Segmentation of Gabor-Filtered Textures Using Deterministic Relaxation

P. P. Raghu and B. Yegnanarayana, *Senior Member, IEEE*

***Abstract*— A supervised texture segmentation scheme is proposed in this article. The texture features are extracted by filtering the given image using a filter bank consisting of a number of Gabor filters with different frequencies, resolutions, and orientations. The segmentation model consists of feature formation, partition, and competition processes. In the feature formation process, the texture features from the Gabor filter bank are modeled as a Gaussian distribution. The image partition is represented as a noncausal Markov random field (MRF) by means of the partition process. The competition process constrains the overall system to have a single label for each pixel. Using these three random processes, the *a posteriori* probability of each pixel label is expressed as a Gibbs distribution. The corresponding Gibbs energy function is implemented as a set of constraints on each pixel by using a neural network model based on Hopfield network. A deterministic relaxation strategy is used to evolve the minimum energy state of the network, corresponding to a maximum *a posteriori* (MAP) probability. This results in an optimal segmentation of the textured image. The performance of the scheme is demonstrated on a variety of images including images from remote sensing.**

I. Introduction

TEXTURE analysis is one of the most important techniques used in the analysis and interpretation of images consisting of repetition or quasirepetition of some fundamental image elements. Texture-based image segmentation techniques have been found useful in the analysis and interpretation of radiographic images in medicine, seismic trace images, and earth cover images obtained using remote sensing techniques [2], [33].

In spite of the importance of textures in many real and synthetic images, it is very difficult to give a universal definition of texture due to the diversity of patterns in several natural and artificial textures. We prefer to adopt the definition suggested by Sklansky [28] because of its generality. Sklansky's definition of texture is given as follows:

> *A region in an image has a constant texture if a set of local statistics or other local properties of the picture are constant, slowly varying, or approximately periodic.*

This definition explains many of the textures found in natural images. The local statistics or property that is repeated over the textured region is called a texture element or *texel*. It must be noted that texture has both local and global meaning—it is characterized by invariance of certain local attributes that are distributed over a region of an image.

Analysis of textures requires the identification of proper attributes or features that differentiate the textures in the image for segmentation, classification, and recognition. The features are assumed to be uniform within the regions containing the same texture. Various feature extraction and classification techniques have been suggested in the past for the purpose of texture analysis [15], [16] and some efficient methods have been suggested recently, for example, in [20]. More recently, the resurgence of interest in neural networks has brought into focus the neural network-based approaches to the problem of texture analysis and texture-based image segmentation [11], [32].

A significant advancement in the field of texture analysis is the introduction of Markov random field (MRF) [34] as a tool for stochastic modeling of images. MRF allows us to model the joint probability distribution of the image pixels in terms of the local spatial interactions, which can be expressed as Gibbs distribution. Geman and Geman [14] proved the equivalence of MRF and Gibbs distribution, and used the results for the problem of image reconstruction. Several studies have been made using the MRF for modeling [8] and segmentation [21] of textures. A major advantage of the stochastic modeling is that the probability distributions applicable to the image pixel intensity as well as the image partition can be implemented using highly parallel neural networks. Also, several deterministic and stochastic relaxation algorithms can be applied to these networks to perform optimal segmentation of the textured images [6].

The models based on MRF combined with the relaxation methods have been applied at the image pixel level using the pixel gray level probability and the pixel partition probability. Since texture is a contextual property described over a spatial region of appropriate resolution, methods based on the image pixel gray level alone may not work satisfactorily. Moreover, it is difficult to know *a priori* the resolution of the neighborhood of MRF when the image contains textures having large variations in texel sizes. Also, MRF assumes stationarity of the textures in the image, which cannot be assured in many real images. Thus, textures need to be characterized in a feature space which can take into account the issues of resolution and nonstationarity.

Analysis techniques have been suggested in the past to characterize the textures based on the spatial frequency distributions and the orientations of texture elements [7], [22]. A Fourier filtering method for texture analysis was proposed, which makes use of the spatial frequency distributions of the

Manuscript received September 16, 1994; revised January 12, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Patrick A. Kelly.

The authors are with the Indian Institute of Technology, Madras, India 600 036 (e-mail: yegna@iitm.ernet.in).

Publisher Item Identifier S 1057-7149(96)08515-6.

subimages by windowing the original textured image [1]. This method characterizes the spatial frequency distribution, but does not consider the information in the spatial domain. The information on the spatial frequency distribution alone may not be adequate in many situations, especially for nonstationary textures as in the case of remotely sensed images. One to has take into consideration the spatial as well as spatial frequency distribution of a textured image [18].

In 1946, Gabor proposed a set of narrowband filters that can optimally concentrate in time and frequency domains for discrete joint time-frequency signal representation [13]. This idea was extended to two dimensions for representing an image into a number of localized frequency channels measured at different spatial resolutions. The importance of the two-dimensional (2-D) Gabor filters is well recognized in the recent past as a joint spatial/spatial-frequency representation of textures [3], [30]. Analysis using Gabor filters enables us to capture the local as well as global information in a texture. The properties of Gabor filters are exploited for unsupervised texture segmentation in [19]. However, the use of image-specific knowledge together with Gabor features should produce significantly better results for texture classification.

In this paper, we explore the combination of image-specific knowledge and Gabor feature characteristics for supervised classification of textures in natural images. In particular, we describe a supervised texture segmentation and classification scheme based on the stochastic modeling of the Gabor feature space of textures and an image partition model described by a Markov random field. The texture features are extracted using a set of Gabor filters, which basically constitute a multiresolution feature extraction mechanism. The joint probability of the texture feature vector at a given pixel is represented as a Gaussian distribution. The parameters of the Gaussian model for each class are estimated using the feature vectors of the pixels in the training sites for that class. A competition process is defined, which constrains the system to have a single label for each pixel. The *a posteriori* probability of the segmentation model is derived from the feature formation, partition, and competition processes, and is expressed as the Gibbs distribution. The MAP estimation of the segmentation is obtained by using a neural network model based on the Hopfield network with a deterministic relaxation method.

The organization of the rest of the paper is as follows. Next section gives a description of the feature extraction scheme based on the multichannel Gabor filters. It includes the properties of the filters suitable for texture analysis. Section III deals with the theoretical formulation of the segmentation model by describing the stochastic processes that characterize the image features and the image pixel labeling. A neural network model to segment the textured image is described in Section IV. In Section V, experimental results on segmenting different types of textured images are described.

II. Texture Feature Extraction Using Gabor Representation of Images

The analysis of textures from natural images is difficult for several reasons. Generally, image textures will have texels of different sizes. So the optimal resolution for feature extraction cannot be specified *a priori*. One efficient way to represent the image details is by using a multiresolution representation [23], [24]. This scheme analyzes the coarse image details first and gradually increases the resolution to analyze the finer details. Further, most of the natural images contain textures having space-varying local properties such as the orientation and frequency of the texture elements. Thus, selection of an appropriate feature extraction mechanism is essential in such situations.

9	8	7	6	7	8	9
8	5	4	3	4	5	8
7	4	2	1	2	4	7
6	3	1	x	1	3	6
7	4	2	1	2	4	7
8	5	4	3	4	5	8
9	8	7	6	7	8	9

Fig. 1. Neighborhood system used for MRF model in the partition process. "x" is the pixel of interest. The numbers in the boxes show the order of neighborhood.

The feature extraction method adapted in this paper is based on the mechanism of multichannel representation of the retinal images in the biological vision system. Studies on the biological vision system have shown that several visual cortical areas of mammals contain a large number of linear and nonlinear neurons having receptive field profiles with selectivity for a variety of stimulus attributes such as location in the 2-D visual space, orientation, motion, stereoscopic depth, and spatial frequency [12]. The receptive fields of the simple cells in the early vision system possess some interesting properties that make the visual representation possible to have space-domain local feature extraction confined to narrow, spatially oriented frequency channels that are quasiindependent [5]. Daugman [9] has shown that these receptive fields can be closely approximated by 2-D Gabor filters. When appropriately tuned, these filters are found extremely useful for performing texture feature extraction and texture edge detection. Details on the biological vision system and the Gabor filter models of cortical cells can be found in [25] and [31].

A 2-D Gabor filter is an oriented complex sinusoidal grating modulated by 2-D Gaussian function, thus forming complex valued function in R^2. The function is given by

$$f(x, y, k, \theta, \sigma) = e^{-(1/2\sigma^2)(x^2+y^2)+jk(x \cos \theta + y \sin \theta)}. \quad (1)$$

The spatial extent of the Gabor function in the spatial domain is defined by σ. The orientation of the span-limited sinusoidal grating is given by θ and its radial frequency is specified as k.

Gabor-filtered output of the image is obtained by the convolution of the image with the Gabor function. If T is the textured image, then the feature value at position (x, y) of the image is given by

$$g_\lambda(x, y) = |T(x, y) * f_\lambda(x, y)| \quad (2)$$

for a filter $f_\lambda(x, y)$ with a given parameter set $\lambda = (k, \theta, \sigma)$. Here, $*$ denotes the convolution.

Gabor filters have desirable properties that are suited for nonstationary texture analysis [4]. For instance, Gabor filters achieve the lowest limit of the uncertainty inequality [10], and it is possible for them to represent the image both in spatial and spatial frequency domain optimally depending on the chosen metric [29]. Also, the Gabor filters encode the textured images into multiple narrow frequency and orientation channels. This is important since each texture in the image is characterized by a given localized spatial frequency or a narrow range of dominant localized spatial frequencies that differ significantly from dominant frequencies of other textures.

Let us assume M number of Gabor filters one for each set of values of the parameter set $\lambda = (k, \theta, \sigma)$. For a given pixel point (x, y) in the image, the M-dimensional vector $g(x, y) = [g_\lambda(x, y)]$ for all M values of λ constitutes the feature vector to characterize the pixel (x, y). In the next section, we describe how the feature vector for each pixel is modeled for the purpose of segmentation.

III. Segmentation Model

Let $\Omega = \{(i, j), 0 \leq i < I, 0 \leq j < J\}$ be the domain designating the pixel positions of a given image T_Ω. Let Y_s, $s \in \Omega$ be the random variable corresponding to the gray level value of the pixel s and can take any integer value in the range $\{0 \cdots 255\}$. The image T_Ω is characterized by a set of M-dimensional feature vectors $G = \{g_s \in \mathrm{R}^M, \forall s \in \Omega\}$, which is generated by Gabor filtering the image. Assume each g_s to be the realization of an M-dimensional random process G_s (called the *feature process*), the *a priori* probability of which is given as $P(G_s)$.

Since our segmentation scheme is a supervised one, let us assume that the image T_Ω consists of K different number of textures, so that each pixel s can take any texture label 0 to $K-1$. The corresponding texture classes are denoted by $C_0 \cdots C_{K-1}$. Also, let Ω_k, a subset of Ω, be the training site for the class C_k. The Gabor features of the training site of a given class are used to estimate the model parameters for that class. We use the notation L_s to denote the random variable describing the texture label of the pixel s and the corresponding random process is named as the *label process*. L_s is described by the *a priori* probability $P(L_s)$. The segmentation strategy is based on a hierarchical model consisting of three different random processes. We describe below each one in detail.

A. Feature Formation Process

The feature formation process describes the probability of assigning a value $g_s \in \mathrm{R}^M$ to the feature process G_s using the model parameters of each texture class for the pixel s. The pixel s can have a label $L_s = k$ corresponding to a class C_k, which is an instantiation of the label process defined by $P(L_s)$. Let us define the conditional probability of G_s given the label of the pixel L_s as

$$P(G_s = g_s | L_s = k) = \frac{e^{-E_1(G_s = g_s | L_s = k)}}{Z_1}. \tag{3}$$

Since the feature at each pixel is contributed by large number of pixels within the window function of the Gabor filter, we assume a Gaussian distribution for the feature formation process, and the energy function $E_1(.)$ takes the following form:

$$E_1(G_s = g_s | L_s = k) = \frac{\|g_s - \theta_k\|^2}{2\sigma_k^2} \tag{4}$$

Also, the normalization factor $Z_1(L_s = k)$ for this Gaussian process is

$$Z_1(L_s = k) = \sqrt{(2\pi)^M \sigma_k^2}. \tag{5}$$

The model parameters θ_k and σ_k are defined for each class C_k as

$$\theta_k = \frac{1}{S_k} \sum_{s \in \Omega_k} g_s \tag{6}$$

$$\sigma_k^2 = \frac{1}{S_k} \sum_{s \in \Omega_k} \|g_s - \theta_k\|^2 \tag{7}$$

where S_k is the cardinality of the set Ω_k.

Equation (3) can also be written as

$$P(G_s = g_s | L_s = k) = e^{-(\|g_s - \theta_k\|^2 / 2\sigma_k^2) - (1/2) \ln [(2\pi)^M \sigma_k^2]}. \tag{8}$$

B. Partition Process

The partition process defines the probability of the label of a pixel given the labels of the pixels in a predefined neighborhood of that pixel. Let N_L^p be a set of displacement vectors corresponding to a pth order noncausal symmetric neighborhood of the image pixels. The neighborhood of any pixel s is defined as the set of pixels $s + r$ for $\forall r \in N_L^p$. The operator + is defined as follows: for any pixel $s = (i, j) \in \Omega$ and any displacement $r = (k, l) \in N_L^p$, $s + r = (i + k, j + l)$.

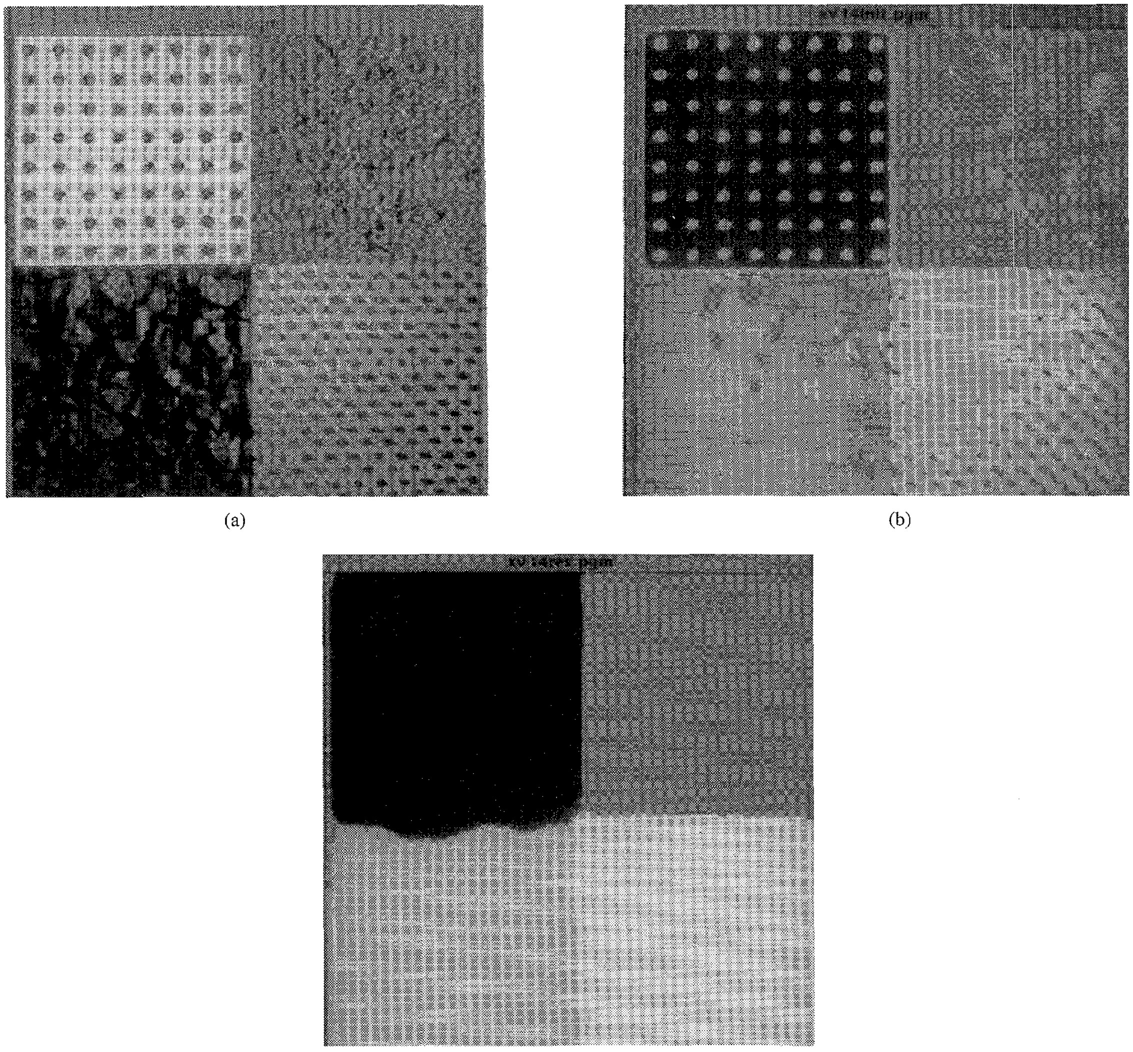
Since N_L^p is symmetric, $r \in N_L^p$ implies that $-r \in N_L^p$. Also, the set has a property that $N_L^{p-1} \subset N_L^p$. It should be noted that a neighborhood for s does not include the pixel of interest s. An example of displacement vectors for a second-order neighborhood is

$$\begin{aligned} N_L^2 = \{&(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), \\ &(1, -1), (1, 0), (1, 1)\}. \end{aligned}$$

Fig. 1 shows the structure of a typical neighborhood system.

Consider the partition process $P(L_s = k | L_{s+r}, \forall r \in N_L^p)$, which describes probability of the label of each pixel s given the labels of the pixels in a uniform pth order neighborhood. The process can be modeled as a pth order MRF model defined by

$$P(L_s = k | L_{s+r}, \forall r \in N_L^p) = \frac{e^{-E_2(L_s = k | L_{s+r}, \forall r \in N_L^p)}}{Z_2}. \tag{9}$$



(a)

(b)

(c)

Fig. 2. Segmentation of textures, textured image-1. (a) Original image containing four textures. (b) Segmentation corresponding to initial network state. (c) Final segmentation result.

The energy function E_2 can be defined as follows:

$$E_2(L_s|L_{s+r}, \forall\, r \in N_L^p) = -\sum_{\forall\, r\in N_L^p} \beta(r)\delta(L_s - L_{s+r}). \quad (10)$$

In this study, we assume that β is independent of r. So throughout this paper $\beta(r) = \beta$, where β is a positive constant. $\delta(.)$ is the Kronecker delta function.

The normalization constant Z_2 for the partition process is given as

$$Z_2 = \sum_{\forall\, L_s} e^{-E_2(L_s|L_{s+r}, \forall\, r\in N_L^p)}. \quad (11)$$

C. Competition Process

The competition process is based on the fact that any given pixel in an image can belong to only one class, and its purpose is to prevent multiple labels for any given pixel in the image. It is defined by the conditional probability of assigning a new label to an already labeled pixel.

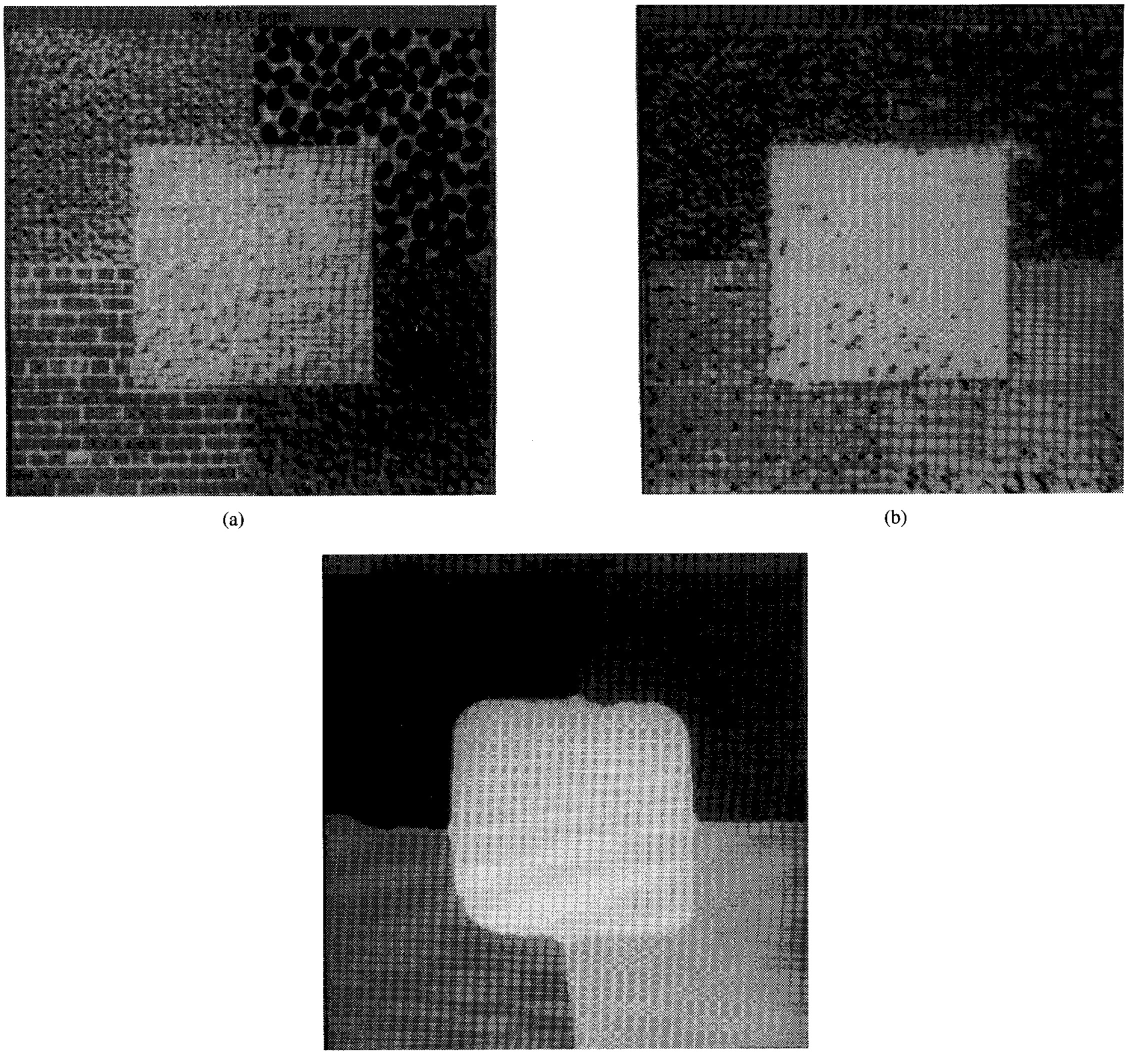
Assuming that l is the label assigned to the pixel s, let us define the probability of assigning a new label k to that pixel as

$$P(L_s = k|L_s = l) = \frac{e^{-\alpha\overline{\delta}(k-l)}}{Z_l} \quad (12)$$

where α is a positive constant and Z_l is a normalization factor. The function $\overline{\delta}$ is the inverse of Kronecker delta function given by

$$\overline{\delta}(l) = \begin{cases} 0, & \text{if } l = 0 \\ 1, & \text{otherwise.} \end{cases} \quad (13)$$

If $\tilde{L}_s$ denotes the set of labels that may be assigned to the pixel s, then the net conditional probability for all labels in the set is given by $\prod_{l\in\tilde{L}_s} P(L_s = k|L_s = l)$, which we denote



(a) (b) (c)

Fig. 3. Segmentation of textures, textured image-2. (a) Original image containing five textures. (b) Segmentation corresponding to initial network state. (c) Final segmentation result.

by $P(L_s = k|\tilde{L}_s)$. Therefore

$$P(L_s = k|\tilde{L}_s) = \prod_{l \in \tilde{L}_s} P(L_s = k|L_s = l) = \frac{e^{-E_3(L_s = k|\tilde{L}_s)}}{Z_3} \tag{14}$$

where the energy function $E_3(.)$ is

$$E_3(L_s = k|\tilde{L}_s) = \alpha \sum_{l \in \tilde{L}_s} \overline{\delta}(k - l) \tag{15}$$

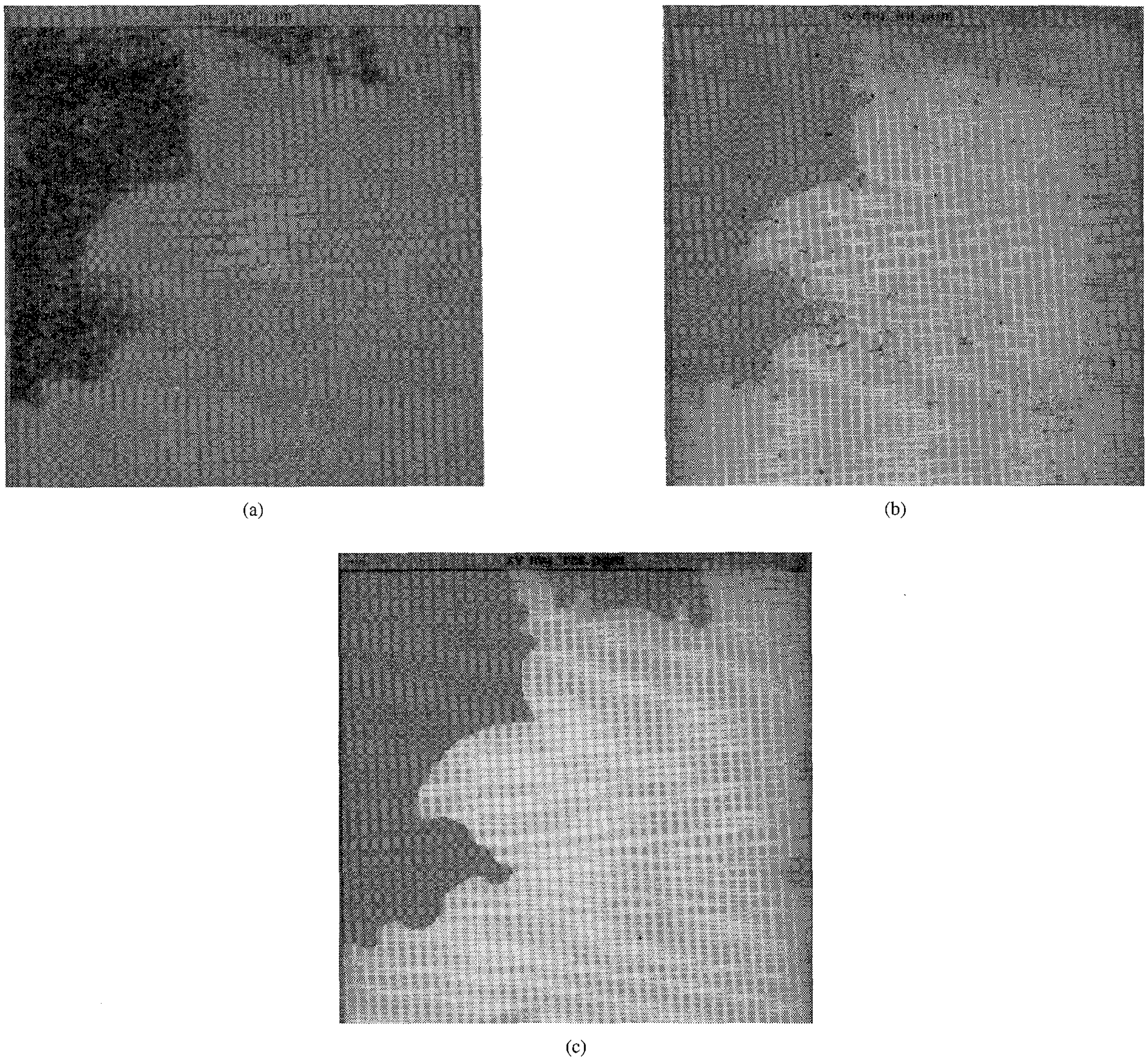
and $Z_3 = \prod_{l \in \tilde{L}_s} Z_l$, independent of s and k. The energy function E_3 is such that it reduces the probability of having another label when the pixel is already labeled.

D. A Posteriori Probability Formulation

We define the *a posteriori* probability as $P(L_s = k|G_s, L_{s+r}, \forall r \in N_L^p, \tilde{L}_s)$, which describes the labeling L_s of the pixel s given the feature measurement at the pixel s as G_s, the labels of the neighborhood pixels and the possible labels previously assigned to the pixel s.Using Bayes' theorem, it can be written as (see Appendix A)

$$P(L_s = k|G_s, L_{s+r}, \forall r \in N_L^p, \tilde{L}_s) = \frac{P(G_s|L_s = k)P(L_s = k|L_{s+r}, \forall r \in N_L^p)P(L_s = k|\tilde{L}_s)}{P(G_s)P(L_s = k)}. \tag{16}$$

Since the feature point for any pixel is fixed in the R^M space and is known *a priori*, $P(G_s)$ is considered as a constant. There are two ways to handle $P(L_s = k)$. One method is to assume that each pixel has equal probability of having any label. So, for any pixel s and any label k, $P(L_s = k) = \rho$ where ρ is a constant such that $\sum_{k=0}^{K-1} P(L_s = k) = 1$. In another method, the probability $P(L_s = k)$ is taken as 1 if $s \in \Omega_k$ and as 0 if $s \in \Omega_l$ where $l \neq k$. For all other s and k,



(a)

(b)

(c)

Fig. 4. Segmentation of textures, Magellan image. (a) Original image containing two textures. (b) Segmentation corresponding to initial network state. (c) Final segmentation result.

the probability is assumed known as in the first method. We have used the first method to characterize $P(L_s = k)$; hence, for all s and k, $P(L_s = k)$ is considered as a constant. This makes the denominator of the expression in (16) a constant value for all s and k.

The *a posteriori* probability can be expressed as a Gibbs distribution

$$P(L_s = k|G_s,\, L_{s+r},\, \forall\, r \in N_L^p,\, \tilde{L}_s) = \frac{e^{-E(L_s=k|G_s,\, L_{s+r},\, \forall\, r\in N_L^p,\, \tilde{L}_s)}}{Z} \tag{17}$$

where the energy function $E(.)$ is

$$\begin{aligned} E(L_s = k|G_s,\, L_{s+r},\, \forall\, r \in N_L^p,\, \tilde{L}_s) &= E_1(G_s|L_s = k) + \tfrac{1}{2}\ln\left[(2\pi)^M \sigma_k^2\right] \\ &\quad + E_2(L_s = k|L_{s+r},\, \forall\, r \in N_L^p) + E_3(L_s = k|\tilde{L}_s) \end{aligned} \tag{18}$$

and $Z = Z_2 Z_3 P(G_s) P(L_s = k)$ is a normalization constant independent of s and k.

Total energy of the system is obtained as

$$E^{total} = \sum_{s,\,k} E(L_s = k|G_s,\, L_{s+r},\, \forall\, r \in N_L^p,\, \tilde{L}_s). \tag{19}$$

Substituting (4), (10) and (15) in (18) and (19), the total Gibbs energy in (19) can be written as

$$E^{total} = \sum_{s,\,k}\left[\frac{\|g_s - \theta_k\|^2}{2\sigma_k^2} + \frac{1}{2}\ln\left[(2\pi)^M \sigma_k^2\right] - \sum_{\forall\, r\in N_L^p} \beta\delta(k - L_{s+r}) + \sum_{l\in \tilde{L}_s} \alpha\bar{\delta}(k - l)\right]. \tag{20}$$

The segmentation of the textured image is carried out by estimating a state configuration L_s for all pixels s, which maximizes the *a posteriori* probability given in (16). This is equivalent to finding a state such that the total Gibbs energy in (20) is minimum. To evolve such an optimal state, we have used a neural network architecture that has an energy function defined in (20) and with a deterministic relaxation mechanism.

IV. Neural Network Representation of the Energy Function

An interesting interpretation of the energy function in (20) is that it can be expressed in terms of a set of constraints defined on each pixel and can be represented on a constraint satisfaction neural network with a suitable relaxation method. Consider a neural network consisting of a three-dimensional (3-D) lattice of neurons. For an image T_Ω of size $I \times J$ with K possible labels (texture classes) for each pixel, the size of the network is $I \times J \times K$. Each neuron in the network is designated as (i, j, k), $0 < i, j, k \leq I, J, K$, where $(i, j) = s$ corresponds to the pixel position and k denotes the label index for that pixel. The network can also be viewed as having K layers of 2-D $I \times J$ arrangement of neurons. Each layer is called a label layer. For a given pixel (i, j), the corresponding neurons in the different label layers constitute a column of neurons, which we call the label column.

Any neuron (i, j, k) in the network represents a hypothesis $h(i, j, k)$ indicating the label status of the pixel (i, j). Each hypothesis is provided with an *a priori* knowledge about its truth value by means of a bias denoted by $B_{i,j,k}$ applied to the corresponding neuron. Also, each hypothesis is influenced by other hypotheses in the lattice by means of a set of constraints. Let us denote the constraint from the hypothesis $h(i, j, k)$ to the hypothesis $h(i_1, j_1, k_1)$ by means of a connection weight $W_{i,j,k;i_1,j_1,k_1}$ from the neuron (i, j, k) to (i_1, j_1, k_1). The weights considered are symmetric, i.e.,

$$W_{i,j,k;i_1,j_1,k_1} = W_{i_1,j_1,k_1;i,j,k}.$$

Let $O_{i,j,k} \in \{0, 1\}$ be the output of neuron (i, j, k), and we call the set $\{O_{i,j,k}, \forall i, j, k\}$ as the state of the network. $O_{i,j,k} = 1$ at any instant indicates that the pixel (i, j) has taken a label k at that instant. We use the notation $O_{i,j,k}(n)$ to denote the output of the neuron at nth iteration of the relaxation algorithm. The neural network described above has the energy function $E^{Hopfield}$ [17] as follows:

$$E^{Hopfield} = -\tfrac{1}{2} \sum_{i,j,k} \sum_{i_1,j_1,k_1} W_{i,j,k;i_1,j_1,k_1} \cdot O_{i,j,k} O_{i_1,j_1,k_1} - \sum_{i,j,k} B_{i,j,k} O_{i,j,k}. \quad (21)$$

Now, we describe how to determine the bias and weights of this neural network in order to represent the energy function in (20) of the segmentation model. The first term in that energy function, which is the contribution of the feature formation process, is active only if $L_s = k$, that is, if $O_{i,j,k} = 1$ where $s = (i, j)$. The instantiation k for the label random variable L_s of the pixel $s = (i, j)$ denotes the truth value $O_{i,j,k} = 1$ of the hypothesis $h(i, j, k)$. Similarly, the instantiation $L_{s+r} = k$, $s + r = (i_1, j_1)$ indicates that $O_{i_1,j_1,k} = 1$ for the hypothesis $h(i_1, j_1, k)$. So the term $\delta(L_s - L_{s+r})$ in (20) is equivalent to the product $O_{i,j,k} O_{i_1,j_1,k}$, and is active only if (i_1, j_1) is in the pth order neighborhood of (i, j). The term $\overline{\delta}(L_s - l)$ is 1 only if $L_s \neq l$. If L_s has an instantiation k, this term is equal to

$$\overline{\delta}(k - l) = \begin{cases} O_{i,j,k} O_{i,j,l} & \text{if } k \neq l \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

So the energy function in (20) is rewritten as

$$\begin{aligned} E^{total} &= \sum_{(i,j),k} \Bigg[\left\{ \frac{\|g_{(i,j)} - \theta_k\|^2}{2\sigma_k^2} + \frac{1}{2} \ln\left[(2\pi)^M \sigma_k^2\right] \right\} O_{i,j,k} \\ &\quad - \sum_{\forall (i-i_1, j-j_1) \in N_L^p} \beta O_{i,j,k} O_{i_1,j_1,k} \\ &\quad + \sum_{\forall l \neq k} \alpha O_{i,j,k} O_{i,j,l} \Bigg]. \end{aligned} \quad (23)$$

Comparing (21) and (23), the bias $B_{i,j,k}$ and the weight $W_{i,j,k;i_1,j_1,k_1}$ can be written as

$$\begin{aligned} B(i, j, k) &= -E_1(G_s = g_s | L_s = k) \\ &= -\frac{\|g_s - \theta_k\|^2}{2{\sigma_k}^2} - \frac{1}{2} \ln\left[(2\pi)^M \sigma_k^2\right] \end{aligned} \quad (24)$$

where $s = (i, j)$ and

$$W_{i,j,k;i_1,j_1,k_1} = \begin{cases} 2\beta, & \text{if } (i - i_1, j - j_1) \in N_L^p \text{ and } k = k_1 \\ -2\alpha, & \text{if } (i_1, j_1) = (i, j) \text{ and } k \neq k_1 \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

The expression for the weight shows the topology of connections in the network. Any neuron (i, j, k) at a pixel position (i, j) is having excitatory connections with strength 2β from all neurons in a neighborhood defined by the displacement vector set N_L^p in the same label layer k. The -2α term denotes the inhibitory connections from all the neurons in the label column for the pixel (i, j) in the lattice.

The state corresponding to a MAP probability is arrived using the deterministic relaxation method where the state of the network is updated iteratively by changing the output of each neuron in a deterministic manner. The state of the network is initialized using the condition

$$O_{i,j,k}(0) = \begin{cases} 1, & \text{if } k = \max\limits_l P[G_{(i,j)} = g_{(i,j)} | L_{(i,j)} = l] \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

where $P[G_{(i,j)} = g_{(i,j)} | L_{(i,j)} = l]$ is the feature formation process described in (8). This is equivalent to performing an initial segmentation using the following pixel labeling strategy without considering the partition and competition processes:

$$\text{Label of the pixel } (i, j) = \max_l P[G_{(i,j)} = g_{(i,j)} | L_{(i,j)} = l]. \quad (27)$$

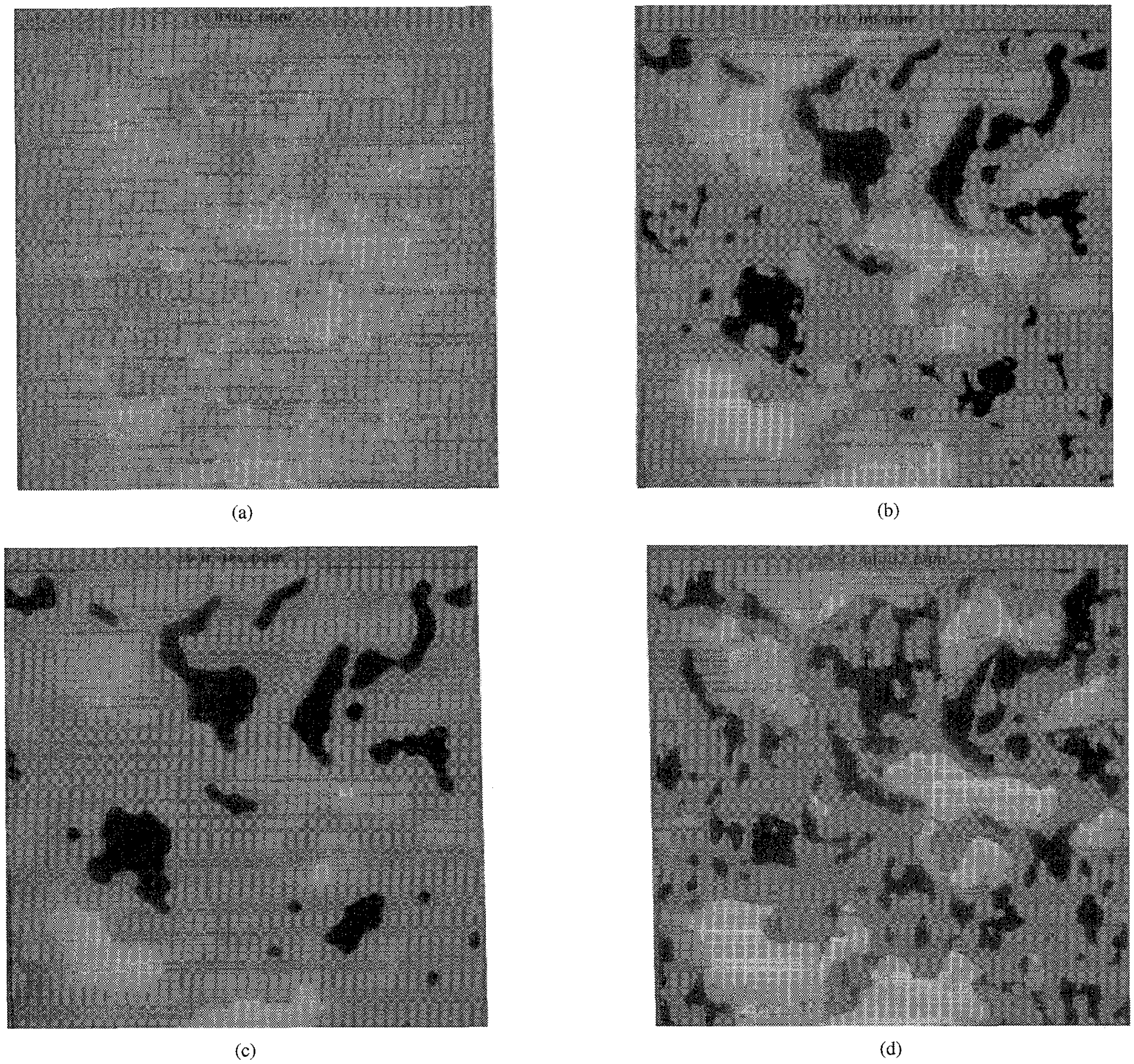


Fig. 5. Segmentation of textures, band-2 IRS image. a) Original band-2 image containing four textures. (b) Segmentation corresponding to initial network state. (c) Final segmentation result using the proposed neural network model. (d) Classification using the multilayer perceptron.

The net input $U_{i,j,k}(n)$ of a neuron (i, j, k) at nth iteration is the weighted sum of all the inputs to that neuron and is given by

$$U_{i,j,k}(n) = \sum_{i_1, j_1, k_1} W_{i,j,k;i_1,j_1,k_1} O_{i_1,j_1,k_1}(n) + B_{i,j,k}. \tag{28}$$

Assuming a threshold function in the output of each neuron, the output of the neuron at $n + 1$th iteration is given by

$$O_{i,j,k}(n+1) = \begin{cases} 1, & \text{if } U_{i,j,k}(n) > 0 \\ 0, & \text{if } U_{i,j,k}(n) < 0 \\ O_{i,j,k}(n), & \text{if } U_{i,j,k}(n) = 0. \end{cases} \tag{29}$$

It can be shown that for a change in the output of any neuron using (29), the change in energy is given as

$$\Delta E^{Hopfield} \leq 0. \tag{30}$$

This means that any change in the state of the network makes the energy decrease or remain same, but it never increases. This property is utilized by the deterministic relaxation strategy to evolve a network state corresponding to the MAP in an iterative manner.

The iteration is continued until the energy changes no more. The state of the network at this position gives the optimal state at which the *a posteriori* probability in (16) is locally maximum. The segmentation result corresponding to this network state is found out for all pixels (i, j) by using the following labeling schedule:

$$\text{Label of the pixel } (i, j) = \max_{k} U_{i,j,k} \tag{31}$$

where $U_{i,j,k}$ is the net input of the neuron (i, j, k) when the network attains the minimum energy.

Now, we discuss briefly the significance of the parameters α, β, and the order of neighborhood p. The values of α and β define the relative importance of the constraints represented by the competition and partition processes with respect to that of feature formation process. Thus, a rough estimate of these parameters can be obtained by comparing the average

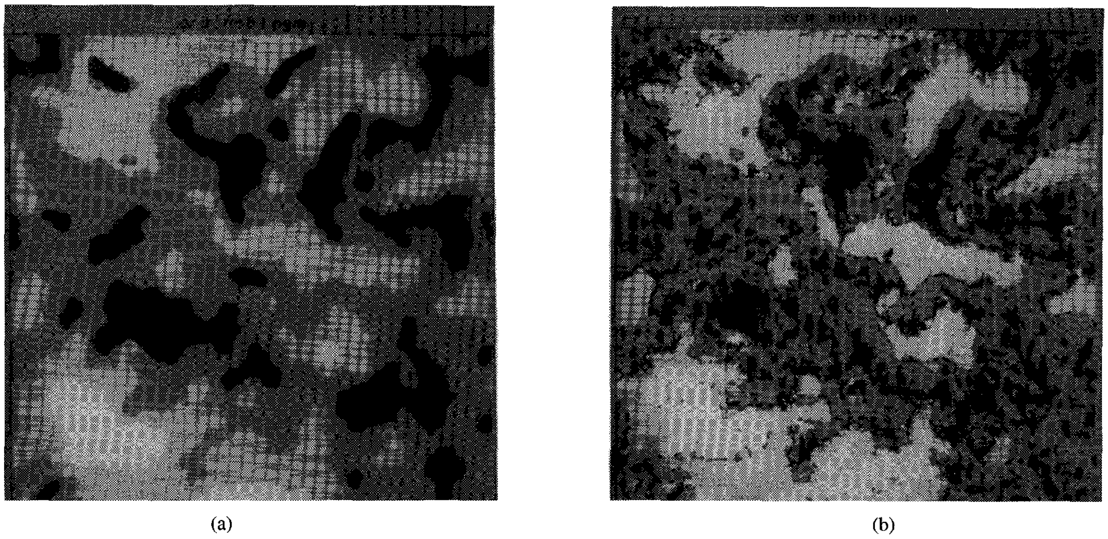


Fig. 6. Segmentation of band-3 image of the IRS data. (a) Segmentation result using the proposed neural network model. (b) Classification using the multilayer perceptron.

contributions of these random processes. After finding the rough estimates of these parameters, small changes in them do not affect the performance of the segmentation result. The order of neighborhood p defines the extent of receptive field of each neuron in a label layer. The higher the p, the higher the smoothing of the regions with same texture. But at texture boundaries, high p smears the boundary. A small p leaves misclassified regions in the output. Also, very high p is meaningless because neighborhood dependency of labels decreases as p increases.

V. Results and Discussion

A number of textured images were considered to study the performance of the proposed segmentation scheme. The images are categorized in increasing order of difficulty. The first set of images (Figs. 2(a) and 3(a)) consists of texture tiles made up of simple textures and are characterized by known texture boundaries and unknown texture models. The second category of images (Figs. 4(a) and 5(a)) contains images from remote sensing. In these images, undefined texture boundaries and unknown texture models make segmentation process difficult. The results in each case show the original image, the initial state of the network, and the final segmentation result. The initial network state is also equivalent to the result of Bayesian classification (using (27)) applied in the feature level without partition and competition processes. It should be noted that the objective in all these cases is to segment the image regions into different classes specified in advance.

Fig. 2(a) is an image of size 256 × 256 pixels consisting of tiles of four different textures, two of them are nearly deterministic (dots and diamonds), and the other two stochastic in nature (sand and pebbles). We have used 16 Gabor filters having two different bandwidths (σ = 12.5 and 6.25), two frequencies (k = 0.2 and 0.4) and four orientations (θ = 0, 45, 90, and 135°) to characterize the image, constituting a 16-dimensional feature vector for each pixel in the image. Here, a training site consisting of 1000 pixels are used for parameter estimation for each class. The initial segmentation is shown in Fig. 2(b), which is the result of applying (27) on each pixel in the image. Fig. 2(c) shows the segmentation result using the proposed scheme that yields the MAP estimate. Here, we have used a ninth-order neighborhood system for the partition process. The values for parameters α and β in this case are taken as 1.0 and 10.0, respectively.

TABLE I
Parameters of the Gabor Filters Used for Segmentation of the Image in Fig. 3(a). Total Number of Filters Used Here is 17

No. of filters	σ	k	θ in degrees
4	12.5	8	0, 45, 90 & 135
4	6.25	16	0, 45, 90 & 135
4	3.125	32	0, 45, 90 & 135
4	3.125	0.8	0, 45, 90 & 135
1	12.5	0.2	90

An image with size 256 × 256 pixels, consisting of five textures from the Brodatz album is shown in Fig. 3(a). Tiles of paper (left upper), beans (right upper), brick wall D95 (left bottom), raffia D84 (right bottom), and cork (middle) are used to generate the image. In this case, we have used 17 Gabor filters with different parameters. The selection of filters is done by visually inspecting the filtered images, so as to make out the filters that characterize different textures in the image. The filter parameters are given in Table I. The training site selected for each class consists of an area of 1000 pixels. The partition process is applied to a neighborhood of order 9. Fig. 3(b) shows the initial segmentation. The final segmentation obtained using the proposed scheme is shown in Fig. 3(c). Values for α and β are 1.0 and 5.0, respectively.

The next two images are samples from remote sensing imagery and are characterized by unknown texture models and boundaries. Fig. 4(a) shows the image of a particular region on the Venus surface, photographed by the Magellan spacecraft. The image is of size 256 × 256 and contains two textures. The features are extracted by using four Gabor filters, and a training site of 200 pixels is used for the estimation of model parameters for each class. The Gabor filter parameters are $\sigma = 0.125$, $k = 3.2$, and $\theta = 0$, 45, 90, and 135°. A fourth-order neighborhood was used to characterize the image partition. The initial and final segmentation results are shown in Fig. 4(b) and (c). We have used the values 1.0 and 0.1 for the parameters α and β, respectively.

Fig. 5(a) shows a band-2 IRS remote-sensed image of size 256 × 256 pixels. For supervised segmentation, we have identified four textures in the image. In this case, we have used 36 Gabor filters (with parameters $\sigma = 6.25$, 12.5, and 25, $k = 0.2$, 0.4, and 0.8, and $\theta = 0$, 45, 90, and 135°) to extract the textural features. A training site of 500 pixels is used for parameter estimation. A neighborhood of order 6 is used for the partition process. Fig. 5(b) shows the initial segmentation. The final segmentation result using the proposed scheme is given in Fig. 5(c). For comparison, we provide the result when a multilayer perceptron (MLP) network trained with a backpropagation algorithm [27] is used to classify the features extracted from the same Gabor filter bank. This is shown in Fig. 5(d) (this result has also been reported in [26]). The training sites used to train MLP are same as those used for estimating the parameters of the proposed method. Visual comparison of the segmentation results with the original image shows that the result from MLP breaks the image into too many smaller regions, as it is not able to capture the texture regions as well as the proposed method. The values of α and β used in the proposed network are 1.0 and 0.5, respectively.

To study the consistency of the proposed method, we have obtained the segmentation results for the same image using band-3 data also, which are shown in Fig. 6(a) and (b) using the proposed and MLP methods, respectively. The result from the proposed method (Fig. 6(a)) has similar textured regions as for the band-2 data (Fig. 5(c)), since the image is same. The differences are due to inadequacy of single band data for classification of the complex texture patterns.

VI. Conclusion

In this paper, we have presented a Bayesian approach for the supervised segmentation of textured images that is based on the Gaussian random process and MRF models. The texture features are extracted using a set of Gabor filters with different frequencies, orientations, and bandwidths, and are modeled as Gaussian distribution by means of the feature formation process. The use of Gabor filters, being a multiresolution approach, takes care of the difference in size and distribution of the texture elements. The partition process not only acts as an image partition mechanism at the texture boundaries, but also smooths the segmented image at every step of iteration. The competition process acts as a *winner-take-all* network along the label column for each pixel, suppressing the other labels of the pixel. The neural network model, which is an extension of a Hopfield network to three dimensions, finds the segmentation state using the MAP criteria. It must be noted that the final state corresponding to a minimum energy is only locally optimal, but as seen from the segmentation results, the local optimal states themselves would give a good segmentation result.

Appendix A
Derivation of *A Posteriori* Probability

Even though the *a posteriori* probability given in (16) is straightforward, we derive the expression for the sake of completeness. The *a posteriori* probability can be written as

$$\begin{aligned}&P(L_s = k|G_s,\, L_{s+r},\, \forall\, r \in N_L^p,\, \tilde{L}_s)\\ &\quad= \prod_{l \in \tilde{L}_s} P(L_s = k|G_s, L_{s+r}, \forall\, r \in N_L^p,\, L_s = l). \qquad (32)\end{aligned}$$

Throughout this discussion, L_{s+r} stands for L_{s+r}, $\forall\, r \in N_L^p$ and L_s' for $L_s = l$. Also, we temporarily use the notation L_s for the event $L_s = k$.

By Bayes' theorem

$$P(L_s|G_s,\, L_{s+r},\, L_s') = \frac{P(G_s, L_{s+r}, L_s'|L_s)P(L_s)}{P(G_s, L_{s+r}, L_s').} \qquad (33)$$

Assuming independence of the processes G_s, L_{s+r}, and L_s', we get

$$P(G_s,\, L_{s+r},\, L_s') = P(G_s)P(L_{s+r})P(L_s'). \qquad (34)$$

Also

$$P(G_s,\, L_{s+r},\, L_s'|L_s) = P(G_s|L_s,\, L_{s+r},\, L_s')P(L_{s+r},\, L_s'|L_s). \qquad (35)$$

So

$$\begin{aligned}&P(L_s|G_s,\, L_{s+r},\, L_s')\\ &\quad= \frac{P(G_s|L_s,\, L_{s+r},\, L_s')P(L_{s+r},\, L_s'|L_s)P(L_s)}{P(G_s)P(L_{s+r})P(L_s')}\\ &\quad= \frac{P(G_s|L_s,\, L_{s+r},\, L_s')P(L_s'|L_s,\, L_{s+r})P(L_{s+r}|L_s)P(L_s)}{P(G_s)P(L_{s+r})P(L_s')}. \qquad (36)\end{aligned}$$

We assume G_s to be independent of L_{s+r} and L_s', and L_s' to be independent of L_{s+r}. So the expression for $P(L_s|G_s,\, L_{s+r},\, L_s')$ becomes

$$\begin{aligned}&P(L_s|G_s,\, L_{s+r},\, L_s')\\ &\quad= \frac{P(G_s|L_s)P(L_s'|L_s)P(L_{s+r}|L_s)P(L_s)}{P(G_s)P(L_{s+r})P(L_s')}. \qquad (37)\end{aligned}$$

Expressing the $P(L_s'|L_s)$ and $P(L_{s+r}|L_s)$ as

$$P(L_s'|L_s) = \frac{P(L_s|L_s')P(L_s')}{P(L_s)} \qquad (38)$$

$$P(L_s = k|G_s, L_{s+r}, \forall r \in N_L^p, \tilde{L}_s) = \prod_{l \in \tilde{L}_s} \frac{P(G_s|L_s = k)P(L_s = k|L_{s+r}, \forall r \in N_L^p)P(L_s = k|L_s = l)}{P(G_s)P(L_s = k)} \tag{42}$$

$$= \frac{P(G_s|L_s = k)P(L_s = k|L_{s+r}, \forall r \in N_L^p) \prod_{l \in \tilde{L}_s} P(L_s = k|L_s = l)}{P(G_s)P(L_s = k)} \tag{43}$$

$$= \frac{P(G_s|L_s = k)P(L_s = k|L_{s+r}, \forall r \in N_L^p)P(L_s = k|\tilde{L}_s)}{P(G_s)P(L_s = k)} \tag{44}$$

and

$$P(L_{s+r}|L_s) = \frac{P(L_s|L_{s+r})P(L_{s+r})}{P(L_s)} \tag{39}$$

and substituting in (37), we get the expression for $P(L_s|G_s, L_{s+r}, L_s')$ as

$$P(L_s|G_s, L_{s+r}, L_s') = \frac{P(G_s|L_s)P(L_s|L_{s+r})P(L_s|L_s')}{P(G_s)P(L_s)}. \tag{40}$$

This is equivalent to

$$P(L_s = k|G_s, L_{s+r}, L_s = l) = \frac{P(G_s|L_s = k)P(L_s = k|L_{s+r})P(L_s = k|L_s = l)}{P(G_s)P(L_s = k)}. \tag{41}$$

Substituting (41) in (32), we get (42)–(44), shown at the top of the page. Equation (44) gives the expression for *a posteriori* probability as in (16).

REFERENCES

[1] R. Bajcsy, "Computer description of textured surfaces," in *Proc. Int. Joint Conf. Artificial Intell.,* Stanford, CA, Aug. 1973, pp. 20–23.

[2] D. H. Berger, "Texture as a discriminant of crops on radar imagery," *IEEE Trans. Geosci. Electr.,* vol. GE-8, no. 4, pp. 344–348, Oct. 1970.

[3] A. C. Bovik, M. Clark, and W. S. Geisler, "Computational texture analysis using localized spatial filtering," in *Proc. IEEE Comp. Soc. Workshop Comput. Vision,* Dec. 1987.

[4] ——, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 12, no. 1, pp. 55–73, Jan. 1990.

[5] F. W. Campbell and J. G. Robson, "Application of Fourier analysis of the visibility of gratings," *J. Phys., London,* vol. 197, pp. 551–556, 1968.

[6] R. Chellappa, B. S. Manjunath, and T. Simchony, "Texture segmentation with neural networks," in *Neural Networks for Signal Proc.,* Bart Kosko, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[7] J. M. Coggins and A. K. Jain, "A spatial filtering approach to texture analysis," *Pattern Recog. Lett.,* vol. 3, pp. 195–203, 1985.

[8] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-5, no. 1, pp. 25–39, Jan. 1983.

[9] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Res.,* vol. 20, pp. 847–856, 1980.

[10] ——, "Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. (A),* vol. 2, no. 7, pp. 1160–1169, July 1985.

[11] F. A. DeCosta and M. F. Chouikha, "Neural network recognition of textured images using third order cumulants as functional links," in *Proc. Int. Conf. Acoust., Speech, Signal Proc.,* San Francisco, CA, 1992, vol. 4, pp. 77–80.

[12] D. Van Essen, "Hierarchical organization and functional streams in the visual cortex," *Ann. Rev. Neurosci.,* vol. 2, pp. 227–263, 1979.

[13] D. Gabor, "Theory of communication," *J. IEE,* vol. 93, pp. 429–459, 1946.

[14] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-6, pp. 721–741, 1984.

[15] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE,* vol. 67, no. 5, pp. 786–804, May 1979.

[16] R. M. Haralick, K. S. Shanmugam, and I. Distein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-3, 1973.

[17] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Acad. Sci.,* Apr. 1982, vol. 79, pp. 2554–2558.

[18] L. D. Jacobson and H. Wechsler, "Joint spatial/spatial frequency representations," *Signal Processing,* vol. 14, no. 1, pp. 37–68, Jan. 1988.

[19] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recog.,* vol. 24, pp. 1167–1186, 1991.

[20] J. M. Keller, S. Chen, and R. M. Crownover, "Texture description and segmentation through fractal geometry," *Comput. Vision Graphics Image Processing,* vol. 45, pp. 150–166, 1989.

[21] 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, no. 8, pp. 799–813, Aug. 1989.

[22] K. I. Laws, "Textured image segmentation," Tech. Rep. USCIPI Rep. 940, Image Processing Inst., Univ. Southern Calif., Los Angeles, CA, 1980.

[23] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, no. 12, pp. 2091–2110, Dec. 1989.

[24] ——, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 11, no. 7, pp. 674–693, July 1989.

[25] D. A. Pollen and S. F. Ronner, "Visual cortical neurons as localized spatial frequency filters," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-13, no. 5, Sept. 1983.

[26] P. P. Raghu, R. Poongodi, and B. Yegnanarayana, "A combined neural network approach for texture classification," *Neural Networks,* vol. 8, no. 6, pp. 975–987, 1995.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. William, "Learning internal representations by error propagations," in *Parallel Distributed Processing,* J. L. McClelland and D. E. Rumelhart, Eds. Cambridge, MA: MIT Press, 1986, vol. 1, ch. 8, pp. 318–362.

[28] J. Sklansky, "Image segmentation and feature extraction," *IEEE Trans. System, Man, Cybern.,* vol. 8, pp. 237–247, 1978.

[29] D. G. Stork and H. R. Wilson, "Do Gabor functions provide appropriate descriptions of visual cortical receptive fields?," *J. Opt. Soc. Amer. A,* vol. 7, no. 8, pp. 1362–1373, Aug. 1990.

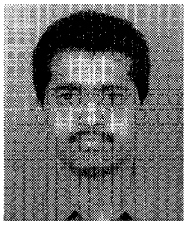
[30] M. R. Turner, "Texture discrimination by Gabor functions," *Biol. Cybern.,* vol. 55, pp. 71–82, 1986.

[31] R. L. De Valois, D. G. Albrecht, and L. G. Thorell, "Spatial frequency selectivity of cells in macaque visual cortex," *Vision Res.,* vol. 22, pp. 545–559, 1982.

[32] A. Visa, "A texture classifier based on neural network principles," in *Proc. Int. Joint Conf. Neural Networks,* San Diego, CA, vol. 1, June 1990, pp. 491–496.

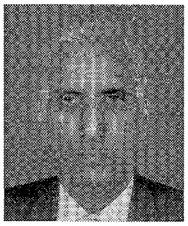
[33] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-6, pp. 269–285, 1976.

[34] J. W. Woods, "Two-dimensional discrete Markov random fields," *IEEE Trans. Inform. Theory,* vol. IT-18, pp. 232–240, Mar. 1972.



P. P. Raghu was born in Kerala, India, in 1967. He received the B.Tech degree in electrical engineering from Calicut University, Kerala, and M. Tech in computer and information sciences from Cochin University of Science and Technology, Kerala, in 1989 and 1991, respectively.

From 1991 to 1995, he was a Senior Project Officer and pursued the Ph.D. at the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India. In 1996, he joined Central Research Laboratory, Bangalore, India, as a member of the Research Staff. His research interests include image processing, artificial neural networks, and pattern recognition.



B. Yegnanarayana (M'78–SM'84) was born in India on January 9, 1944. He received the B.E., M.E., and Ph.D. degrees in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1964, 1966, and 1974, respectively.

He was a Lecturer from 1966 to 1974 and an Assistant Professor from 1974 to 1978 in the Department of Electrical Communication Engineering, Indian Institute of Science. From 1977 to 1980, he was a Visiting Associate Professor of Computer Science at Carnegie-Mellon University, Pittsburgh, PA. He was a visiting scientist at ISRO Satellite Center, Bangalore, from July to December 1980. Since 1980, he has been a Professor in the Department of Computer Science and Engineering, Indian Institute of Technology, Madras. He was a Visiting Professor at the Institute for Perception Research, Eindhoven Technical University, Eindhoven, The Netherlands, from July 1994 to January 1995. From 1966 to 1971, he was engaged in the development of evnironmental test facilities for the Acoustics Laboratory at the Indian Institute of Science. Since 1972, he has been working on problems in the area of speech signal processing. He is presently engaged in research activities in digital signal processing, speech recognition, and neural networks.

Dr. Yegnanarayana is a fellow of the Indian National Academy of Engineering.