

Design of Vector Quantizer for Image Compression Using Self-Organizing Feature Map and Surface Fitting

Arijit Laha, Nikhil R. Pal, and Bhabatosh Chanda

Abstract—We propose a new scheme of designing a vector quantizer for image compression. First, a set of codevectors is generated using the self-organizing feature map algorithm. Then, the set of blocks associated with each code vector is modeled by a cubic surface for better perceptual fidelity of the reconstructed images. Mean-removed vectors from a set of training images is used for the construction of a generic codebook. Further, Huffman coding of the indices generated by the encoder and the difference-coded mean values of the blocks are used to achieve better compression ratio. We proposed two indices for quantitative assessment of the psychovisual quality (blocking effect) of the reconstructed image. Our experiments on several training and test images demonstrate that the proposed scheme can produce reconstructed images of good quality while achieving compression at low bit rates.

Index Terms—Cubic surface fitting, generic codebook, image compression, self-organizing feature map, vector quantization.

I. INTRODUCTION

WITH THE advent of World Wide Web and proliferation of multimedia contents, data compression techniques have gained immense importance. Data compression has become an enabling technology for efficient storage and transmission of multimedia data. In this paper, we propose a method for image compression by vector quantization of the image [1] using the self-organizing feature map [2] algorithm. We also propose refinement of the codebook using a method of cubic surface fitting for reduction of psychovisually annoying blocking effect.

A vector quantizer (VQ) [1] \mathcal{Q} of dimension k and size S can be defined as a mapping from data vectors (or “points”) in k -dimensional Euclidean space, \mathcal{R}^k into a finite subset \mathcal{C} of \mathcal{R}^k . Thus

$$\mathcal{Q}: \mathcal{R}^k \rightarrow \mathcal{C} \quad (1)$$

where $\mathcal{C} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_S\}$ is the set of S reconstruction vectors, called a *codebook* of size S , and each $\mathbf{y}_i \in \mathcal{C}$ is called a *code vector* or *codeword*. For each \mathbf{y}_i , $i \in \mathcal{I} = \{1, 2, \dots, S\}$ is called the index of the code vector and \mathcal{I} is the index set. Encoding a data vector $\mathbf{x} \in \mathcal{R}^k$ involves finding the index j of the

code vector $\mathbf{y}_j \in \mathcal{C}$ such that $\|\mathbf{x} - \mathbf{y}_j\| \leq \|\mathbf{x} - \mathbf{y}_i\|, \forall i / j$ and $i, j \in \mathcal{I}$. The decoder uses the index j to look up the codebook and generates the reconstruction vector \mathbf{y}_j corresponding to \mathbf{x} . The distortion measure $d(\mathbf{x}, \mathbf{y}_j) = \|\mathbf{x} - \mathbf{y}_j\|$ represents the penalty of reproducing \mathbf{x} with \mathbf{y}_j . If a VQ minimizes the average distortion, it is called the optimal VQ of size S .

Vector quantization has been used for image compression successfully by many researchers [6], [8], [10], [12]–[16]. The oldest, as well as most commonly used, method is the generalized Lloyd algorithm (GLA) [3], also known as the k -means algorithm. GLA is an iterative gradient descent algorithm that tries to minimize an average squared error distortion measure. Design of optimal VQ using GLA has been proposed and studied in [4]. However, GLA being a greedy algorithm, its performance is sensitive to initialization and converges to the local minima closest to the initial point. Fuzzy k -means algorithms [5] and several other fuzzy vector quantization techniques have been studied and used for image compression in [6]. Zeger *et al.* [7] proposed methods for designing a globally optimal vector quantizer using stochastic relaxation techniques and simulated annealing. Though these techniques can produce nearly optimal codebook, they are, in general, computationally intensive and slow to converge.

A self-organizing feature map [2] is a neural network clustering technique having several desirable features, and, consequently, it has attracted the attention of the researchers in the field of vector quantization [8]–[10]. The learning scheme of the self-organizing feature map (SOFM) is an application of the least mean square (LMS) algorithm where the weight of the neurons are modified “on the fly,” for each input vector, as opposed to the usual batch update scheme of GLA. Thus, the codebook is updated using an instantaneous estimate of the gradient, known as stochastic gradient, which does not ensure monotonic decrease of the average distortion. Consequently, the algorithm has a better chance of not getting stuck at a local minima. GLA can also incorporate incremental update through purely competitive learning. However, due to incorporation of neighborhood update (opposed to the “winner only” update in pure competitive learning) in the training stage, SOFM networks exhibit the interesting properties of topology preservation and density matching [2]. The former means that the vectors nearby in input space are mapped to the same node or nodes nearby in the output space (lattice plane of the SOFM nodes). The density matching property refers to the fact that after training the distribution of the weight vectors of the nodes reflects the distribution of the training vectors in the input space. Thus, more code vectors are

Manuscript received April 29, 2003; revised October 14, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lina J. Karam.

A. Laha is with National Institute of Management, Calcutta 700 027, India (e-mail: arijitl@yahoo.com).

N. R. Pal and B. Chanda are with the Electronics and Communication Science Unit, Indian Statistical Institute, Calcutta 700 108, India (e-mail: nikhil@isical.ac.in; chanda@isical.ac.in).

Digital Object Identifier 10.1109/TIP.2004.833107

placed in the regions with high density of training vectors. In other clustering algorithms, a dense and well-separated cluster is usually represented by a single cluster center. Though it is very good if the clusters are subsequently used for pattern classification task, in case of vector quantization, where the aim is to reduce the reconstruction error, this may not be that good.

The total reconstruction error of a VQ is the sum of *granular error* and *overload error* [1]. The granular error is the component of the quantization error due to the granular nature of the quantizer for an input that lies within a bounded cell. Due to the density-matching property, the SOFM places several prototypes in a densely populated region and, thus, makes the quantization cells small in such areas. This leads to the reduction in the granular error component resulting in preservation of finer details. This is a highly desirable property in a VQ. The overload error component of quantization error arises when the input lies in any unbounded cell representing data at the boundary of training sample distribution. Since the distribution of the code-words replicate the distribution of the training data, the overload error is also low when the distribution of the test data is well represented by that of the training data.

In [11], Chang and Gray introduced an online technique for VQ design using the stochastic gradient algorithm, which can be considered a special case of the SOFM algorithm, and it is shown to perform slightly better than GLA. Nasrabadi and Feng [8] also used SOFM for VQ design and demonstrated performance better than or similar to GLA. Yair *et al.* [9] used a combination of SOFM and stochastic relaxation and obtained consistently better performance than GLA. Amerijckx *et al.* [10] used Kohonen's SOFM algorithm to design a VQ for the coefficients of discrete cosine transform of the image blocks. The output of VQ encoder is further compressed using entropy coding. They reported performance equivalent to or better than standard JPEG algorithm.

In this paper, we propose a scheme for designing a spatial vector quantizer (SVQ) for image compression using Kohonen's SOFM[2] algorithm and surface fitting. We use a set of training images to design a generic codebook that is used for encoding the training as well as other images. The codebook is designed for *mean-removed* (also known as *residual*) [1], [15], [16] vectors. The mean is added to the reproduced vectors by the decoder. An initial codebook is generated by training an SOFM with the training vectors. Then, to achieve better psychovisual fidelity, each codevector is replaced with the best-fit cubic surface generated by the training vectors mapped to the respective codevector. The set of code indices produced by the encoder is further compressed using Huffman coding. A scheme of difference coding for the average values of the blocks is used. The difference coding enables us to utilize Huffman coding for the averages, which also leads to more compression.

In Section II, we describe the VQ design scheme in detail and report the experimental results in Section III. We use *peak signal-to-noise ratio* (PSNR) as one of the performance measures of the VQ. The PSNR (in decibels) for a 256-level image of size $h \times w$ is defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{hw} \sum_{i=1}^h \sum_{j=1}^w (x_{ij} - \hat{x}_{ij})^2} \quad (2)$$

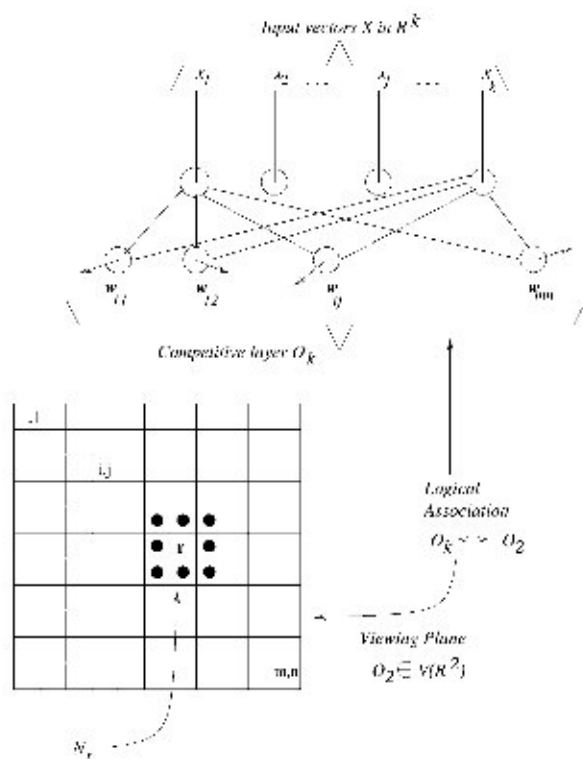


Fig. 1. SOFM architecture.

where x_{ij} is the value of the ij th pixel in the original image and \hat{x}_{ij} is that of the reconstructed image. In addition, we also propose two indices for quantitative assessment of psychovisual quality in the context of blocking effect. We use them to demonstrate the improvement of reconstruction quality by the surface-fitting codebooks over initial SOFM generated codebooks.

II. VECTOR QUANTIZER DESIGN

A. SOFM Architecture and Algorithm

The self-organizing feature map is denoted here by $A_{\text{SOFM}}^D: \mathcal{R}^k \rightarrow V(\mathcal{R}^l)$. This is often advocated for visualization of metric-topological relationships and matching density property of feature vectors (signals) $X = \{x_1, \dots, x_N\}$ in \mathcal{R}^k . Usually, X is transformed into a display lattice of $l \leq 3$ dimensions. In this article, we concentrate on $(m \times n)$ displays in \mathcal{R}^2 .

SOFM is realized by a two-layer network, as shown in Fig. 1. The first layer is the input layer or fan-out layer with k neurons and the second layer is the output or competitive layer. The two layers are completely connected. There are lateral inhibitory connections and self-excitatory connections between the neurons in layer two, which enable the neurons to compete among themselves to find the winner node given an input signal. An input vector $x \in \mathcal{R}^k$, when applied to the input layer, is distributed to each of the $(m \times n)$ output nodes in the competitive layer. Each node in this layer is connected to all nodes in the input layer; hence, it has a weight vector prototype w_{ij} attached to it.

SOFM begins with a (usually) random initialization of the weight vectors w_{ij} . For notational clarity, we suppress the

double subscripts. Now let $\mathbf{x} \in \mathcal{R}^k$ enter the network and let t denote the current iteration number. The neurons in layer two now compete among themselves to find the neuron whose weight vector matches best with the input \mathbf{x} . In other words, it finds $\mathbf{w}_{i,t-1}$ that best matches \mathbf{x} in the sense of minimum Euclidean distance in \mathcal{R}^k . Then $\mathbf{w}_{i,t-1}$ and the other weights in its spatial neighborhood are updated using the rule

$$\mathbf{w}_{i,t} = \mathbf{w}_{i,t-1} + \alpha_t g_t (\mathbf{x} - \mathbf{w}_{i,t-1}) \quad (3)$$

where α_t is the learning parameter and $g_t = \exp(-\|\mathbf{x} - \mathbf{w}_{i,t-1}\|^2 / \sigma_t^2)$. α_t and σ_t both decrease with time t . The topological neighborhood also decreases with time. This scheme when repeated long enough, usually preserves spatial order in the sense that weight vectors which are metrically close in \mathcal{R}^k generally have, at termination of the learning procedure, visually close images in the viewing plane. Also, the distribution of the weight vectors in \mathcal{R}^k resembles closely the distribution of the training vectors X . So, the weight vectors approximate the distribution of the training data as well as preserve topology of input data on the viewing plane. These features make this algorithm attractive for VQ design because if there are many similar vectors, *unlike a clustering algorithm*, which will place only one prototype, SOFM will generate more code vectors for the high density region. Consequently, finer details will be better preserved.

B. Surface Fitting

Once the SOFM is trained, the codebook can readily be designed using the weight vectors as the reconstruction vectors. Images can be encoded by finding out, for each image vector, the code vector with the least Euclidean distance. However, all spatial vector quantizers produce some blockiness in the reconstructed image [14], [17], [18], i.e., in the reconstructed image, the boundaries of the blocks become visible. Even though the reconstructed image shows quite good PSNR, this effect often has some adverse psychovisual impact. Often, transform and/or subband coding is used to overcome this effect. However, this adds substantial computational overhead, since the raw image has to be transformed into frequency domain before encoding using the quantizer, and, also in decoder, the image has to be converted back into the spatial domain from the frequency domain.

In our method, we adopt a scheme of *polynomial surface fitting* for modifying the codevectors generated by SOFM algorithm that reduces the blockiness of the reconstructed image and improves its psychovisual quality. In this case, the computational overhead occurs only at the codebook design stage, not during encoding or decoding of each image. Although, in computer graphics, polynomial surfaces serve as standard tools for modeling the surface of graphical objects [19], in image coding, their application is mostly restricted to image segmentation and representation of segmented patches. In [20] and [21], low-degree polynomials are used for local approximation of segmented patches, while in [22], Bezier–Bernstein polynomials are used for image compression by globally approximating many segmented patches by a single polynomial with local corrections. To the best of our knowledge, there is no reported work that uses polynomial surfaces in context of vector quantization.

For using polynomial surfaces, one has to decide on the degree of the surface to be used. Lower-degree polynomials are easier to compute, but they have less expressive power. On the other hand, higher-degree polynomials, though have more expressive power, they tend to replicate the training data exactly and lose the power of generalization, and, thus, unacceptable results may be produced when presented with an image not used in the training set. There is also the issue of solving for the coefficients of the polynomial. If a small block size is used, there may not be enough points in a block to find a solution of a high-degree polynomial.

In our work, we have experimented with block sizes 4×4 , 4×8 , and 8×8 . It is found that for 4×4 blocks, biquadratic surfaces give performances comparable to that of bicubic surfaces, while for larger block sizes, bicubic surfaces perform significantly better. Surfaces of degree 4 do not improve the performance significantly for test images. So, we use the bicubic surfaces throughout our work.

In this scheme, we try to find for each reconstruction block a cubic surface centered at the middle of the block so that the gray value x of a pixel at (α, β) (with respect to the origin set at the middle of the block) can be expressed by the *bicubic* equation

$$\begin{aligned} x(\alpha, \beta) = & a_0 + a_1\alpha^3 + a_2\beta^3 + a_3\alpha^2\beta + a_4\alpha\beta^2 \\ & + a_5\alpha^2\beta^2 + a_6\beta^2 - a_7\alpha\beta + a_8\alpha + a_9\beta \\ & \mathbf{p}^T \mathbf{a} \end{aligned} \quad (4)$$

where

$$\mathbf{p} = [1, \alpha^3, \beta^3, \alpha^2\beta, \alpha\beta^2, \alpha^2, \beta^2, \alpha\beta, \alpha, \beta]^T$$

and

$$\mathbf{a} = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9]^T.$$

Thus, for a whole block, we can write

$$P = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_k^T \end{bmatrix}_{k \times 10} \quad (5)$$

where \mathbf{p}_i is the \mathbf{p} vector for i th component of feature vector \mathbf{x} .

To find the coefficients of the polynomials corresponding to the codevectors generated by the SOFM, we divide the training vectors into groups such that a training vector belongs to i th group, if it is mapped to the i th codevector. Let $X = \{\mathbf{x} \in \mathcal{R}^k\}$ be the set of training vectors and $X^{(i)} = \{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \dots, \mathbf{x}_{n_i}^{(i)}\} \subset X$ where $n_i = |X^{(i)}|$ is the set of training vectors mapped to the i th code vector. We write for the set $X^{(i)}$

$$\mathbf{G}^{(i)} = \begin{bmatrix} \mathbf{x}_1^{(i)} \\ \mathbf{x}_2^{(i)} \\ \vdots \\ \mathbf{x}_{n_i}^{(i)} \end{bmatrix}_{n_i \times 10}$$

and

$$\mathbf{a}^{(i)} = [a_0^{(i)}, a_1^{(i)}, \dots, a_9^{(i)}]^T.$$

Then, (4) can be written in a matrix form for all the vectors mapped to the i th code vector as

$$\mathbf{G}^{(i)} = \mathbf{P}^{(i)} \mathbf{a}^{(i)} \quad (6)$$

where

$$\mathbf{P}^{(i)} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix}_{kn_i \times 10} \quad \text{and } P_i = P \quad \forall i = 1, 2, \dots, n_i.$$

Thus, the total squared error due to reconstruction of all blocks corresponding to vectors in $\mathcal{X}^{(i)}$ using the i th reconstruction block generated by cubic surface specified by the coefficient vector $\mathbf{a}^{(i)}$ can be expressed as

$$E = \|\mathbf{G}^{(i)} - \mathbf{P}^{(i)} \mathbf{a}^{(i)}\|^2. \quad (7)$$

Differentiating (7) with respect to $\mathbf{a}^{(i)}$, we obtain

$$\frac{\partial E}{\partial \mathbf{a}^{(i)}} = \mathbf{P}^{(i)T} (\mathbf{G}^{(i)} - \mathbf{P}^{(i)} \mathbf{a}^{(i)}). \quad (8)$$

Thus, the coefficient vector $\mathbf{a}^{(i)}$ corresponding to the minimum of E can be obtained by solving the equation

$$\mathbf{P}^{(i)T} (\mathbf{G}^{(i)} - \mathbf{P}^{(i)} \mathbf{a}^{(i)}) = 0 \quad (9)$$

for $\mathbf{a}^{(i)}$. From (9), we obtain

$$\mathbf{a}^{(i)} = (\mathbf{P}^{(i)T} \mathbf{P}^{(i)})^{-1} \mathbf{P}^{(i)T} \mathbf{G}^{(i)} = \mathbf{P}^{(i)+} \mathbf{G}^{(i)} \quad (10)$$

where $\mathbf{P}^{(i)+} = (\mathbf{P}^{(i)T} \mathbf{P}^{(i)})^{-1} \mathbf{P}^{(i)T}$ is the *pseudoinverse* of $\mathbf{P}^{(i)}$.

We compute the coefficient vectors $\mathbf{a}^{(i)}$ s for all codevectors obtained from SOFM. Once the $\mathbf{a}^{(i)}$ s are available, the codebook can be designed to contain the coefficients $\{\mathbf{a}^{(i)}\}$. If we do so, then for encoding, we have to do the following. For each polynomial $\{a^{(i)}\}$, obtain the surface by computing the pq real values corresponding to each pixel of a block. Round these real values, subject to the constraint that all values lie in $\{0, 1, \dots, L-1\}$. Notice, here, that given a coefficient vector $\mathbf{a}^{(i)}$, the surface generated $\hat{\mathbf{x}}_i$ remains the same irrespective of the spatial location of the associated block on the image. The reason is that for every block we use the central point of the block as the origin. So, we can avoid a lot of useless computation if, instead of storing $\mathbf{a}^{(i)}$ s, we store the $\hat{\mathbf{x}}_i$ s in the code book. Let these generated surfaces be $\{\hat{\mathbf{x}}_i, i = 1, 2, \dots, S\}$. Then, for every block of size pq of the image, we find the closest $\hat{\mathbf{x}}_i$ and use its index to encode. Then, while decoding, we reverse the procedure using $\hat{\mathbf{x}}_i$ s and the indices used to code the blocks. In other words, *we do not store the $\mathbf{a}^{(i)}$ s, but the code vectors reconstructed using $\mathbf{a}^{(i)}$ s*. We use the new codebook for encoding, as well as decoding, subsequently. We have tested the effectiveness of this method using several images. Although the quality improvement in terms of PSNR appears marginal, significant



Fig. 2. Original Lena image.

improvement in terms psychovisual quality is observed consistently. Fig. 2 shows the original Lena image. Fig. 3(a) shows the reconstructed Lena image using SOFM codebook (block size 8×8 and codebook size 256), while Fig. 3(b) depicts the same using surface-fitting codebook. Though the surface-fitting codebook marginally increases the PSNR (0.3 dB) for this image, reduction of blockiness and improvement of psychovisual quality are evident. To demonstrate the effect more clearly, we show in Fig. 4 enlarged portions (containing lips) of the images in Fig. 3(a) and (b).

C. Construction of Generic Codebook

In most of the experimental works on vector quantization, the codebooks are trained with the test image itself. However, this poses a problem in practical use of such algorithm for transmission/storage of the compressed image. While transmitting a compressed image, both the compressed image and the codebook must have to be transmitted. The overhead in transmitting the codebook diminishes the compression ratio largely. This problem can be avoided if a generic codebook is used by both the transmitter and the receiver [23]. Construction of such a generic codebook poses a formidable problem, since, in general, if an image is compressed using a codebook trained on a different image, the reconstruction error tends to be substantially high.

Next, we explore the possibility of construction of a generic codebook that can be used to encode any image (i.e., images other than those used to construct the codebook) with acceptable fidelity. Such a codebook can allow us one time construction of the encoder-decoder and making the codebook a permanent part of it. To achieve this, we select a set of images having widely varied natures in terms of details, contrasts, and textures. We use these images together for construction of the generic codebook. We prepare a 768×512 training image (Fig. 5) which is a composite of six smaller 256×256 images and train the SOFM using this training image and then construct the surface-fitting codebook for the VQ.



Fig. 3. (a) Reconstructed Lena image with VQ using SOFM weight vectors as reconstruction vectors. PNSR = 28.19 dB. (b) Reconstructed Lena image with VQ using surface fitting. PNSR = 28.49 dB. Here, the Lena image is used for training the VQ.

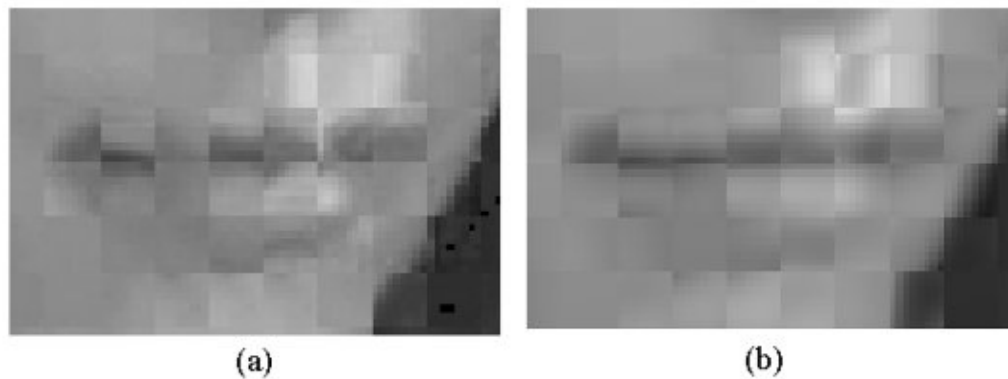


Fig. 4. (a) Enlarged portion of Lena image shown in Fig. 3(a). (b) Enlarged portion of Lena image shown in Fig. 3(b).

We emphasize that the generic codebook constructed by the above-mentioned method can neither be claimed as universal nor as the best possible generic codebook. Our aim is merely to demonstrate experimentally that a generic codebook constructed from some judiciously chosen images can be used for effective compression of other images having “similar characteristics.” If the images to be compressed are of a radically different nature, say containing texts or geometric line drawings, use of another generic codebook constructed with images of a similar type will be appropriate. Note that, when we say images of “similar” characteristics, we do not refer to images which are visually similar, but images with similar distribution of gray level over small blocks of size, say 8×8 .

III. EXPERIMENTAL RESULTS

We study three vector quantizers using block sizes 8×8 (VQ1), 4×8 (VQ2), and 4×4 (VQ3), respectively. Each of the VQs uses a codebook of size 256 and is trained with *mean-removed vectors*. Thus, to represent each block in the encoded image, 1 byte is required for the index and another byte is required for the block average. This makes the compression ratios 0.25 bpp (bits per pixel), 0.5 bpp, and 1 bpp for VQ1,

VQ2, and VQ3, respectively. To achieve more compression, lossless Huffman encoding is applied separately to the indices and the block averages. Codeword assignment for the indices is based on the frequency distribution of the codevectors in the encoded training image. Because of a strong correlation between neighboring blocks, the absolute differences between average values of neighboring blocks are found to have a monotonically decreasing distribution and codewords are assigned exploiting this.

Each VQ is trained with the image shown in Fig. 5. The training image is a composite of six 256×256 images. The individual images in the training set are **Blood cell**, **Peppers**, **Keyboard**, **Lena**, **Cameraman**, and **Chair**. We report the test results for six images, of which **Lena**, **Barbara**, and **Boat** are of size 512×512 and **Bird**, **House**, and **Mattface** are of size 256×256 . Please note that the Lena images used in training and test sets are different. The former is of size 256×256 while the later has the size 512×512 . The performances of the vector quantizers for the training images are summarized in Table I and those for the test images are summarized in Table II. In Table II, we also present the performances of standard JPEG algorithm. For every image, while using the JPEG algorithm, we tried to maintain the same compression rates as achieved by our scheme.



Fig. 5. Training image.

TABLE I
PERFORMANCE OF THE VECTOR QUANTIZERS ON TRAINING IMAGES

Image	Size	VQ1 (Blocksize 8×8)		VQ2 (Blocksize 4×8)		VQ3 (Blocksize 4×4)	
		PSNR (dB)	CR (bpp)	PSNR (dB)	CR (bpp)	PSNR (dB)	CR (bpp)
Blood Cell	256×256	28.32	0.21	30.75	0.44	33.50	0.80
Caucasian	256×256	23.58	0.16	24.65	0.34	26.51	0.65
Chair	256×256	31.25	0.14	32.39	0.28	34.09	0.54
Keyboard	256×256	30.28	0.17	31.01	0.36	33.92	0.67
Lena	256×256	26.42	0.21	27.82	0.44	30.49	0.82
Peppers	256×256	25.49	0.22	27.05	0.45	29.38	0.86

We used the routines available in Matlab 5 software to generate the JPEG images.

Our results for the test images show that the proposed algorithm consistently performs better than the JPEG in terms of PSNR for the lowest bit rate studied. In this case, the difference of PSNR value between the images produced by our scheme and JPEG varied between 0.23 dB for Lena image to 5.10 dB for the Bird image. For the higher bit rates, JPEG shows consistently higher PSNR values than the proposed algorithm at similar compression rates. To compare the psychovisual quali-

ties, we display in Fig. 6 the Lena images compressed using the proposed algorithm and JPEG algorithm. The images Fig. 6(a), (c), and (e) are compressed using the proposed algorithm with block sizes 8×8 , 4×8 , and 4×4 , respectively. The images Fig. 6(b), (d), and (f) are compressed using the JPEG algorithm with bit rates similar to the corresponding images in the left panels. It is evident from the images shown in Fig. 6 that, despite a small difference in PSNR values at the lowest bit rate (0.23 dB), the image compressed by the proposed algorithm is quite superior to the JPEG image in terms of psychovisual

TABLE II
PERFORMANCE OF THE VECTOR QUANTIZERS ON TEST IMAGES AND THEIR COMPARISON WITH BASELINE JPEG

Image	Compression Algorithm	VQ1 (Blocksize 8×8)		VQ2 (Blocksize 4×8)		VQ3 (Blocksize 4×4)	
		PSNR (dB)	CR (bpp)	PSNR (dB)	CR (bpp)	PSNR (dB)	CR (bpp)
Lena (512 \times 512)	Proposed	28.47	0.18	30.14	0.38	33.03	0.74
	JPEG	28.24	0.20	33.28	0.38	36.43	0.73
Barbara (512 \times 512)	Proposed	23.85	0.19	24.33	0.41	25.09	0.79
	JPEG	23.48	0.19	27.16	0.41	31.12	0.73
Boat (512 \times 512)	Proposed	26.44	0.17	25.11	0.35	31.11	0.69
	JPEG	24.10	0.17	30.35	0.36	34.14	0.68
Bird (256 \times 256)	Proposed	30.27	0.15	32.56	0.30	35.60	0.57
	JPEG	25.17	0.16	34.42	0.29	38.63	0.54
House (256 \times 256)	Proposed	25.39	0.16	26.39	0.33	29.51	0.67
	JPEG	24.20	0.18	30.15	0.33	34.36	0.64
Mandrill (256 \times 256)	Proposed	28.69	0.18	31.96	0.37	35.46	0.69
	JPEG	27.04	0.19	35.35	0.37	39.60	0.66

quality. For higher bit rates, though the JPEG images have significantly higher PSNR values, their psychovisual qualities are similar to the corresponding images compressed with the proposed algorithm. Similar results are obtained for other images studied in this paper. For all other test images, the original images and the images reconstructed using the proposed algorithm for three VQs are shown in Figs. 7–9. The PSNR and the compression rates for these images [panels (b), (c), and (d) of Figs. 7–9] are reported in Table II.

A plethora of studies on the Lena image are available in the literature. Many of them can be found in the excellent survey paper of Cossman *et al.* [25]. However, they have concentrated exclusively on vector quantization of image subbands and the results reported cannot be readily compared with those of the SVQ techniques. Instead, we compare our results with other works using SVQ techniques. Zeger *et al.* [7] developed a variant of simulated annealing for VQ design and compared it with GLA. They used Lena and Barbara images of size 512 \times 512 to test the algorithms. They used 4 \times 4 block size and a codebook size of 256. The reported PSNRs are 30.48 and 25.80 dB, respectively, for GLA and 30.59 and 25.87 dB, respectively, for simulated annealing. Thus, the results reported in this paper are comparable for Barbara image and significantly superior for the Lena image. Karayiannis and Pai [6] developed several variants of a fuzzy vector quantization algorithm. They also reported results for VQs using traditional k -means algorithm and fuzzy k -means algorithm. They used the 256 \times 256 Lena image for both training and testing. In their study, they used image blocks of size 4 \times 4 and considered codebooks of different sizes. For a codebook of size 256, they reported the PSNRs 27.06 dB using k -means algorithm, 29.91 dB using fuzzy k -means algorithm, and 29.60, 29.93, and 29.95 dB for three different variants of fuzzy vector quantization algorithms. These results are similar to the results reported in this paper for the Lena image used in training the set. In [10], Kohonen's SOFM algorithm is used for designing a VQ. Here, SOFM is trained with the low-order coefficients of discrete cosine transform (DCT) of 4 \times 4 image blocks. The output of the encoder is further compressed using a differential encoding scheme. The result reported for the Lena image shows a PSNR of 24.7 dB with a compression rate 25.22 (i.e., 0.32 bpp approximately). In a recent work, Schneider *et al.* [17] studied wavelet-based lattice vector quantization methods. For the same image, they

reported a PSNR 32.06 dB at a compression ratio 24:1 (i.e., 0.33 bpp approximately). All other images studied in this paper also show high PSNR with good compression rates.

A. Quantitative Assessment of Preservation of Psychovisual Quality in Terms of Blockiness

Our goal is to devise a simple scheme of designing VQ that can compress images with good perceptual fidelity. It is a well-established fact that the mean squared error (MSE) based distortion measures, such as PSNR, are not very good for measuring perceptual fidelity. However, there is no universally accepted quantitative measure for psychovisual quality. We presented a surface-fitting method for quantization that smoothes the reconstructed image resulting in a reduction of the blocking effect. Sometimes, it may introduce some blurring of sharp features. However, moderate blurring is not considered annoying by a human observer since it is a “natural” type of distortion [18]. The effectiveness of the proposed scheme for preserving psychovisual quality in the reconstructed images has been demonstrated visually in Figs. 6–9. The increased ability of reducing the blocky effect by surface-fitting scheme is also demonstrated visually in Figs. 4 and 5. Fig. 5 depicts enlarged views of some portion of the images shown in Fig. 4. The portion is selected in such a way that it contains fine detail as well as smooth non-linear intensity variation.

Now, we define two quantitative indices that can assess the preservation of psychovisual quality with respect to the blocking effect. The development is based on the following observations made by Ramstad *et al.* [18].

- 1) The blocking effect is a natural consequence of splitting the image into blocks and independent processing of each block. The quantization errors will lead to appearance of the blocks that the image is split into.
- 2) Blocking effects are visible in smooth image areas, whereas in complex areas, such as textures, any underlying blocking effect is effectively masked.

Thus, the degradation of psychovisual quality is contributed by 1) the reduction of smoothness, due to the difference between an image block and the quantized block that replaces it, and 2) the additional discontinuity imposed across the block boundary due to quantization. So, we develop a pair of quantitative indices. The first one measures the loss of smoothness per



Fig. 6. (a), (c), (e) Compressed Lena images using proposed algorithm. (b), (d), (f) compressed Lena images using JPEG.

pixel across the block boundary due to vector quantization. We call it the *boundary smoothness mismatch index* (BSMI). The second index deals with the difference of smoothness per pixel between the original image and the reconstructed image for the nonboundary pixels (i.e., all pixels in a block that are not on the boundary of the block). We call it the *inner smoothness dif-*

ference index (ISDI). Evidently, for both indices, lower values imply better preservation of psychovisual quality.

The development of these indices is based on the fact that the second derivative, i.e., *Laplacian* of a surface at a point, can be used as a measure of the lack of smoothness at that point. This fact is often used for detection of edges in an image [26], where



Fig. 7. Results on 512×512 barbara image. (a) Original image. (b) Reconstructed image for VQ with 8×8 blocks. (c) Reconstructed image for VQ with 4×8 blocks. (d) Reconstructed image for VQ with 4×4 blocks.

the pixels showing abrupt variation of intensity with respect to their neighbors, are detected. In our approach, we use the Laplacian to measure the lack of smoothness in intensity variation. The discrete realization of the operator in form of a convolution mask is shown in Fig. 10. Henceforth, we shall denote this mask as $\mathcal{L}(i, j)$, where (i, j) denotes the coordinate of the pixel on which the mask is applied.

Now, we present the formulae for computing the indices. Let P_B and P_I denote the set of pixels at the block boundaries and the set of nonboundary pixels in an image respectively. Then, the BSMI of an image is defined as

$$\text{BSMI} = \frac{\sum_{(i,j) \in P_B} [x_{ij} * \mathcal{L}(i, j)]^2}{\text{Number of pixels in } P_B}. \quad (11)$$

The computation of the ISDI is computed for a reconstructed image with respect to the original image and defined as

$$\text{ISDI} = \frac{\sum_{(i,j) \in P_I} [x_{ij} * \mathcal{L}(i, j) - \hat{x}_{ij} * \mathcal{L}(i, j)]^2}{\text{Number of pixels in } P_I} \quad (12)$$

where x_{ij} and \hat{x}_{ij} denote the intensities of the (i, j) th pixel in the original image and the reconstructed image, respectively. Note that, for both measures, the lower the value, the better the performance. We report the results of our study using the proposed indices in Table III.

As shown in Table III, for all 18 cases, the surface-fitting codebooks show better performances in terms of BSMI. This clearly indicates that for the surface-fitting codebooks the block boundaries maintain better continuity. Table III also reveals that for 13 (out of 18) cases, the ISDI values for the surface-fitting codebook are smaller than the corresponding ISDI values using the SOFM codebook. This means that, in these 13 cases, the similarity of the blocks reconstructed by the surface-fitting codebook is more like the original image than the similarity of the SOFM VQ reconstructed images with original ones. The remaining five cases involve three images with small block sizes (Barbara and House images with block sizes 4×8 and 4×4 and Boat image with block size 4×4). These results can be attributed to the fact that each of these images (original) contains substantial portions covered with complex texture-like areas, and, for smaller block sizes, the gain due to surface fitting over



Fig. 8. Results on 512×512 boat image. (a) Original image. (b) Reconstructed image for VQ with 8×8 blocks. (c) Reconstructed image for VQ with 4×8 blocks. (d) Reconstructed image for VQ with 4×4 blocks.

the SOFM is not reflected in ISDI values. Overall, Table III indicates that the replacement of the codevectors obtained directly from SOFM with the codevectors obtained by least square-error surface fitting results in VQs preserving the psychovisual fidelity to a better extent.

IV. CONCLUSION

We presented a comprehensive scheme for designing vector quantizers for image compression using generic codebooks that produce reconstructed images with good psychovisual quality. The scheme exploits the special features of SOFM for codebook generation and introduces a novel surface-fitting scheme for refinement of the codevectors generated by SOFM algorithm for reducing the blockiness in the reconstructed images. It also puts together some well-known concepts, such as mean-removed vectors and entropy coding of indices and difference-coded mean values. The proposed scheme, as a whole, achieves compression at low bit rates with good quality reconstructed images.

Due to the density matching and topology preservation properties of SOFM, it can be used to generate a good set of code vectors. Use of mean-removed vectors reduce the reconstruction error significantly, but they necessitated doubling of the amount of the data to be stored or transmitted. However, lower reconstruction error allows us to use larger image blocks with acceptable fidelity. The use of cubic surface fitting for refinement of the codevectors enables us to decrease the unpleasant blocking effect that appears in spatial VQs at low bit rates. The improvement due to surface fitting is demonstrated visually, as well as quantitatively, using two indices proposed in this paper. The computational overload due to surface fitting as proposed here is restricted to the codebook generation stage only, unlike the transform or subband-coding techniques, where every image has to be transformed into the frequency domain at the encoder side and inverse transformed into the spatial domain at the decoder side.

The use of generic codebook not only enables us to construct the codebook only once, but the knowledge of distribution of indices for the training images can also be exploited as the *a priori* knowledge of distribution of indices for the test images. This

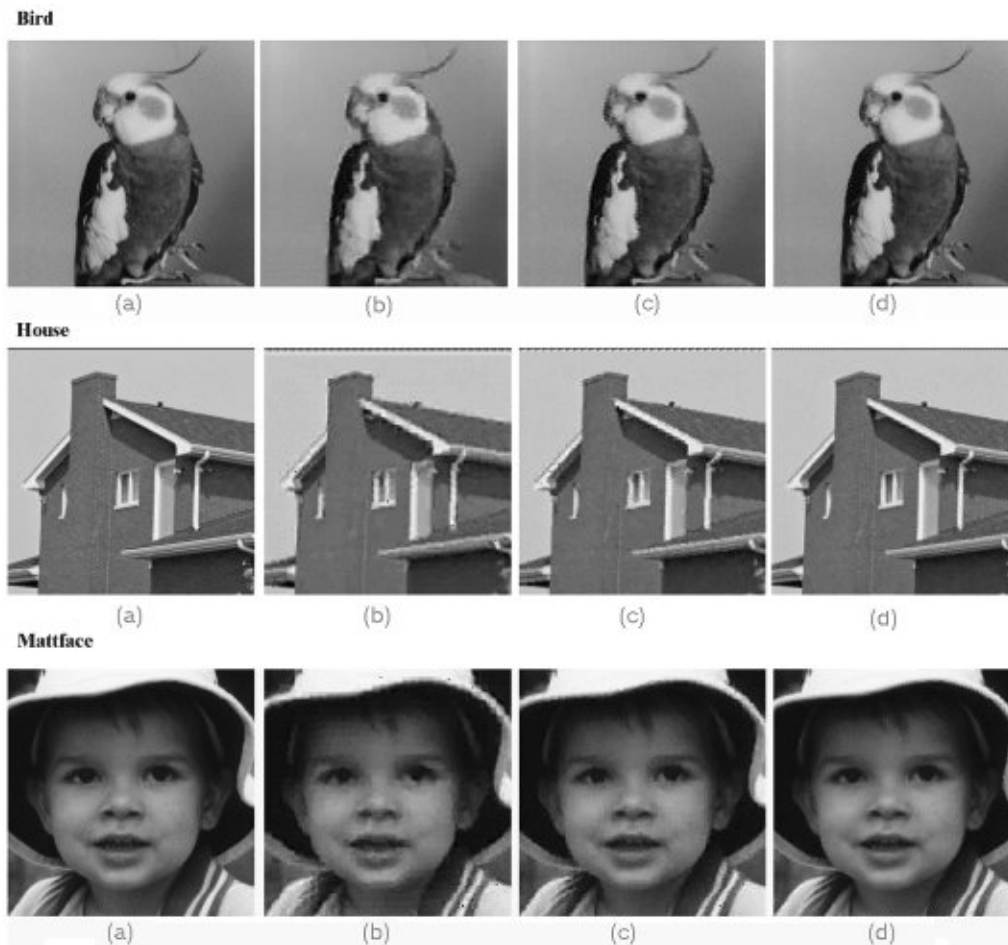


Fig. 9. Results on 256×256 images. (a) Original images. (b) Reconstructed images for VQ with 8×8 blocks. (c) Reconstructed images for VQ with 4×8 blocks. (d) Reconstructed images for VQ with 4×4 blocks.

TABLE III
COMPARISON OF PERFORMANCES REGARDING PRESERVATION OF PSYCHOVISUAL FIDELITY BETWEEN THE VECTOR QUANTIZERS USING SOFM CODE BOOKS AND SURFACE-FITTING CODE BOOKS

Image	Code book used	Blocksize 8×8		Blocksize 4×8		Blocksize 4×4	
		BPMI	ISDI	BPMI	ISDI	BPMI	ISDI
Leua (512×512)	SOPM	527.05	404.35	467.90	398.79	433.22	337.73
	Surface fit	188.36	385.90	415.84	375.38	405.20	330.05
Barbara (512×512)	SOPM	666.77	2695.70	979.52	2285.50	1788.10	1562.50
	Surface fit	570.73	3556.00	703.03	2535.60	1151.50	2177.90
Bear (512×512)	SOPM	737.51	711.16	761.78	607.40	700.26	487.01
	Surface fit	704.76	623.99	636.32	597.27	660.99	491.98
Bird (256×256)	SOPM	429.23	209.12	312.93	185.65	270.50	150.17
	Surface fit	388.41	168.04	298.22	163.57	257.19	141.68
House (256×256)	SOPM	758.57	872.20	681.49	932.19	599.03	430.73
	Surface fit	756.92	647.84	658.20	878.57	562.31	592.54
Matiface (256×256)	SOPM	540.77	209.37	411.19	190.10	312.87	161.88
	Surface fit	512.84	133.09	365.63	136.58	251.44	110.99

0	-1	0
-1	4	-1
0	-1	0

Fig. 10. Convolution mask corresponding to the Laplacian operator.

is used for devising an entropy-coding scheme for the index values. Further, the difference coding of the means of the image blocks also leads to Huffman coding of the average values.

We have reported results with three VQs using block sizes 8×8 , 4×8 , and 4×4 . Among them, as expected, VQ1 gives

the highest compression rate, but PSNR is comparatively low. On the other hand, VQ3 produces excellent quality of the reconstructed images with the lowest compression rate. VQ2 paves a middle path by achieving nice reconstruction fidelity at good compression rate. We have compared our results with standard JPEG algorithm. VQ1 is found to be superior to JPEG at comparable bit rates both in terms of PSNR and psychovisual quality. VQ2 and VQ3 scored less than the corresponding JPEG in terms of PSNR, but produced comparable psychovisual quality. We have compared our method with some published works and found that our results are superior or comparable to other spa-

tial VQs. Further, we compared our results with two recent published work using DCT and wavelets, respectively, and our results for the Lena image are comparable to them in terms of PSNR.

We proposed two indices for quantitative assessment of blockiness introduced in the reconstructed images by the VQ process. The first index, BSMI, measures the lack of continuity/smoothness at the block boundaries in the reconstructed images. The other index, ISDI, measures the deviation of the reconstructed image from the original image in terms of smoothness property of the nonboundary pixels. We have compared the images reconstructed using the codevectors generated directly from SOFM and those using the codevectors obtained by surface-fitting method. We found that the surface-fitting codebooks produce images with better psychovisual quality with respect to the blockiness.

ACKNOWLEDGMENT

The authors would like to thank the referees and the associate editor for their valuable suggestions which have helped to improve the quality of this paper considerably.

REFERENCES

- [1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
- [2] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, Dec. 1990.
- [3] S. P. Lloyd, "Least-squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Jan. 1982.
- [4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
- [5] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
- [6] N. B. Karayiannis, "Fuzzy vector quantization algorithms and their application in image compression," *IEEE Trans. Image Processing*, vol. 4, pp. 1193–1201, July 1995.
- [7] K. Zeger, J. Vaisey, and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation," *IEEE Trans. Signal Processing*, vol. 40, pp. 310–322, Feb. 1992.
- [8] N. M. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organization feature maps," in *Proc. 2nd ICNN Conf.*, vol. 1, 1988, pp. 101–108.
- [9] E. Yair, K. Zeger, and A. Gersho, "Competitive learning and soft competition for vector quantizer design," *IEEE Trans. Signal Processing*, vol. 40, pp. 394–309, Feb. 1992.
- [10] C. Amenjckx, M. Verleysen, P. Thissen, and J. Legat, "Image compression by self-organized Kohonen map," *IEEE Trans. Neural Networks*, vol. 9, pp. 503–507, June 1998.
- [11] P. C. Chang and R. M. Gray, "Gradient algorithms for designing adaptive vector quantizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 679–690, Mar. 1986.
- [12] A. B. R. Klautau Jr., "Predictive vector quantization with intrablock predictive support region," *IEEE Trans. Image Processing*, vol. 8, pp. 293–295, Feb. 1999.
- [13] R. Hamzaoui and D. Saupe, "Combining fractal image compression and vector quantization," *IEEE Trans. Image Processing*, vol. 9, pp. 197–208, Feb. 2000.
- [14] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, pp. 957–971, June 1988.
- [15] R. L. Baker and R. M. Gray, "Differential vector quantization of achromatic imagery," in *Proc. Int. Picture Coding Symp.*, 1983, pp. 105–106.
- [16] F. Kossentini, W. Chung, and M. Smith, "Conditional entropy constrained residual VQ with application to image coding," *IEEE Trans. Image Processing*, vol. 5, pp. 311–321, Feb. 1996.
- [17] M. Schnaider and A. P. Papliński, "Still image compression with lattice quantization in wavelet domain," *Adv. Imag. Electron Phys.*, vol. 119, 2000.
- [18] T. A. Ramstad, S. O. Aase, and J. H. Husøy, *Subband Compression of Images: Principles and Examples*. Amsterdam, The Netherlands: Elsevier, 1995.
- [19] T. Pavlidis, *Algorithms for Graphics and Image Compression*. New York: Springer-Verlag, 1982.
- [20] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high compression image coding," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 1306–1336, Aug. 1987.
- [21] L. Shen and R. M. Rangayyan, "A segmentation based lossless image coding method for high resolution medical image compression," *IEEE Trans. Med. Imaging*, vol. 16, pp. 301–307, Feb. 1997.
- [22] S. Biswas, "Segmentation based compression for gray level images," *Pattern Recognit.*, vol. 36, pp. 1501–1517, 2003.
- [23] K. Sayood, *Introduction to Data Compression*, 2nd ed. San Mateo, CA: Morgan-Kaufmann, 2000.
- [24] D. A. Huffman, "A method of construction of minimum redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, 1952.
- [25] P. C. Cossman, R. M. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Trans. Image Processing*, to be published.
- [26] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.



Arijit Laha received the B.Sc. (with honors) and M.Sc. degrees in physics from the University of Burdwan, Burdwan, India, in 1991 and 1993, respectively, and the M.Tech. degree in computer science from the Indian Statistical Institute (ISI), Calcutta, in 1997.

From August 1997 to May 1998, he was with Wipro Infotech Global R&D as a Senior Software Engineer. From May 1998 to December 1999, he worked on a real-time expert system development project at ISI. Currently, he is a Lecturer at the

National Institute of Management, Calcutta. His research interests include pattern recognition, data compression, neural networks, fuzzy systems, and expert systems.



Nikhil R. Pal received the B.Sc. degree in physics and the M.S. degree in business management from the University of Calcutta, Calcutta, India, in 1978 and 1982, respectively, and the M.Tech. and Ph.D. degrees in computer science from the Indian Statistical Institute (ISI), Calcutta, in 1984 and 1991, respectively.

Currently, he is a Professor in the Electronics and Communication Sciences Unit, ISI. He has coauthored a book titled *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*

(Norwell, MA: Kluwer, 1999), co-edited two volumes titled "Advances in Pattern Recognition and Digital Techniques" in ICAPRDT'99 and "Advances in Soft Computing" in AFSS 2002, and edited a book titled *Pattern Recognition in Soft Computing Paradigm* (Singapore: World Scientific, 2001). He serves the editorial/advisory board of the *International Journal of Fuzzy Systems*, *International Journal of Approximate Reasoning*, *International Journal of Hybrid Intelligent Systems*, *Neural Information Processing—Letters and Reviews*, *International Journal of Knowledge-Based Intelligent Engineering Systems*, and *Iranian Journal of Fuzzy Systems*. He is an Area Editor of *Fuzzy Sets and Systems* and a Steering Committee Member of the journal *Applied Soft Computing*. He is an Independent Theme Chair of the World Federation of Soft Computing and a Governing Board Member of the Asia Pacific Neural Net Assembly. He was the Program Chair of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, Calcutta, and the General Chair of 2002 AFSS International Conference on Fuzzy Systems, Calcutta. He is the General Chair of the 11th International Conference on Neural Information Processing, ICONIP 2004. His research interests include image processing, pattern recognition, fuzzy sets theory, measures of uncertainty, neural networks, evolutionary computation, and fuzzy logic controllers.

Dr. Pal serves the editorial/advisory board of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS.



Bhabatosh Chanda was born in 1957. He received the B.E. degree in electronics and telecommunication engineering and the Ph.D. degree in electrical engineering from the University of Calcutta, Calcutta, India, in 1979 and 1988, respectively.

He was with the Intelligent System Laboratory, University of Washington, Seattle, as a Visiting Faculty from 1995 to 1996. He has published more than 70 technical articles in international refereed journals. He is the Secretary of Indian Unit for *Pattern Recognition and Artificial Intelligence*. His

research interests include image processing, pattern recognition, computer vision, and mathematical morphology. He is currently a Professor at the Indian Statistical Institute, Calcutta.

Dr. Chandra is a Fellow of the Institution of Electronics and Telecommunication Engineers and the National Academy of Sciences, India. He received the "Young Scientist Medal" from the Indian National Science Academy in 1989, the "Computer Engineering Division Medal" from the Institution of Engineers (India) in 1998, and the "Vikram Sarabhai research Award" from the Physical Research Lab in 2002. He is also the recipient of a UN fellowship and Diamond Jubilee fellowship from the National Academy of Sciences, India.