

Unsupervised Texture Classification Using Vector Quantization and Deterministic Relaxation Neural Network

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

Abstract—This paper describes the use of a neural network architecture for classifying textured images in an unsupervised manner using image-specific constraints. The texture features are extracted by using two-dimensional (2-D) Gabor filters arranged as a set of wavelet bases. The classification model comprises feature quantization, partition, and competition processes. The feature quantization process uses a vector quantizer to quantize the features into codevectors, where the probability of grouping the vectors is modeled as Gibbs distribution. A set of label constraints for each pixel in the image are provided by the partition and competition processes. An energy function corresponding to the *a posteriori* probability is derived from these processes, and a neural network is used to represent this energy function. The state of the network and the codevectors of the vector quantizer are iteratively adjusted using a deterministic relaxation procedure until a stable state is reached. The final equilibrium state of the vector quantizer gives a classification of the textured image. A cluster validity measure based on modified Hubert index is used to determine the optimal number of texture classes in the image.

Index Terms—Deterministic relaxation, Gabor filter, Hopfield model, image analysis, neural networks, remote sensing, texture classification, unsupervised classification, vector quantization.

I. INTRODUCTION

CLASSIFICATION of textured images in an unsupervised manner is useful to delineate different regions present in many natural images. Texture can be formally defined [1] as follows: *A region in an image is a texture if a set of local statistics or other local properties of the image are constant, slowly varying, or approximately periodic.* The local statistic or property that is repeated over a textured region can be viewed as a visual primitive with certain invariant properties. This is called *texture element* or *texel*, which repeats in each textured region in a given area.

Classification methods based on supervised classification assume that the number and types of textures in the image are known *a priori*. This needs a set of training sites in the image to extract features relevant for each class. On the other hand, methods that perform unsupervised classification of textures do not make use of assumption regarding the number of textures in the image [2]. Unsupervised classification is important in

the case of natural images, especially in the domain of remote sensing, because of the difficulty in determining the number and types of textures in a given image.

In this paper, we describe a method for unsupervised classification of image textures based on vector quantization (VQ) [3] and deterministic relaxation techniques. The texture features are extracted using a set of Gabor filters arranged as a set of wavelet bases [4]. Gabor filtering is a multiresolution scheme that provides spatial and spatial frequency representation of images. We define a feature quantization process that uses VQ principles to encode the Gabor features in the feature space. In general, VQ is a process for approximating a set of input vectors into a finite number of codevectors [3]. It divides the given feature space into number of groups or clusters, and computes iteratively the centroids of the clusters (codevectors) in an unsupervised fashion. The use of VQ in image classification is well known in the literature [5], [6]. Two other random processes, namely partition process and competition process, are used to express a set of label constraints on the image pixels. The *a posteriori* probability, derived from these three processes, is expressed as Gibbs distribution, and the corresponding energy is used to derive a constraint satisfaction neural network model. A deterministic relaxation strategy is applied on the network to determine a state corresponding to the maximum *a posteriori* probability. The resulting clusters are interpreted as different texture classes in the image. A cluster validity measure based on modified Hubert index is used to decide the number of texture classes in the image. A similar work was reported in [7], but it was for supervised texture classification. The work reported in this paper is an extension of this earlier work to an unsupervised classification scheme.

Unlike other unsupervised methods [8], which depend only on the feature vector of each pixel to label it, the proposed method uses not only the feature-dependent knowledge but also a number of image-specific constraints as an additional information. This should help to improve the performance of the texture classification scheme significantly.

Some notations used in the proposed classification model are as follows: Let the textured image T_Ω be designated by a domain $\Omega = \{(i, j), 0 \leq i < I, 0 \leq j < J\}$ of pixel positions. A set of M -dimensional feature vectors $\{g_s \in \mathbb{R}^M, \forall s \in \Omega\}$, generated by Gabor filtering, is used to characterize the textured image T_Ω . Each g_s can be considered to be the realization of an M -dimensional random process G_s . We call

Manuscript received November 13, 1995; revised October 14, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. William E. Higgins.

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

Publisher Item Identifier S 1057-7149(97)07120-0.

this the *feature process*, described by the *a priori* probability distribution $P(G_s)$. We denote the random variable describing the texture label of the pixel s by L_s . We assume that L_s can take any value from the set of labels $\{0, 1, \dots, K-1\}$ where K is the number of texture classes. The label random process is described by the *a priori* probability distribution $P(L_s)$.

The paper is organized as follows. The procedure for feature extraction using Gabor filters is reviewed in Section II. We present our classification model in Section III, which involves feature quantization, partition, and competition random processes. In Section IV, a neural network representation of the model is presented. The section also includes a discussion on the relaxation algorithm and the codevector adjustment. The cluster validation measure used in our experiments is described in Section V. Finally, in Section VI, the classification performance for a number of textured images is discussed.

II. FEATURE EXTRACTION USING GABOR FILTERS

Classification of natural textures is difficult for several reasons. In an image containing a number of different textures, texture elements may have different sizes and shapes, making it difficult to determine *a priori* the resolution needed for analysis of the textures. This necessitates the use of multiresolution methods in the extraction of texture features. Also, most of the natural textures are nonstationary. They may have space-varying local properties such as orientation, frequency, and size of the texture elements. Spatial filtering approaches [9], [10] characterize the spatial frequency content, but they do not capture the variations in the spatial domain. The spatial frequency information alone may not be adequate when the textures are nonstationary. One has to take into account the spatial as well as spatial frequency characteristics of the textured image.

Thus, it is necessary to extract appropriate features to differentiate textures of different types in the image for classification. For our experiments, we have adapted a feature extraction method that resembles the mechanism of multi-channel representation of the retinal images in the biological vision system [11]. It has been shown that the receptive fields of simple cells in the early vision system can perform space-domain local feature extraction confined to narrow, spatially oriented frequency channels that are quasi-independent in nature [12]. Daugman has shown that these receptive fields can be closely approximated by the 2-D Gabor filters [13]. The decomposition using Gabor filters closely resembles the mechanism of multichannel representation of the retinal images in the biological vision system [14]. When properly tuned, these filters are useful for extracting the texture features [15], [4] and for detecting texture edges [16].

A 2-D Gabor filter is an oriented complex sinusoidal grating modulated by a two-dimensional (2-D) Gaussian function. The expression for the impulse response of a 2-D Gabor filter is given as

$$f(x, y, k_x, k_y, \sigma) = e^{-(1/2\sigma^2)(x^2+y^2)+j(k_x x+k_y y)}. \quad (1)$$

The width of the Gaussian in the Gabor function is defined by σ . The orientation of the span limited sinusoidal grating is given by $\tan^{-1}(k_y/k_x)$, and its frequency is specified along the x and y coordinates by k_x and k_y , respectively.

The assumption in texture processing using the Gabor filters is that each texture is characterized by a given localized spatial frequency or a narrow range of dominant localized spatial frequencies that differ significantly from the dominant frequencies of other textures. The frequency and orientation of the complex sinusoid in the Gabor filter describe the local structure of the texture in that frequency channel, while the Gaussian envelope defines the resolution with which the texture structure is characterized. Thus, a set of properly selected Gabor filters allows us to obtain information regardless of differences in dominant sizes, orientations and distributions of the texture elements.

An important property of the 2-D Gabor filters is their ability to represent the image both in spatial and spatial frequency domains optimally by achieving the theoretical lower bound of joint uncertainty [17], depending upon the chosen metric [18]. This is equivalent to achieving maximum possible joint resolution in the two domains [19], providing simultaneous spectral and spatial localization. The localization, in the context of texture segmentation, defines the ability of the segmentation strategy to accurately distinguish emergent texture frequencies and to accurately mark the boundaries separating the textures [20]. Since Gabor filters reduce the uncertainty to a minimum level, they are able to segment the textures by discriminating narrow range of spatial frequencies.

One attractive feature of the Gabor filters is their orientation selectivity. This will be clear when the expression (1) is rewritten in polar coordinates as

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

where $\theta = \tan^{-1}(k_y/k_x)$ is the orientation of the Gabor filter, and $k = \sqrt{k_x^2 + k_y^2}$ is the radial frequency. The orientation selectivity of the Gabor filters allows us to discriminate textures having different orientations.

For our experiments, Gabor filters with varying support (in spatial as well as spatial frequency domains) were used for feature extraction, in order to detect and localize texture features at different scales. This leads to the selection of filters similar to wavelet decomposition where the mother wavelet is a 2-D Gabor function. This is expressed as

$$f(x, y, a, k, \theta, \sigma) = e^{-(a^2/2\sigma^2)(x^2+y^2)+jak(x \cos \theta+y \sin \theta)} \quad (3)$$

where a is the wavelet scale factor. A set of filters $\{f(x, y, a, k, \theta, \sigma)\}_{a, \theta}$ for a given k and σ forms the Gabor wavelet family. This set of filters provides a number of scaled and rotated versions of the mother wavelet. Usually, the values of a is chosen as $a = 2^\gamma$, where γ is an integer.

The feature vector at each spatial location (i, j) is specified as

$$g(i, j) = \{g_{(i, j)}(a, k, \theta, \sigma)\}_{a, \theta} \quad (4)$$

where

$$\begin{aligned} g_{(i,j)}(a, k, \theta, \sigma) &= \left\| \iint T_{\Omega}(x, y) f(i-x, j-y, a, k, \theta, \sigma) dx dy \right\|^2 \\ &= \|T_{\Omega}(i, j) * f(i, j, a, k, \theta, \sigma)\|^2 \end{aligned} \quad (5)$$

where “*” denotes a 2-D convolution operator. For a given k and σ , and for a given set of values of a and θ , let us assume M Gabor filters are used for feature extraction. The M -dimensional vector $g_{(i,j)}$ then constitutes the feature vector to characterize the pixel (i, j) in the image.

For implementation of Gabor filtering, we consider the following discrete version of the expression in (5):

$$g_{(i,j)}(a, k, \theta, \sigma) = \left\| \sum_{x=0}^{I-1} \sum_{y=0}^{J-1} T_{\Omega}(x, y) f(i-x, j-y, a, k, \theta, \sigma) \right\|^2 \quad (6)$$

The values of $g_{(i,j)}(a, k, \theta, \sigma)$ for all pixels (i, j) are computed by employing the circular convolutions implemented using forward and inverse discrete Fourier transforms.

In the next section, we describe the stochastic modeling of the feature vectors and the related image-specific constraints for developing a model for unsupervised classification.

III. CLASSIFICATION MODEL

The classification model is defined by the *a posteriori* probability of the label of a pixel given the feature vector and the label constraints on the pixel. Maximization of this probability is used to obtain a classification of the image. We have used three random processes to formulate these image constraints. Each one is described below.

A. Feature Quantization Process

The feature quantization process describes the division of the M -dimensional feature space into a set of disjoint regions $\{R_k, 0 \leq k < K\}$ using principles of VQ. A vector quantizer is a mapping of the Euclidean M -dimensional space \mathbb{R}^M onto a finite set $C = \{\mu_k, 0 \leq k < K\}$ of points in \mathbb{R}^M . The set C is called a *codebook*. Each element $\mu_k \in C$ is called a *codevector*, representing a region R_k in \mathbb{R}^M .

VQ can be viewed as a multidimensional optimization problem in which the overall distance between the source vectors and the codevectors should be minimized [21]. A distance function $d(g_s, \mu_k)$ is defined as the cost of associating the feature vector g_s with the region R_k . Two necessary conditions for optimal VQ are the nearest neighbor condition for the division of the feature space into regions and the centroid condition for estimation of codevectors [22]. The former ensures that each feature vector g_s is mapped to the nearest codevector, thus defining a region $R_k = \{g_s \in \mathbb{R}^M | d(g_s, \mu_k) < d(g_s, \mu_l), \forall l \neq k\}$. The centroid condition defines each μ_k as the centroid of the vectors in the region R_k . Hence, $\mu_k = (1/|R_k|) \sum_{g_s \in R_k} g_s$, where $|R_k|$ is the number of patterns in the region R_k .

The state S of a VQ describes the set of labels assigned to all the pixels at a given instance. That is, $S = \{\sigma_1, \sigma_2, \dots, \sigma_{|\Omega|}\}$, where $\sigma_i = k$, is the label of the pixel i and k is any one of the labels in the range zero to $K-1$. $|\Omega|$ is the cardinality of the set Ω , and is equal to the number of pixels in the image. The assumption here is that the next state differs from the current state in the label for any one pixel chosen at random. Hence, the next state is given by the product of the probabilities of the current state and the transition probability from the current state to the next state. Assuming that the transition probabilities are independent of the state of the VQ, the stationary probability distribution of the state of VQ is given by the following Gibbs distribution [21]:

$$P(S) = \frac{1}{Z_q} e^{-D(S)} \quad (7)$$

where $D(S) = \sum_k \sum_{g \in R_k} d(g, \mu_k)$ is the average distance and Z_q is a normalization constant so that $\sum_{S} P(S) = 1$. It is clear from (7) that the probability of being in a state S is related to the average distance of the feature vectors of the pixels from the corresponding cluster centers. Smaller average distance indicates better performance from VQ and, hence, the probability of the corresponding state will be high.

The probability $P(g_s \in R_k)$ of assigning a label k to a feature vector [23] g_s is given by

$$P(g_s \in R_k) = \frac{e^{-d(g_s, \mu_k)}}{Z_f} \quad (8)$$

where Z_f is the normalization constant given by $\sum_l e^{-d(g_s, \mu_l)}$. It is obtained by setting $\sum_l P(g_s \in R_l) = 1$. This probability is equivalent to the joint probability of assigning a feature value g_s and a label k for the pixel s . That is, $P(g_s \in R_k) = P(G_s = g_s, L_s = k)$. Let $P(G_s = g_s | L_s = k)$ be the conditional probability that the feature value for the pixel s is g_s given that the label for s is k . Then we have

$$P(G_s = g_s | L_s = k) = \frac{P(g_s \in R_k)}{P(L_s = k)} \quad (9)$$

$$= \frac{e^{-d(g_s, \mu_k)}}{Z_f P(L_s = k)}. \quad (10)$$

The random process defined by the conditional probability $P(G_s = g_s | L_s = k)$ is called *feature quantization* process.

The choice of Euclidean distance $\|g_s - \mu_k\|^2$ for the distance function $d(g_s, \mu_k)$ makes the expression (8) a Gaussian distribution with unit variance. Hence, the normalization factor Z_f becomes independent of s and k .

B. Partition Process

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

$r = (k, l) \in N^p$. The set has a property that $N^{p-1} \subset N^p$. A neighborhood for s does not include the pixel of interest s .

The partition process $P(L_s|L_{s+r}, \forall r \in N^p)$ can be described in terms of Gibbs distribution as follows:

$$P(L_s|L_{s+r}, \forall r \in N^p) = \frac{e^{-E_p(L_s|L_{s+r}, \forall r \in N^p)}}{Z_p} \quad (11)$$

where the energy function E_p is defined such that the pixel s has high probability of having the label similar to that of the neighborhood pixels. This can be achieved by selecting the following energy function:

$$E_p(L_s|L_{s+r}, \forall r \in N^p) = - \sum_{\forall r \in N^p} \beta \delta(L_s - L_{s+r}) \quad (12)$$

where β is a positive constant. $\delta(\cdot)$ is the Kronecker delta function defined as

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

The normalization constant Z_p for the partition process is given as

$$Z_p = \sum_{\forall L_s} e^{-E_p(L_s|L_{s+r}, \forall r \in N^p)} \quad (14)$$

and it is independent of s and L_s . The importance of the partition process is that it acts as a mechanism for partitioning an image into its texture regions. It also provides a smoothing effect on pixel labels at each step of relaxation.

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

$$P(L_s = k|L_s = l) = \frac{e^{-\alpha \bar{\delta}(k-l)}}{Z_l} \quad (15)$$

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

$$\bar{\delta}(l) = \begin{cases} 0, & \text{if } l = 0 \\ 1, & \text{otherwise.} \end{cases} \quad (16)$$

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

$$\begin{aligned} P(L_s = k|\tilde{L}_s) &= \prod_{l \in \tilde{L}_s} P(L_s = k|L_s = l) \\ &= \frac{e^{-E_c(L_s=k|\tilde{L}_s)}}{Z_c} \end{aligned} \quad (17)$$

where the energy function $E_c(\cdot)$ is

$$E_c(L_s = k|\tilde{L}_s) = \alpha \sum_{l \in \tilde{L}_s} \bar{\delta}(k-l) \quad (18)$$

and $Z_c = \prod_{l \in \tilde{L}_s} Z_l$, independent of s and k . The energy function E_c is such that it reduces the probability of having another label when the pixel is already labeled. The competition process acts as a *winner-take-all* mechanism that controls the labeling of each pixel by shutting off other possible labels of that pixel.

D. Derivation of Classification Model

The objective in classification is to determine the label of a pixel subjected to the constraints provided by feature quantization process, partition process, and competition process. This is achieved by maximizing the *a posteriori* probability that the label L_s of the pixel s is k , given the feature vector of the pixel, the labels of the neighborhood pixels and the possible labels that can be assigned to the pixel s . This *a posteriori* probability is given by the conditional probability $P(L_s = k|G_s = g_s, L_{s+r}, \forall r \in N^p, \tilde{L}_s)$. Applying Bayes theorem [7]

$$\begin{aligned} P(L_s = k|G_s, L_{s+r}, \forall r \in N^p, \tilde{L}_s) &= \frac{P(G_s|L_s = k)P(L_s|L_{s+r}, \forall r \in N^p)P(L_s = k|\tilde{L}_s)}{P(G_s)P(L_s = k)}. \end{aligned} \quad (19)$$

By substituting (10), (11), and (17) in (19), we can write

$$\begin{aligned} P(L_s = k|G_s, L_{s+r}, \forall r \in N^p, \tilde{L}_s) &= \frac{e^{-d(g_s, \mu_k)} e^{-E_p(L_s|L_{s+r}, \forall r \in N^p)} e^{-E_c(L_s=k|\tilde{L}_s)}}{Z_f Z_p Z_c P(G_s) P(L_s = k)^2} \end{aligned} \quad (20)$$

$$= \frac{e^{-[d(g_s, \mu_k) + E_p(L_s|L_{s+r}, \forall r \in N^p) + E_c(L_s=k|\tilde{L}_s)]}}{Z_f Z_p Z_c P(G_s) P(L_s = k)^2} \quad (21)$$

$$= \frac{e^{-[d(g_s, \mu_k) - \sum_{\forall r \in N^p} \beta \delta(k - L_{s+r}) + \sum_{l \in \tilde{L}_s} \alpha \bar{\delta}(k-l)]}}{Z_f Z_p Z_c P(G_s) P(L_s = k)^2}. \quad (22)$$

Comparing (22) with the following Gibbs distribution

$$\begin{aligned} P(L_s = k|G_s, L_{s+r}, \forall r \in N^p, \tilde{L}_s) &= \frac{e^{-E(L_s=k|G_s, L_{s+r}, \forall r \in N^p, \tilde{L}_s)}}{Z} \end{aligned} \quad (23)$$

we get the energy function as

$$\begin{aligned} E(L_s = k|G_s = g_s, L_{s+r}, \forall r \in N^p, \tilde{L}_s) &= d(g_s, \mu_k) - \sum_{\forall r \in N^p} \beta \delta(k - L_{s+r}) \\ &\quad + \sum_{l \in \tilde{L}_s} \alpha \bar{\delta}(k-l) \end{aligned} \quad (24)$$

and normalization constant Z as $Z_p Z_c P(G_s = g_s) P(L_s = k)^2$. For any pixel s with a feature vector g_s , the *a priori* probability $P(G_s = g_s)$ can be considered as a constant. Also, any pixel s may be considered to have an equal *a priori*

probability $P(L_s = k)$ of having any label k . This makes Z in (23) a constant value independent of s and k .

Assuming that the labeling of the pixels subjected to the constraints of feature quantization process, partition process, and competition process is independent, the probability of the state of VQ is the product of individual conditional probabilities. Hence, the classification model is given by

$$P(S) = \prod_{s,k} P(L_s = k | G_s, L_{s+r}, \forall r \in N^p, \tilde{L}_s) \quad (25)$$

The energy function for $P(S)$ is

$$E^{\text{total}} = \sum_{s,k} \left[d(g_s, \mu_k) - \sum_{\forall r \in N^p} \beta \delta(k - L_{s+r}) + \sum_{l \in \tilde{L}_s} \alpha \bar{\delta}(k - l) \right]. \quad (26)$$

This energy captures the constraints acting on each pixel and on the possible labels for the pixel. Estimation of a state configuration S , which minimizes the energy in (26), will yield a classification result for the textured image. A neural network model, which is a modification of Hopfield network, together with a deterministic relaxation procedure is proposed to determine the state with minimum energy.

Thus, the overall texture classification scheme is now transformed into designing a vector quantizer that has a distance function E^{total} defined by (26). In this approach, the quantization and the determination of the codevectors are both dependent not only on the feature vector of the pixel of interest but also on the labels of the neighboring pixels and possible labels for the pixel. This is in contrast with the conventional VQ model given in (7), which encodes the features vectors independent of image-specific constraints.

IV. NEURAL NETWORK REPRESENTATION

The energy function in (26) can be represented on a constraint satisfaction neural network of size $I \times J \times K$ for an image T_Ω of size $I \times J$ with K number of maximum possible texture classes. Consider a neural network consisting of a three-dimensional (3-D) lattice of nodes. Each node in the network is denoted by (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. Any node (i, j, k) in the network has a bias $B_{i,j,k}$. Also, each node (i, j, k) is connected to any other node (i_1, j_1, k_1) in the lattice with a connection weight $W_{i,j,k;i_1,j_1,k_1}$, which is assumed to be symmetric, i.e., $W_{i,j,k;i_1,j_1,k_1} = W_{i_1,j_1,k_1;i,j,k}$.

We use $\Lambda_{i,j,k} \in \{0, 1\}$ for denoting the output of the node (i, j, k) . The set $\{\Lambda_{i,j,k}, \forall i, j, k\}$ is called state of the network. This has similar meaning as the state of VQ described in feature quantization process. $\Lambda_{i,j,k} = 1$ at any instant indicates that the pixel (i, j) has taken a label k at that instant. $\Lambda_{i,j,k}(n)$ denotes the output of node (i, j, k) at n th iteration of the relaxation algorithm. Hopfield defined the following energy

function for such a network [24]:

$$E^{\text{Hopfield}} = -\frac{1}{2} \sum_{i,j,k} \sum_{i_1,j_1,k_1} W_{i,j,k;i_1,j_1,k_1} \Lambda_{i,j,k} \Lambda_{i_1,j_1,k_1} - \sum_{i,j,k} B_{i,j,k} \Lambda_{i,j,k}. \quad (27)$$

We describe how to determine the bias and weights of this neural network in order to represent the energy function in (26) of the classification model. The first term in the energy function, which is the contribution of the feature quantization process, is active only if $L_s = k$, that is, if $\Lambda_{i,j,k} = 1$ where $s = (i, j)$. The instantiation k for the label random variable L_s of the pixel $s = (i, j)$ denotes the truth value $\Lambda_{i,j,k} = 1$ of the hypothesis $h(i, j, k)$. Similarly, the instantiation $L_{s+r} = k, s+r = (i_1, j_1)$ indicates that $\Lambda_{i_1,j_1,k} = 1$ for the hypothesis $h(i_1, j_1, k)$. So the term $\delta(L_s - L_{s+r})$ in (26) is equivalent to the product $\Lambda_{i,j,k} \Lambda_{i_1,j_1,k}$, and is active only if (i_1, j_1) is in the p th order neighborhood of (i, j) . The term $\bar{\delta}(L_s - l)$ is one only if $L_s \neq l$. If L_s has an instantiation k , this term is equal to

$$\bar{\delta}(k - l) = \begin{cases} \Lambda_{i,j,k} \Lambda_{i,j,l} & \text{if } k \neq l \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

So the energy function in (26) is rewritten as

$$E^{\text{total}} = \sum_{(i,j),k} \left[d(g_{(i,j)}, \mu_k) \Lambda_{i,j,k} - \sum_{\forall (i-i_1, j-j_1) \in N^p} \beta \Lambda_{i,j,k} \Lambda_{i_1,j_1,k} + \sum_{\forall l \neq k} \alpha \Lambda_{i,j,k} \Lambda_{i,j,l} \right]. \quad (29)$$

Comparing equations (27) and (29), the bias $B_{i,j,k}$ and the weight $W_{i,j,k;i_1,j_1,k_1}$ can be written as

$$B(i, j, k) = -d(g_{(i,j)}, \mu_k) \quad (30)$$

and

$$W_{i,j,k;i_1,j_1,k_1} = \begin{cases} 2\beta, & \text{if } (i - i_1, j - j_1) \in N^p \text{ and } k = k_1 \\ -2\alpha, & \text{if } (i_1, j_1) = (i, j) \text{ and } k \neq k_1 \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

Initially, each component $\sigma_s, s \in \Omega$ in the state S of VQ is randomized. Correspondingly, the state of network is initialized as

$$\Lambda_{i,j,k}(0) = \begin{cases} 1, & \text{if } \sigma_{(i,j)} = k \\ 0, & \text{otherwise.} \end{cases} \quad (32)$$

Also, the initial VQ state is used to compute the codevectors as the centroids of initial clusters.

Note that for a fixed α , the term β controls the activities of feature quantization process and partition process on labeling the pixels. For small β , the quantization process will be prominent, reducing the effect of partition process. The reverse

will result for high β . With a random initial state, a high β value forces the network to settle in a random state itself because of the uncorrelated behavior of pixel labels at the beginning. We avoid this by starting the relaxation with a very small value of β and increasing its value at each iteration. An exponential rise with a comparatively high time constant is used for this purpose. However, the shape of this curve and the value of time constant are not critical in the relaxation process.

The network state corresponding to the maximum *a posteriori* probability is arrived at by using a deterministic relaxation method in which the state of network is updated iteratively by changing the output of each node. The steps in the n th iteration of the relaxation algorithm are as follows.

- 1) Select a pixel (i, j) .
- 2) Compute the net input $U_{(i,j,k)}(n)$ of each node (i, j, k) , for all k as the weighted sum of inputs to that node. This is given by

$$U_{i,j,k}(n) = \sum_{i_1, j_1, k_1} W_{i,j,k;i_1,j_1,k_1} \Lambda_{i_1,j_1,k_1}(n) + B_{i,j,k} \quad (33)$$

- 3) Choose the label k^+ as the winner for the pixel (i, j) using the following conditions:

$$k^+ = \arg \max_k U_{i,j,k}(n) \quad (34)$$

and

$$U_{i,j,k^+}(n) > 0. \quad (35)$$

Second condition given in (35) is equivalent to assuming a threshold function for the output of each node. Correspondingly, there is a loser k^- which was the winner for the pixel (i, j) in the previous iteration. That is, $k^-(n) = k^+(n-1)$.

- 4) Update the output of node (i, j, k^+) as $\Lambda_{i,j,k^+}(n+1) = 1$. A penalty is given to the loser by setting $\Lambda_{i,j,k^-}(n+1) = 0$.
- 5) Detach the feature vector $g_{(i,j)}$ from the region R_{k^-} and assign it to the region R_{k^+} . Corresponding component in the state of VQ is updated as, $\sigma_{(i,j)} = k^+$. New centroid of the region R_{k^+} is calculated and assigned to the corresponding codevector as

$$\mu_{k^+} = \frac{1}{|R_{k^+}|} \sum_{g_s \in R_{k^+}} g_s. \quad (36)$$

In this expression, $|R_{k^+}|$ denotes the number of patterns belonging to the region R_{k^+} after the pattern $g_{(i,j)}$ is deleted from the region R_{k^-} and added to the region R_{k^+} . The change of codevector leads to a change in the bias of node (i, j, k^+) , which is computed using the expression (30). It should be noted that steps 4 and 5 are performed only if $k^+(n) \neq k^+(n-1)$.

- 6) Repeat steps 1 to 5 for all pixels selected sequentially from the image.
- 7) Update β according to the schedule

$$\beta(n) = \beta_\infty (1 - e^{-(n/\tau)}) \quad (37)$$

where β_∞ is the saturation value (at $n \rightarrow \infty$) and τ is the time constant.

The above model differs from the conventional Hopfield network in two aspects. In the first place, the bias and the connection weights in the network are modified continuously according to the expressions (36) and (37). Secondly, the outputs of two nodes—the winner and the loser—are updated simultaneously in the proposed network, whereas the Hopfield network updates only one node at a time. Energy in the Hopfield network is a function of state of the network, as the weights and bias are assumed constant. But in the present model, the energy is a function of weights and bias also. The energy analysis of such a network is much more complex and is not addressed in this paper. However, we show experimentally that the energy decreases at each iteration and gradually settles to a constant value after a sufficient number of iterations. Fig. 6 illustrates the behavior of the energy for the four examples discussed in Section VI. Similar behavior of energy curves can be obtained for different values of number of clusters.

Define $\Delta E = E(n) - E(n-1)$ as the energy change at the n th iteration, where $E(n)$ and $E(n-1)$ are the energies at the end of n th and $(n-1)$ th iterations, respectively. The criteria we have adopted for stopping the relaxation iteration is that $|\Delta E| \leq \epsilon$, where ϵ is an acceptably small positive quantity. The state of the network at this position gives the equilibrium state at which the *a posteriori* probability in (19) is maximum. At this point, the label of each pixel (i, j) is directly obtained from the corresponding component $\sigma_{(i,j)}$ of the state S , and this provides the classification of the given image. For each choice of number of classes, we obtain a classification result. The suitable number of classes is chosen based on a cluster validation index to be described in the next section.

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 quantization process. Thus, a rough estimate of these parameters can be obtained by comparing the average 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. It must be noted that the presented scheme does not describe an efficient method to determine the optimal value of p . However, this is an open problem in the Markov models [25] that lacks a satisfactory solution even in specific domains.

V. CLUSTER VALIDATION

Cluster validation is important in deciding the correct number of classes in the case of unsupervised classification of

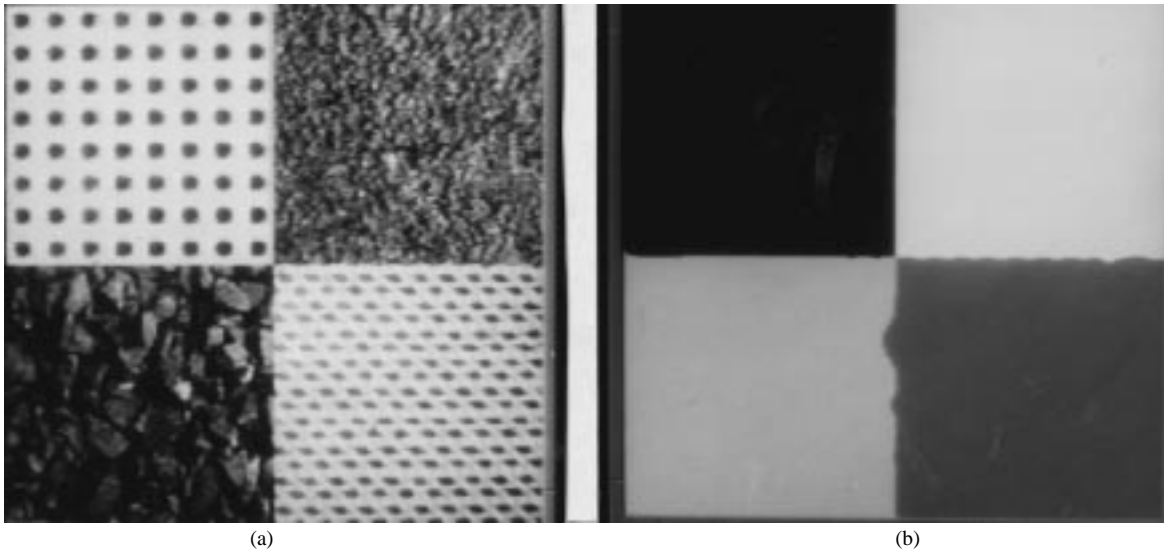


Fig. 1. Unsupervised classification of textures. (a) Original texture. (b) Classification result with four classes.

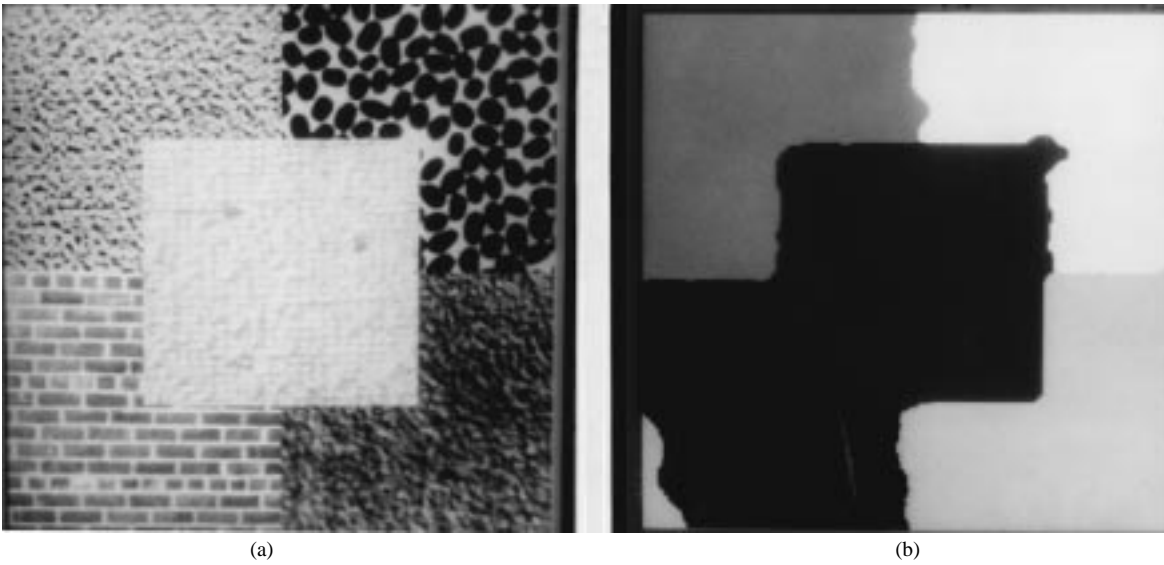


Fig. 2. Unsupervised classification of textures. (a) Original texture. (b) Classification result with five classes.

patterns. Several methods have been proposed for determining optimum number of clusters in a given data [26]. In our experiments, we have used the modified Hubert (MH) index proposed by Dubes [27].

For the purpose of using this index, we will redefine a few symbols as follows: Let us assume that the pixels are denoted by a single index i such that $0 \leq i < C_\Omega$, where C_Ω is the cardinality of the set Ω of pixel positions. Then, g_i refers to the feature vector of i th pixel in the image. Let σ_i refer to the cluster to which the pixel i is assigned. That is, $\sigma_i = k$ where $k = 0, 1, \dots, K - 1$. Then μ_{σ_i} is the centroid of the cluster σ_i . In terms of the above notation, the MH index for K clusters is defined by [27]

$$\text{MH}(K) = \frac{r - \Phi_f \Phi_c}{\nu_f \nu_c} \quad (38)$$

where

$$r = \sum_{i=0}^{C_\Omega-2} \sum_{j=i+1}^{C_\Omega-1} d(g_i, g_j) d(\mu_{\sigma_i}, \mu_{\sigma_j}) \quad (39)$$

$$\Phi_f = A \sum_{i=0}^{C_\Omega-2} \sum_{j=i+1}^{C_\Omega-1} d(g_i, g_j) \quad (40)$$

$$\Phi_c = A \sum_{i=0}^{C_\Omega-2} \sum_{j=i+1}^{C_\Omega-1} d(\mu_{\sigma_i}, \mu_{\sigma_j}) \quad (41)$$

$$\nu_f^2 = A \left[\sum_{i=0}^{C_\Omega-2} \sum_{j=i+1}^{C_\Omega-1} d^2(g_i, g_j) - \Phi_f^2 \right] \quad \text{and} \quad (42)$$

$$\nu_c^2 = A \left[\sum_{i=0}^{C_\Omega-2} \sum_{j=i+1}^{C_\Omega-1} d^2(\mu_{\sigma_i}, \mu_{\sigma_j}) - \Phi_c^2 \right]. \quad (43)$$

Here, $A = 2/C_\Omega(C_\Omega - 1)$.

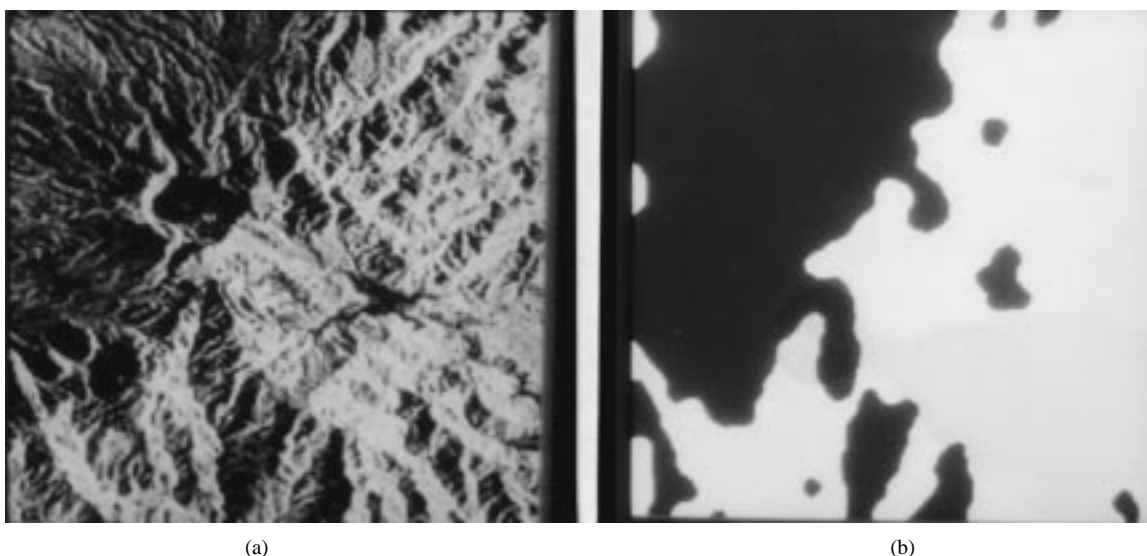


Fig. 3. Unsupervised classification of textures. (a) Original texture. (b) Classification result with two classes.

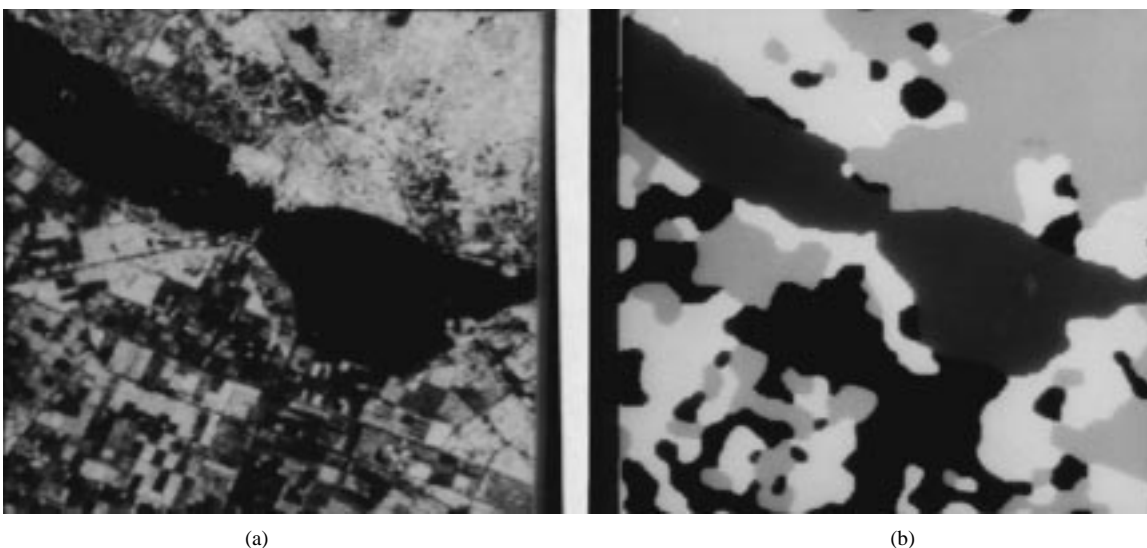


Fig. 4. Unsupervised classification of textures. (a) Original texture. (b) Classification result with four classes.

Thus, the MH index can be defined as a point serial correlation coefficient between the matrix of Euclidean distances of the patterns and a matrix containing the Euclidean distances between the cluster centers to which any two patterns belong. Assuming that the actual number of clusters is less than K_{\max} , $MH(k)$ for $k = 1 \dots K_{\max}$ is plotted against k . The MH index first increases with k and then becomes a constant. The number of clusters corresponding to the significant knee in the MH index denotes the actual number of clusters in the data. A detailed description and analysis of MH index can be found in [8] and [27].

VI. RESULTS AND DISCUSSION

We have used a number of textured images to study the performance of the proposed classification scheme. The images are categorized in increasing order of difficulty for classification. Images containing texture tiles comprise the first set of images [Figs. 1(a) and 2(a)]. In these images, the correct

number of texture classes and the spatial extent of each texture are visually interpretable. The second set [Figs. 3(a) and 4(a)] contains remote sensed images from the Spaceborne Imaging Radar-C and X-Synthetic Aperture Radar (SIR-C/X-SAR) of NASA/Jet Propulsion Laboratory [28]. Undefined texture boundaries and unknown texture models make the classification of these images difficult and, hence, a cluster validity measure is necessary to decide number of classes in the image. The results in each case show the original image and the classified image (with the optimal number of class determined by the MH index) using the proposed method. A graph showing the variation of MH index with the number of clusters is also provided for each image from which the correct number of texture classes for that image is selected. The graph showing the variation of network energy with respect to number of iterations is also given for each case.

Fig. 1(a) shows an image of size 256×256 pixels consisting of tiles of four different textures; two of them are nearly

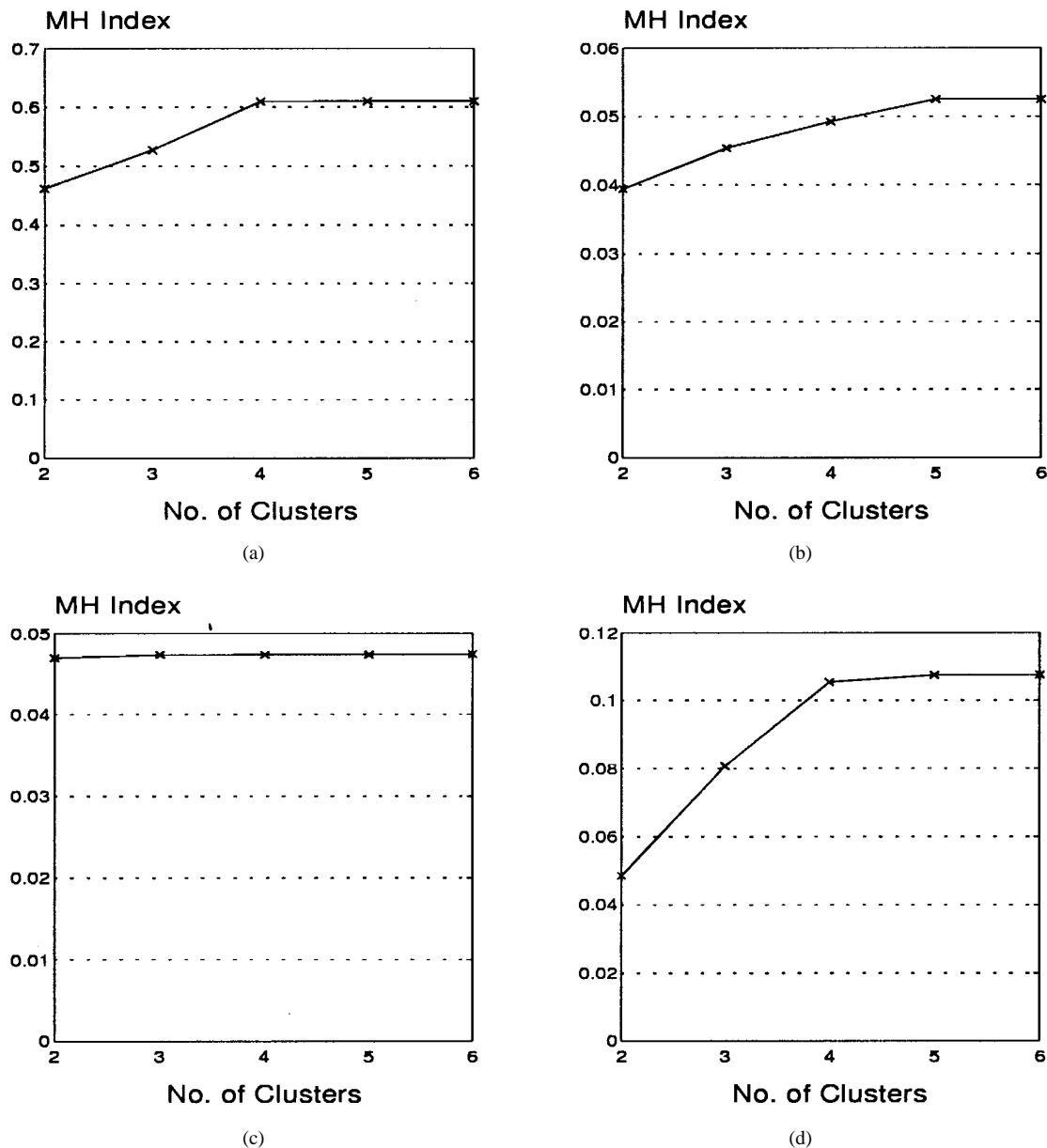


Fig. 5. Variation of MH index with number of clusters for the image in (a) Fig. 1; (b) Fig. 2; (c) Fig. 3; (d) Fig. 4.

deterministic (dots and diamonds) and the other two are stochastic in nature (sand and pebbles). The mother Gabor wavelet used to characterize this texture has parameters $\sigma = 25$ and $k = 0.1$. We have used two scales ($\gamma = 4$ and 5) and four orientations ($\theta = 0^\circ, 45^\circ, 90^\circ$, and 135°) for this mother wavelet, constituting an eight-dimensional feature vector for each pixel in the image. Fig. 1(b) shows the classification result using the proposed scheme for an optimal number (4) of classes. This is determined from the graph of MH index versus number of clusters given in Fig. 5(a). The result in Fig. 1(b), when compared with the original image, clearly shows the performance of the proposed scheme in correctly classifying the texture regions and in accurately locating the texture boundaries. Here, we have used a ninth-order neighborhood system for the partition process with a β_∞ value of 0.8 and a τ of 100.

An image with size 256×256 pixels, consisting of five textures from Brodatz album, is shown in Fig. 2(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 two different Gabor wavelet sets, one with $\sigma = 50, k = 0.1$ and another with $\sigma = 100, k = 0.1$, respectively. Each of these mother wavelets are scaled with $\gamma = 4, 5$ and rotated with $\theta = 0^\circ, 45^\circ, 90^\circ$, and 135° . This yields a total of 16 Gabor filters and leads to a 16-dimensional (16-D) feature vector for each pixel. The final classification obtained using the proposed scheme is shown in Fig. 2(c) for five classes, which is justified using the validity graph shown in Fig. 5(b) with a significant knee in the graph for $x = 5$. A comparison of the original image and the classification result shows that the proposed method has classified the given image well,

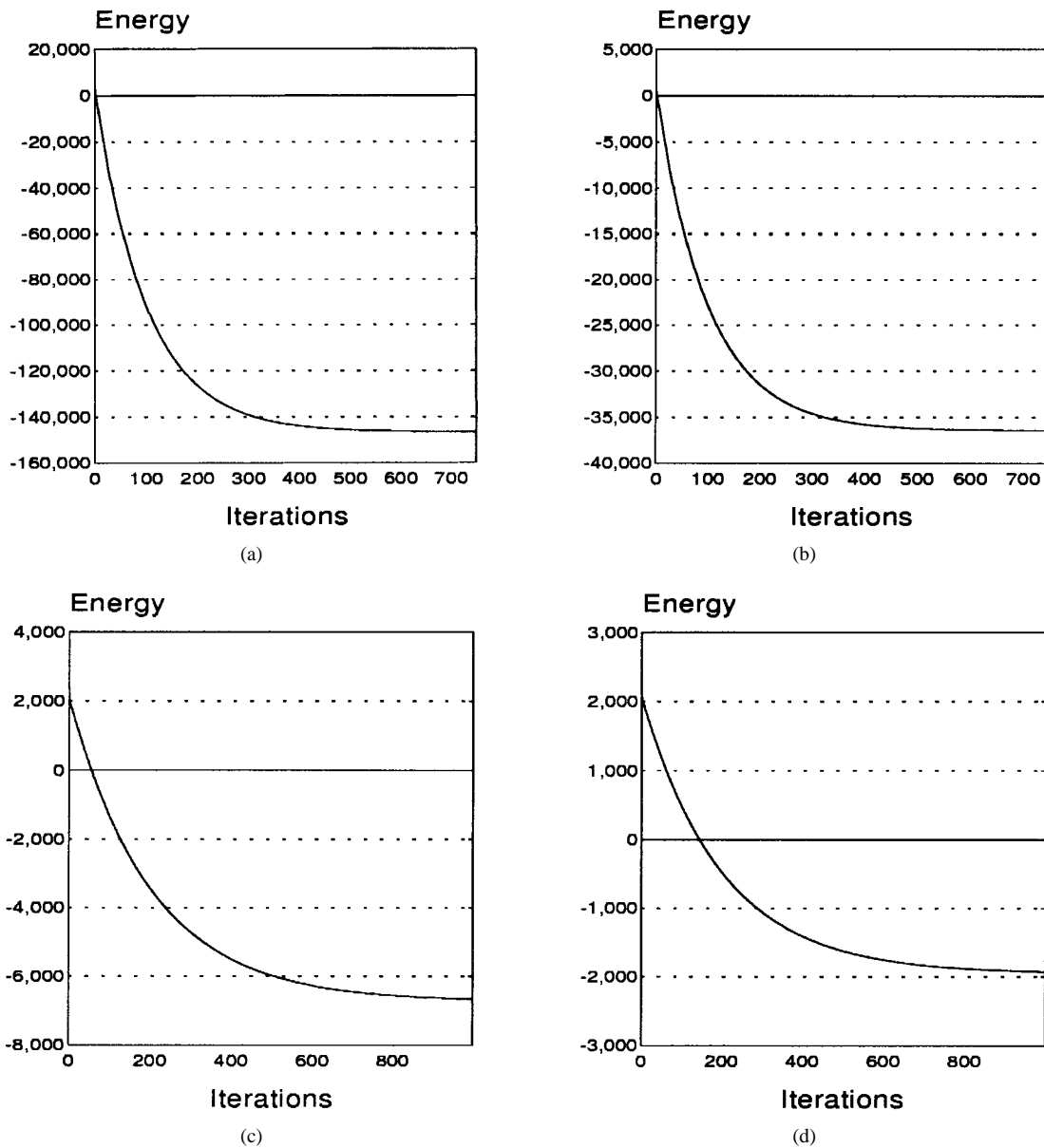


Fig. 6. Variation of energy with number of iterations in classifying the image in (a) Fig. 1; (b) Fig. 2; (c) Fig. 3; (d) Fig. 4.

especially in the interior portions of each texture tile. However, there are certain misclassified regions at the boundaries of textures. This is mainly due to the fact that the nearby texture classes affect the feature vectors of boundary pixels in each texture when convolved with the Gabor filters having comparatively high bandwidth. The parameters of the partition process used for this image are same as those in the previous experiment.

The next two images, which are from SIR-C/X-SAR, are characterized with unknown texture boundaries. So, in order to take into account the lack of *a priori* knowledge regarding the spatial extent of each texture present in the image, we have used a comparatively small order of neighborhood for partition process. Also, a higher value for τ is used to make the effect of feature quantization process much more prominent compared to the effect of the neighborhood pixel labels for a considerably large number of iterations.

The next data used for experiments consists of SIR-C/X-SAR image of the area around Mount Pinatubo in Philippines, and is given in Fig. 3(a). This image shows the regions affected with rough ash deposits during volcano eruptions. The image is of size 256×256 . For this image, the features are extracted by using 16 Gabor filters as the scaled ($\gamma = 2, 3, 4$ and 5) and rotated ($\theta = 0^\circ, 45^\circ, 90^\circ$, and 135°) versions of a mother wavelet with $\sigma = 50$ and $k = 0.1$, giving a 16-D vector for a pixel in the image. A sixth-order neighborhood was used in this experiment to characterize the partition process. Values for β_∞ and τ are 0.8 and 200. For an optimal number of classes of two, the final classification result is shown in Fig. 3(b). The MH index vs number of clusters graph is shown in Fig. 5(c) for this image. From this graph, it can be seen that the MH index has a constant value from the beginning of the graph, and this suggests a value two for the optimal number of classes. The classification result clearly brings out

the areas that are affected by volcano ash (the upper left part of the image) irrespective of certain artifacts present in the original image.

Finally, a 256×256 pixel SIR-C/X-SAR image of Flevoland, The Netherlands, shown in Fig. 4(a), is used as data for the classification experiment. For this image, we have used 16 Gabor filters with four scalings ($\gamma = 2, 3, 4$ and 5) and four rotations ($\theta = 0^\circ, 45^\circ, 90^\circ$, and 135° degrees) of a mother wavelet having parameters $k = 0.1$ and $\sigma = 25$. The neighborhood of order six was used for the partition process. For the experiment, we have used the values 0.8 and 200 for the parameters β_∞ and τ , respectively. Fig. 4(b) shows the final classification for four classes. A clear bend in the MH index graph in Fig. 5(d) shows that the optimal number of classes is four for this image. In fact, the description supplied along with this image provides the category of these classes. They are urban region (upper right part of the image), forest area (upper left part of the image), water body (middle region in the image), and bare soil in the agricultural fields (lower part of the image). Visual comparison of the original image and the classification result indicates that the result in Fig. 4(b) provides an acceptable classification of the given image, demonstrating the performance of the proposed scheme on such complicated natural images. However, the spurious spots in the classification result (especially, in the lower parts of the figure) are due to insignificant changes in the principal land type with similar properties as that of the forest and urban classes. To obtain a finer classification of these regions, one may have to incorporate multispectral features also along with the textural information.

In general, it is necessary to use many Gabor filters representing possible ranges of spatial extent, orientation, and frequency of texture elements. Thus, it is safe to use a wide range of values for σ , k and γ that determine the set of Gabor filters to be used to represent the features for unsupervised classification. We obtain similar results by using $\sigma = 25, 50, 100$, $k = 0.1$, and $\gamma = 2, 3, 4, 5$ for the Gabor filters in all the experiments reported in this paper. However, after looking at the results of the preliminary studies with large set of Gabor filters, we have selectively chosen a set of values for these parameters suitable for each experiment separately. This is just to reduce the overall computation. Thus, the specific choice of the parameters for each experiment does not limit the utility of the proposed method. It is always possible to obtain the same result using a much larger set of Gabor filters. For all experiments, we have used a constant value of 0.1 for the parameter α .

VII. CONCLUSION

In this paper, we have proposed an unsupervised texture classification scheme based on a neural network model working with a deterministic relaxation strategy. The textural features are extracted using a set of Gabor wavelets. A vector quantizer encodes these feature vectors into a set of codevectors. The VQ, expressed as Gibbs distribution, works as a feature quantization random process. A set of spatial as well as label constraints were provided by the

partition process and competition process. These different constraints were expressed as an *a posteriori* probability, and they were represented on a constraint satisfaction neural network. A deterministic relaxation algorithm was used to find the minimum energy, corresponding to maximum *a posteriori* probability, leading to the optimal classification of the textured image.

The influence of the labels of the neighboring pixels on the label of each pixel was gradually increased as iteration proceeds by adjusting the parameters of the partition process. As iteration proceeds, each codevector learns the centroids of the corresponding class. Unlike the conventional VQ methods, which encode the feature vectors independent of other feature vectors, the proposed method computes the codevectors dependent on the feature vector of the pixel of interest as well as the labels of the neighboring pixels. The validity of the number of texture classes was examined by using a measure known as modified Hubert index.

REFERENCES

- [1] J. Sklansky, "Image segmentation and feature extraction," *IEEE Trans. Syst., Man, Cybern.*, vol. 8, pp. 237-247, 1978.
- [2] J. Y. Haiso and A. A. Sawchuk, "Unsupervised texture image segmentation using feature smoothing and probabilistic relaxation techniques," *Comput. Vis. Graph. Image Process.*, vol. 48, pp. 1-21, 1989.
- [3] R. M. Gray, "Vector quantization," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 1, pp. 4-29, Apr. 1984.
- [4] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55-73, Jan. 1990.
- [5] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray, "Using vector quantization for image processing," *Proc. IEEE*, vol. 81, pp. 1326-1341, Sept. 1993.
- [6] K. L. Oehler and R. M. Gray, "Combining image compression and classification using vector quantization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 461-473, May 1995.
- [7] P. P. Raghu and B. Yegnanarayana, "Segmentation of Gabor-filtered textures using deterministic relaxation," *IEEE Trans. Image Processing*, vol. 5, pp. 1625-1636, Dec. 1996.
- [8] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognit.*, vol. 24, pp. 1167-1186, 1991.
- [9] J. M. Coggins and A. K. Jain, "A spatial filtering approach to texture analysis," *Pattern Recognit. Lett.*, vol. 3, pp. 195-203, 1985.
- [10] R. Bajcsy, "Computer description of textured surfaces," in *Proc. Int. Joint Conf. on Artificial Intelligence*, Stanford, CA, Aug. 1973, pp. 20-23.
- [11] S. Marceljia, "Mathematical description of the responses of simple cortical cells," *J. Opt. Soc. Amer. A*, vol. 70, pp. 1297-1300, 1980.
- [12] F. W. Campbell and J. G. Robson, "Application of Fourier analysis of the visibility of gratings," *J. Physiol.*, London, U.K., vol. 197, pp. 551-556, 1968.
- [13] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vis. Res.*, vol. 20, pp. 847-856, 1980.
- [14] D. A. Pollen and S. F. Ronner, "Visual cortical neurons as localized spatial frequency filters," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, Sept. 1983.
- [15] M. R. Turner, "Texture discrimination by Gabor functions," *Biolog. Cybern.*, vol. 55, pp. 71-82, 1986.
- [16] B. S. Manjunath and R. Chellappa, "A unified approach to boundary perception: Edges, textures, and illusory contours," *IEEE Trans. Neural Networks*, vol. 4, pp. 96-108, Jan. 1993.
- [17] J. G. Daugman, "Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, pp. 1160-1169, July 1985.
- [18] 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, pp. 1362-1373, Aug. 1990.
- [19] J. G. Daugman, "Entropy reduction and decorrelation in visual coding by oriented neural receptive fields," *IEEE Trans. Biomed. Eng.*, vol. 36, pp. 107-114, Jan. 1989.

- [20] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. Signal Processing*, vol. 39, pp. 2025–2043, Sept. 1991.
- [21] E. Yair, K. Zeger, and A. Gersho, "Competitive learning and soft competition for vector quantizer design," *IEEE Trans. Signal Processing*, vol. 40, pp. 294–309, Feb. 1992.
- [22] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COMM-28, pp. 84–95, Jan. 1980.
- [23] K. Rose, E. Guewitz, and G. C. Fox, "Vector quantization by deterministic annealing," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1249–1257, July 1992.
- [24] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, pp. 2554–2558, Apr. 1982.
- [25] R. L. Kashuap 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.
- [26] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [27] R. C. Dubes, "How many clusters are best? An experiment," *Pattern Recognit.*, vol. 20, pp. 645–663, 1987.
- [28] R. L. Jordan, B. L. Huneycutt, and M. Werner, "The SIR-C/X-SAR synthetic aperture radar system," *Proc. IEEE*, vol. 79, pp. 827–838, June 1991.



P. P. Raghu was born in Kerala, India in 1967. He received the B.Tech. degree in electrical engineering from Calicut University, Kerala, and the M. Tech. degree in computer and information sciences from Cochin University of Science and Technology, Kerala, in 1989 and 1991, respectively. He received the Ph.D. degree from the Indian Institute of Technology, Madras, India in 1996.

From 1991 to 1995, he was a Senior Project Officer in the Department of Computer Science and Engineering, Indian Institute of Technology.

In February 1996, he joined Tata Unisys Limited, Bangalore, India, as Senior Systems Analyst. His research interests include image processing, artificial neural networks, pattern recognition, and object-oriented software engineering.



tems.

R. Poongodi was born in Tamilnadu, India, in 1971. She received the B.E. degree in computer science in 1992 from Government College of Technology, Coimbatore, India.

She was a Project Associate in the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, during which the main focus of her work was on the application of neural networks for remote sensing and image processing. Her interests include image processing, neural networks, and data base management sys-



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, India. He was a visiting Professor at the Institute for Perception Research, Eindhoven Technical University, Eindhoven, The Netherlands, from July 1994 to January 1995. During the period of 1966 to 1971, he was engaged in the development of environmental test facilities for the Acoustics Laboratory, 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.