation. We give a brief description of their model in Section V and experimental results to illustrate that if one makes the same assumptions, a simple nearest neighbor classification rule produces results very close to those obtained using simulated annealing as in [1]. In [4], no specific texture model is assumed. Certain features are extracted from the sub-images and the image is segmented based on the disparity measure between the feature vectors from different subimages.

Manuscript received November 6, 1989; revised October 31, 1990. Recommended for acceptance by C. Brown. This work was supported in part by the Air Force Office of Scientific Research under Grant 86-0196.

The authors are with the Department of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089.

IEEE Log Number 9042522.

The approach to texture segmentation presented here is similar to the work of Cohen and Fan [5]. In [5] the textures are modeled as second order GMRF and the texture parameters are estimated from disjoint windows. The windows are later grouped based on clustering analysis. Finer segmentation is obtained by using the parameters from the coarse segmentation in a suitable relaxation algorithm [6].

In the next section we give a brief description of the texture model. Section III details the segmentation scheme and the experimental results are provided in Section IV. In Section V, the adaptive segmentation scheme of [1] is discussed along with the results of a simple nearest neighbor classification scheme.

II. Texture Model

The GMRF model for textures has been used by many researchers [7]. In this correspondence we consider a second order GMRF model for the conditional probability density of the intensity given the texture label.

Let Ω denote the set of grid points in the $M \times M$ lattice, i.e., $\Omega = \{(i, j),\ 1 \le i, j \le M\}$. Let $\{L_s,\ s \in \Omega\}$ and $\{Y_s,\ s \in \Omega\}$ denote the labels and zero mean gray level arrays respectively. Let N_s be the symmetric second order neighborhood of a site s (consisting of the 8 nearest neighbors of s). Then assuming that all the neighbors of s also have the same label as that of s, we can write the following expression for the conditional density of the intensity at the pixel site s:

$$P(Y_s = y_s \mid Y_r = y_r,\ r \in N_s,\ L_s = l) = \frac{e^{-U(Y_s = y_s \mid Y_r = y_r,\ r \in N_s,\ L_s = l)}}{Z(l \mid y_r,\ r \in N_s)} \quad (1)$$

where $Z(l \mid y_r,\ r \in N_s)$ is the partition function of the conditional Gibbs distribution and

$$U(Y_s = y_s \mid Y_r = y_r,\ r \in N_s,\ L_s = l) = \frac{1}{2\sigma_l^2}\left(y_s^2 - 2\sum_{r \in N_s} \Theta_{s,r}^l y_s y_r\right). \quad (2)$$

In (2), σ_l and Θ^l are the GMRF model parameters of the lth texture class. The model parameters satisfy $\Theta_{r,s}^l = \Theta_{r-s}^l = \Theta_{s-r}^l = \Theta_\tau^l$.

Further, the joint probability in a window W_s centered at s can be written as

$$P(\boldsymbol{Y}_s^* \mid L_s = l) = \frac{e^{-U_1(y_s^* \mid L_s = l)}}{Z_1(l)}$$

where $Z_1(l)$ is the partition function and

$$U_1(\boldsymbol{y}_s^* \mid L_s = l) = \frac{1}{2\sigma_l^2}\sum_{r \in W_s}\left\{y_r^2 - \sum_{\tau \in N^* \mid r+\tau \in W_s} \Theta_\tau^l y_r (y_{r+\tau} + y_{r-\tau})\right\} \quad (3)$$

$\boldsymbol{y}_s^*$ represents the intensity array in the window W_s. The above equation assumes a free boundary model. N^* is a set of shift vectors corresponding to the second order GMRF model

$$N^* = \{\tau_1, \tau_2, \tau_3, \tau_4\} = \{(0, 1), (1, 0), (1, 1), (-1, 1)\}. \quad (4)$$

A. GMRF Parameter Estimation

There are many existing methods for estimating the GMRF parameters, but none of them can guarantee both consistency (estimates converging to the true values of the parameters) and stability (the covariance matrix in the expression for the joint probability density of the MRF must be positive definite) together. Normally an optimization algorithm is used to obtain the stable estimates. The GMRF parameters are used in obtaining certain measures for segmentation and not for texture synthesis. Hence it is appropriate to use computationally less demanding schemes which can provide reasonably good estimates of these parameters for the segmentation process to work on, even if they do not necessarily guarantee the stability of the estimates. The choice of the least squares estimate [8] used here is motivated by this simplicity–stability tradeoff, and we do not check for the stability of the estimates so computed.

Consider a region of size $N \times N$ containing a single texture. Let Ω be the lattice under consideration and let Ω_I be the interior region of Ω, i.e.,

$$\Omega_I = \Omega - \Omega_B,$$
$$\Omega_B = \{s = (i,j), \quad s \in \Omega \text{ and } s \pm \tau \notin \Omega \text{ for at least some } \tau \in N^*\}. \quad (5)$$

Let

$$Q_s = [y_{s+\tau_1} + y_{s-\tau_1}, \cdots, y_{s+\tau_4} + y_{s-\tau_4}]^T. \quad (6)$$

Then the least square estimates of the parameters are

$$\hat{\Theta} = \left[\sum_{\Omega_I} Q_s Q_s^T\right]^{-1} \left[\sum_{\Omega_I} Q_s y_s\right] \quad (7)$$

$$\hat{\sigma}^2 = \frac{1}{N^2} \sum_{\Omega_I} \left[y_s - \hat{\Theta}^T Q_s\right]^2. \quad (8)$$

If $\hat{\mu}$ is the mean of the subimage, then the feature vector for the region is denoted by

$$F = (f_1, f_2, f_3, f_4, f_5, f_6) = (\theta_1, \theta_2, \theta_3, \theta_4, \hat{\mu}, \hat{\sigma}^2). \quad (9)$$

Label field: The label field is modeled as second order discrete MRF. It does not play any role in parameter estimation or in obtaining the initial coarse segmentation. The label field is characterized by a single parameter β which determines the bonding between different regions in the image.

III. Segmentation

A. Clustering

The given image is divided into a number of nonoverlapping subimages. For each of these subimages the corresponding feature vectors are estimated as described in the previous section. It is assumed that all these subimages are homogeneous. A normalized Eucledian distance measure is defined for these vectors as

$$d\left(F^i, F^j\right) = \sum_k \frac{\left(f_k^i - f_k^j\right)^2}{\left(f_k^i\right)^2 + \left(f_k^j\right)^2}. \quad (10)$$

A simple clustering is done based on this distance measure. First the maximum distance between any two regions in the image is found as

$$d_{\max} = \max_{i,j} d\left(F^i, F^j\right).$$

The regions are now grouped such that any two subimages i and j belonging to the same class satisfy

$$d\left(F^i, F^j\right) < \rho d_{\max} \quad (11)$$

where ρ is a clustering parameter. Since ρ affects the number of clusters that are formed, a good guess of ρ should be based on the knowledge about the approximate number of classes present in the image. In our experiments we used a simple heuristic $\rho = 1/(\text{approximate number of classes})$. In the above clustering process all isolated regions are marked as ambiguous. Also all regions which satisfy the criterion (11) for two different classes should be labeled ambiguous. Usually the boundary regions which have more than one texture inside fall into this class. Note that alternate schemes like k-mean clustering can also be used in obtaining such a coarse segmentation.

From these clustered regions the parameters are recomputed. These parameters are then used in pixel based segmentation algorithms [3] to obtain finer segmentation.

B. Deterministic and Stochastic Algorithms

1) Deterministic Relaxation: Assuming that the parameters of the model and the number of classes in the image are known, the texture segmentation problem can be formulated as a minimization problem. Further, for the case when the model is an MRF, mapping this problem onto a relaxation network is straightforward. The function to be minimized can be written as [3]

$$E = \frac{1}{2} \sum_s \sum_l U(s,l) V_{sl} - \frac{\beta}{2} \sum_l \sum_s \sum_{s' \in N_s} V_{s'l} V_{sl} \quad (12)$$

where N_s is the second order neighborhood of site s and $\{V_{sl}\}$ are variables taking on values from $\{0,1\}$. If V_{sl} is 1, it indicates that the site s belongs to class l. Note that for each s, only one V_{sl} has a value one and all others are zero. β represents the binding between textures of the same class and characterizes the initial distribution of the class labels. $U(s,l)$ includes all the information regarding the intensity and parameter values for the site $s \in$ class l. It gives a measure of the joint distribution of the intensities in a small window W_s centered at s and for the case when all pixels inside the window belong to class l, it is given by

$$U(s,l) = w(l) + U_1(\boldsymbol{Y}_s^*, l) \quad (13)$$

where $U_1(\cdot)$ is as in (3) and $w(l)$ is the bias corresponding to class l [3]. The $U(s,l)$'s are computed (for each s and l) before starting the relaxation with the assumption that all pixels in the window belong to the same class as l. This is an approximation, since, as the relaxation proceeds, the pixels in a window might take different labels. A computationally expensive alternative would be to compute the $U(\cdot)$'s during each iteration taking into account the label configuration existing at that time. The bias $w(l)$ can be estimated from the given data as we have a coarse initial segmentation to begin with. Before starting the relaxation, we can selectively fix the labels of the pixels from which the parameters are initially estimated, so that the relaxation process can be faster.

During each visit to site s, the class corresponding to the lowest energy E is selected. This is equivalent to setting the appropriate V_{sl} to 1. The process is repeated until there is no further change in the energy E. Since the energy is nonincreasing at each stage, the procedure is bound to converge, although the convergence in general will be to a local optimum. This algorithm is similar to the iterated conditional mode rule proposed by Besag [9].

2) Stochastic Algorithms: The alternative to deterministic relaxation is to update the class labels in a random way. Simulated annealing can be used to get the MAP solution [2]. Here we consider another criterion which minimizes the expected classification error per pixel (or alternatively, maximizes the posterior marginal distribution) and use the algorithm suggested in [10] for this. This algorithm is equivalent to running simulated annealing at a fixed

temperature $T = 1$ (i.e., no annealing) and for details we refer to [3]. The final labels chosen correspond to the most frequently selected ones. For convenience we refer to this as the MPM (maximizing the posterior marginal) algorithm in the following. We also implemented an algorithm which combines deterministic relaxation with stochastic learning [3]. This has an advantage that it requires fewer number of iterations compared to simulated annealing and the results are better than using the deterministic relaxation alone. Learning is introduced by defining a probability distribution over the class labels at each pixel site and these probabilities are updated at each convergence of the deterministic relaxation procedure. A new starting state for the relaxation is obtained by sampling from this updated probability distribution and the process is repeated. Usually about 20–40 such learning cycles are enough to get good results.

IV. Experimental Results

In the experiments described below, the subimage size was chosen to be 32×32. The value of the clustering parameter depends on the number of texture classes present and as mentioned earlier we used the heuristic $\rho = 1/$(approximate number of classes). To eliminate very small isolated regions one can use a penalty function in the relaxation algorithm which prohibits small clusters from being formed [4]. We found it convenient to use a smoothing filter (size 3×3 or 5×5) to do the same. A useful observation is that with this kind of "post-processing," the performances of both the deterministic and stochastic relaxation algorithms are comparable. The results given below correspond to those obtained after performing the smoothing. However the boundaries obtained by the stochastic algorithms are more accurate.

Example 1 (Grass and Leather Texture): Fig. 1(a) shows this 128×128 image and ρ is set to 0.5. Fig. 1(b) shows the result of coarse clustering described in Section III-A. Fig. 1(c) is the result of the deterministic relaxation. This normally takes about 10–20 iterations. The result of using learning in the deterministic relaxation is shown in Fig. 1(d). About 10 learning cycles are used in this experiment. Fig. 1(e) gives the result for the MPM algorithm after about 500 iterations. The boundary obtained by the MPM is the most accurate and also there are no misclassifications inside the homogeneous regions.

Example 2 (Grass, Raffia, and Wood): Fig. 2(a) shows the original image and Fig. 2(b) gives the coarse clustering obtained using $\rho = 0.3$. Note the presence of an ambiguous region (dark region at the top), which could not be classified into any of the other classes. The results of the various algorithms are shown in Fig. 2(c)–(e). Here again the MPM gave the best result.

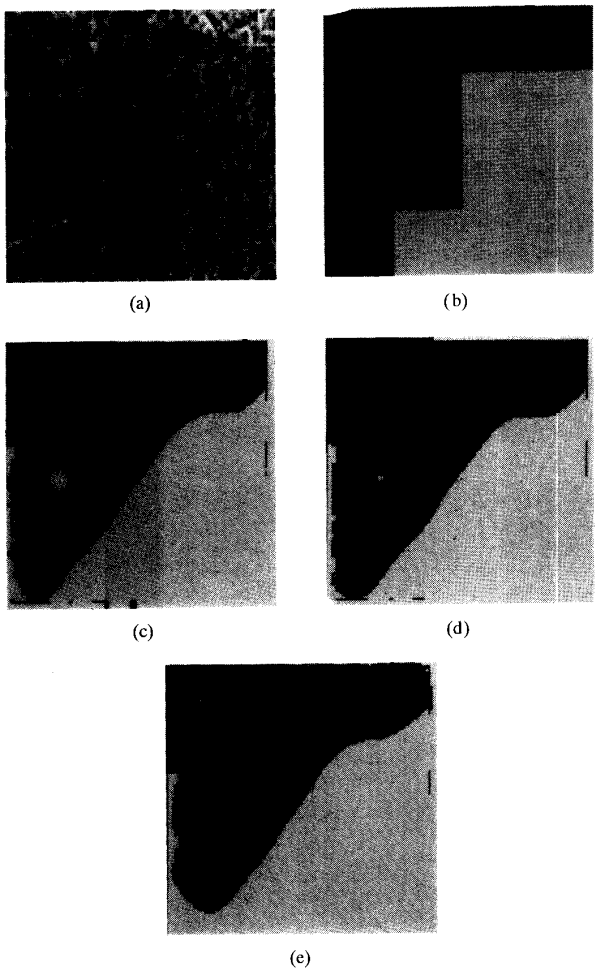

Fig. 1. Unsupervised segmentation of an image consisting of two textures (grass and leather). (a) Original image. (b) Coarse clustering. (c) Deterministic relaxation. (d) Stochastic learning. (e) MPM result.

V. Comments on an Adaptive Segmentation Algorithm

In [1], a simpler model based on GRF is used to model the intensity process. This model can be summarized as

$$Y_{ij} = X_{ij} + W_{ij} \tag{14}$$

where Y_{ij} is observed intensity at location (i, j), X_{ij} is the true intensity and W_{ij} is an independent identically distributed zero mean Gaussian noise and it is assumed that its variance is known. Further, $X_{ij} \in \{s_1, \cdots, s_N\}$, s_i being the intensity of the ith region, and N are assumed to be known. The process X is modeled as an MRF taking one of these N values. The joint distribution of X can be written as a Gibbs distribution and the particular form of this used in [1] is called a multilevel logistic (MLL) distribution. Hence the parameters correspond to this MLL distribution. A maximum likelihood estimate of the parameters of the MLL distribution are computed and combined with simulated annealing to obtain an optimum segmentation of the scene. A convergence result is also proved for this adaptive segmentation scheme.

In the analysis of the algorithm, the assumptions made play an important role. Even with these simple assumptions, due to computational difficulties further approximations have to be made. For example in the above scheme, a pseudo-likelihood algorithm is used to approximate the MLE's to avoid the computational burden involved in the estimation of MLE. The use of simulated annealing makes the algorithm computationally demanding. Further, if any of the assumptions made above (e.g., known number of regions, their intensity values or known noise parameters) are relaxed [11], the resulting convergence may not be even to a local optimum. Thus, even though the principle of simultaneous parameter estimation and segmentation could be used in more general cases like the textured images considered in this correspondence, it is not clear if it has any advantages compared to the scheme detailed in this correspondence where we first estimate the parameters from windows to obtain a coarse segmentation and then use pixel based schemes for finer segmentation.

A. A Simple Nearest-Neighbor Classification Scheme

Adaptive segmentation should be data driven, but at the same

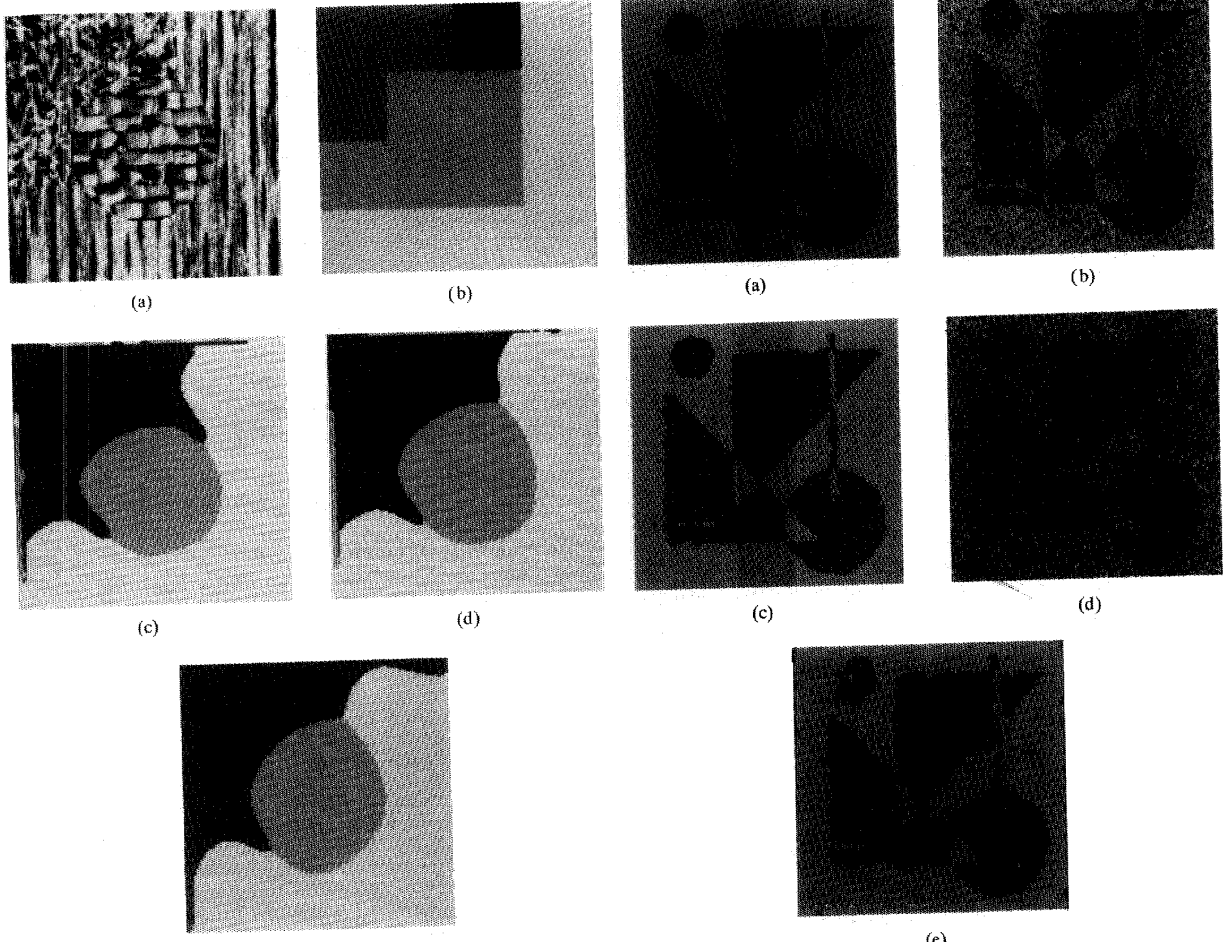

Fig. 2. Unsupervised segmentation of an image having three textures (grass, raffia, and wood). (a) Original image. (b) Coarse clustering. Note the presence of an ambiguous region (darkest region at the top). (c) Deterministic relaxation. (d) Stochastic learning. (e) MPM result.

Fig. 3. Segmentation of two region hand-drawn image using a nearest neighbor classification rule. (a) Original image. This is the same as the one used in [1]. (b) With SNR 2 (standard deviation 25). (c) Segmented image from (b). (d) With SNR 1 (standard deviation 50). (e) Segmented image from (d).

time we should make use of whatever information that is available about the data in the design of such algorithms. For example in this correspondence we made an assumption that the texture intensities could be modeled by GMRF which simplified the parameter estimation significantly. To further illustrate the usefulness of the prior knowledge about data, we give below a simple classification scheme which makes the same assumptions as in the adaptive segmentation scheme of [1] and does not need expensive algorithms like simulated annealing to obtain comparable results. The data is the same as the one used in [1]. Also the information about the noise variance is not used in this segmentation operation. For obtaining the segmented image from a noisy version of it, we used the following algorithm:

1) At each pixel site (i, j), compute the average μ_{ij} in a small window (of size 3×3 in our case) around the pixel (i, j).
2) Then the intensity of the pixel is estimated as

$$\hat{x}_{ij} = s_k = \min_l |s_l - \mu_{ij}|. \tag{15}$$

3) The resulting segmentation is processed through a smoothing filter (similar to the one described in Section IV) to eliminate small isolated regions.

Fig. 3 shows the performance of this scheme on a two region hand-drawn image. Fig. 3(a) is the original image with the two intensity levels being 100 and 150. This is one of the images used in [1]. Fig. 3(b) is the noisy version with the noise being additive i.i.d. zero mean Gaussian with standard deviation 25 (signal-to-noise ratio (SNR) of 2). The classification result we obtained is shown in Fig. 3(c) with the classification error of 1.73%. Fig. 3(d) and 3(e) show the results when the noise deviation is 50 (SNR 1).

Corresponding results for the four region case (with intensity values 100, 150, 200, and 250) are shown in Fig. 4. The maximum difference in the performance of the nearest neighbor classification rule to that reported in [1] is for the four region case with SNR 2, where we obtained an error of 2.21% compared to 0.4% reported in [1]. Table I compares the performance of this nearest neighbor classification scheme with the adaptive segmentation algorithm of [1]. As far as the computation time required, this clustering technique takes few seconds of CPU time (for the 128×128 images, on a SUN-3 workstation) compared to 15–30 minutes (on VAX 8600) reported in [1].

As can be seen from these experiments, complexity of the seg-

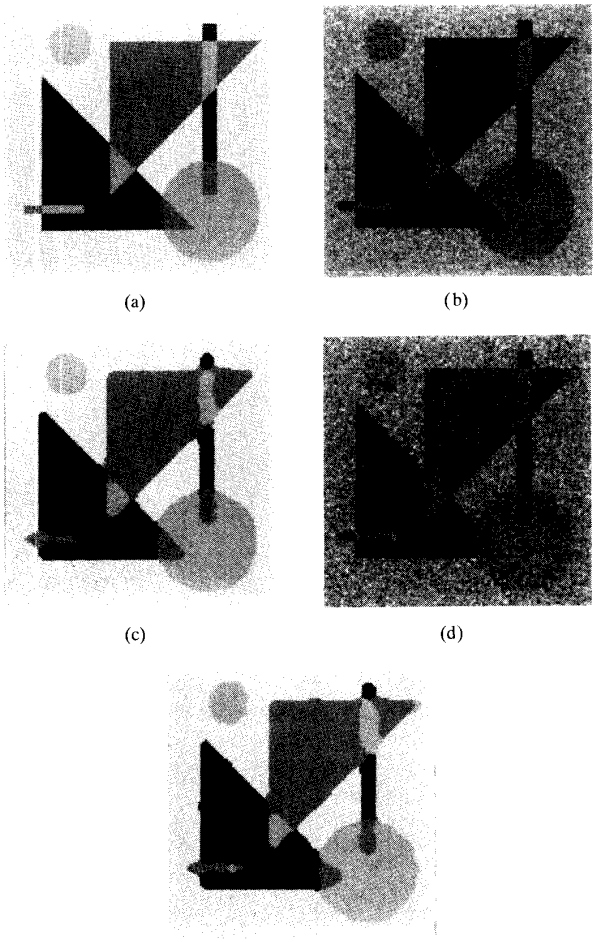

Fig. 4. Segmentation of a four region hand-drawn image using a nearest neighbor classification algorithm. (a) Original image, same as the one used in [1]. (b) With SNR 2 (standard deviation 25). (c) Segmented image from (b). (d) With SNR 1 (standard deviation 50). (e) Segmented image from (d).

mentation algorithms can be greatly reduced by a proper use of prior information about the assumed models. The texture model considered in Section II is more complicated than the one discussed in this section and the only explicit assumption made was that the textures can be modeled by a second order GMRF. Depending on the textures, this may or may not be a valid assumption. However from our experience with the different real textures like wood, wool, water, etc., this appears to be a good approximation and our experimental results also support this fact. From the computational viewpoint it is better to separate the estimation and segmentation stages. However, by doing this the algorithm will not lend itself to easy theoretical analysis.

VI. Conclusions

Unsupervised segmentation is a difficult problem. Even before estimating the parameters of any assumed model, one has to decide whether the model is applicable to a particular image or not. As we observed in the previous section, often the choice of appropriate models play a significant role. Separating estimation from segmentation simplifies the problem and enables computationally manageable algorithms. In cases where the number of textures in an image is reasonably small, we are able to estimate the model parameters and segment the scene. We also noticed that by introducing a penalty for the small regions, which is equivalent to doing simple smoothing operations, deterministic relaxation schemes give results comparable to those of stochastic techniques like simulated annealing and MPM.

TABLE I
Comparison of the Adaptive Segmentation Algorithm in [1] to the Nearest-Neighbor Classification Scheme. The Numbers Indicate Percentage Classification Errors. SNR 2 Corresponds to a Noise Standard Deviation of 25 and SNR 1 Corresponds to a Deviation of 50.

Algorithm	2 Regions		4 Regions	
	SNR 2	SNR 1	SNR 2	SNR 1
Nearest-Neighbor Classification	1.73	4.60	2.21	3.19
Adaptive Segmentation (from [1])	0.96	3.88	0.40	1.98

Acknowledgment

We would like to thank Prof. Derin of University of Massachusetts for providing us with the hand-drawn images [Figs. 3(a) and 4(a)] used in Section V.

References

[1] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 799–813, Aug. 1989.
[2] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 39–55, Jan. 1987.
[3] B.S. Manjunath, T. Simchony, and R. Chellappa, "Stochastic and deterministic networks for texture segmentation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1039–1049, June 1990.
[4] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 609–628, July 1990.
[5] F.S. Cohen and Z. Fan, "Unsupervised textured image segmentation," Dep. Elec. Eng., Univ. Rhode Island, Tech. Rep. 86-1, June 1986.
[6] F.S. Cohen and D.B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 195–219, March 1987.
[7] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition 2*, L.N. Kanal and A. Rosenfeld, Eds. Amsterdam, The Netherlands: Elsevier Science (North-Holland), 1985, pp. 79–112.
[8] R.L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial interaction models of images," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 60–72, 1983.
[9] J. Besag, "On the statistical analysis of dirty pictures," *J. Royal Statist. Soc. B*, vol. 48, pp. 259–302, 1986.
[10] J.L. Marroquin, "Probabilistic solution of inverse problems," Ph.D. dissertation, Artificial Intell. Lab., Massachusetts Inst. Technol., Cambridge, Sept. 1985.
[11] C.S. Won and H. Derin, "Unsupervised image segmentation using Markov random fields—Part I: Noisy images," preprint.