Noncausal Predictive Image Codec

Nikhil Balram and José M. F. Moura, *Fellow, IEEE*

Abstract— **The paper describes a lossy image codec that uses a noncausal (or bilateral) prediction model coupled with vector quantization. The noncausal prediction model is an alternative to the causal (or unilateral) model that is commonly used in differential pulse code modulation (DPCM) and other codecs with a predictive component. We show how to obtain a recursive implementation of the noncausal image model without compromising its optimality and how to apply this in coding in much the same way as a causal predictor. We report experimental compression results that demonstrate the superiority of using a noncausal model based predictor over using traditional causal predictors. The codec is shown to produce high-quality compressed images at low bit rates such as .375 b/pixel. This quality is contrasted with the degraded images that are produced at the same bit rates by codecs using causal predictors or standard discrete cosine transform/Joint Photographic Experts Group-based (DCT/JPEG-based) algorithms.**

I. Introduction

DIGITAL image compression is an area of increasing importance. Many software applications manipulate still images along with text and graphics. Such applications require compression algorithms providing good visual quality at very low bit rates. Requirements of compression ratios of 20 : 1 (0.4 b/pixel) or more are not uncommon. At such low bit rates, the block discrete cosine transform-based (DCT-based) Joint Photographic Experts Group (JPEG) standard algorithm exhibits severe degradation that appears unacceptable on high-resolution monitors. This offers an opportunity for new still image codecs that provide good visual quality at low bit rates. The paper describes a new image codec. The codec couples a noncausal predictor with vector quantization (VQ). The results of the paper show that noncausal prediction is better suited to model images than causal prediction. The paper is intended to demonstrate the usefulness of the noncausal predictive processing stage as a preprocessor to vector quantization type schemes. The new noncausal-VQ image codec produces images compressed at rates below 0.4 b/pixel whose quality is better than that delivered by current standards like JPEG. The novelty and superior performance of the codec is due to the new predictive coding paradigm.

Traditional predictive coding schemes such as differential pulse code modulation (DPCM) describe the image in terms of a causal or unilateral model. By subtracting a minimum mean square error (MMSE) prediction of each pixel value from its actual value, DPCM aims at generating an uncorrelated error field. Typically, a three neighbor third-order Markov mesh [1], shown in Fig. 1(a), is used. It is described as the causal autoregressive (AR) field

$$x_{i,j} - (\beta_h^c x_{i,j-1} + \beta_v^c x_{i-1,j} + \beta_d^c x_{i-1,j-1}) = w_{i,j} \qquad (1)$$

driven by white Gaussian noise, which corresponds to the MMSE prediction error. Such models are used for practical reasons, in particular the algorithmic convenience afforded by the recursiveness of the model. However, causality imposes an arbitrary and artificial directionality on the image. For example, the model in Fig. 1(a) assumes a north, west, and northwest dependence for the pixel values. It is well known that the use of causal models introduces streaking in the processed images [1]. In practice, images are usually noncausal or bilateral in nature; in other words, a pixel value will depend on its neighbors in all directions. Fig. 1(b) shows a first-order noncausal model.

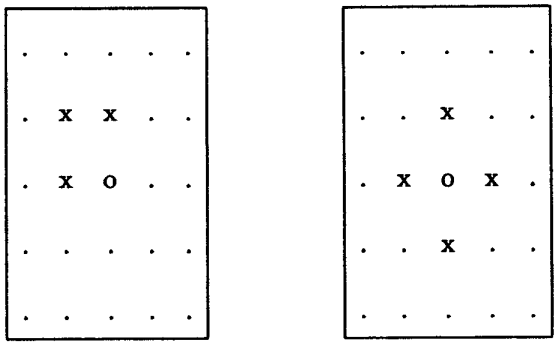


Fig. 1. Image models: The pixels marked by "x" are the only ones that affect the value of the pixel at "o." In *causal* models such as in (a), the dependence is from one direction only. In contrast, in *noncausal* models such as in (b), the dependence is from all directions.

Our codec uses a noncausal image model. The key to our approach is an information-preserving transformation that converts the noncausal model into an equivalent unilateral representation. The equivalent unilateral transformation allows the use of Kalman–Bucy-type recursive algorithms that are used with causal models. This affords all the advantages of recursive processing while retaining the optimality of the noncausal model. The noncausal predictive (NCP) component is coupled with a VQ to obtain low bit rates. The combination is referred to in the paper as NCP-VQ. The VQ component that we use in the results reported here with NCP-VQ is the VQ algorithm as described in [2]. It can be replaced

Manuscript received February 16, 1994; revised January 17, 1996. This work was supported by ONR Grant N00014–91–J–1001 and a Bellcore Grant through INI.

N. Balram is with Kaiser Electronics, San Jose, CA 95134 USA (e-mail: balramn@kaisere.com).

J. M. F. Moura is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: moura@ece.cmu.edu).

Publisher Item Identifier S 1057-7149(96)05263-3.

by more sophisticated or faster forms of VQ, for example, see that of [3] or [4], which will provide better tradeoffs between compression ratio and image quality than the VQ we incorporate in our scheme. The major novelty (and the reason for the high performance) in the NCP-VQ codec is the noncausal predictive component.

We contrast the results of a predictive coder with a causal model with the results obtained when the causal model is substituted by a noncausal model. The images produced by the causal model show directional streaking while those produced by the noncausal model display no obvious artifacts. We then compare NCP-VQ with three codecs. The first is a causal model based predictor coupled with VQ. The second is VQ alone. The third is the JPEG baseline standard. These four codecs are applied to compress an image to .375 b/pixel, which corresponds to a compression ratio of 23.

Our experimental results demonstrate clearly the visual superiority of NCP-VQ over the three competing alternatives. NCP-VQ has no objectionable visual artifacts. In contrast, the traditional causal prediction model coupled with VQ shows severe streaking artifacts in the reconstructed image. VQ alone, without the NCP component, and JPEG produce visible blocking and other unsatisfactory artifacts.

Quantitatively, causal prediction with VQ exhibits a loss over NCP-VQ of 4 to 7 dB in peak signal-to-noise ratio (PSNR). NCP-VQ has a small loss of .6 dB with regard to VQ alone and JPEG. With respect to entropy, NCP-VQ reduces more significantly the entropy of the original image (from 7.6 to 4) than JPEG (with entropy of 5) or VQ alone (7.2).

The paper is organized as follows. The remainder of the section overviews the codec. Sections II and III present the theoretical foundations of the noncausal predictor. Section IV describes the algorithm with further details in the appendices. Section V discusses the experimental results. Section VI summarizes the conclusions.

Codec Structure: The compression or coding procedure has a predictive component followed by a quantization component. The predictive component has three stages (see Fig. 2). In the first stage, the model parameters contained in the model matrix A are estimated from the image (X) after removal of the deterministic component (DC) or mean level of the image. In the second stage, the noncausal model is converted to an equivalent recursive or unilateral form, represented either by U or L in Fig. 2. The recursive form is obtained by Cholesky factorization of A, either using an upper/lower triangular factorization $(A = UU^T)$ or a lower/upper triangular factorization (LL^T). In the third stage, the recursive equivalent model is used to "whiten" the image data, i.e., to generate an uncorrelated error field (W) for the image. The error field is quantized and possibly entropy coded. To obtain low bit rates, we use VQ.

The decoder is also shown in Fig. 2. The error image is reconstructed and input to a state variable representation of the model that outputs the reconstructed image. The state variable representation is obtained by converting the noncausal model to its equivalent recursive form exactly as done in the coder.

The key to the approach is the transformation going from a noncausal model to the equivalent unilateral representation.

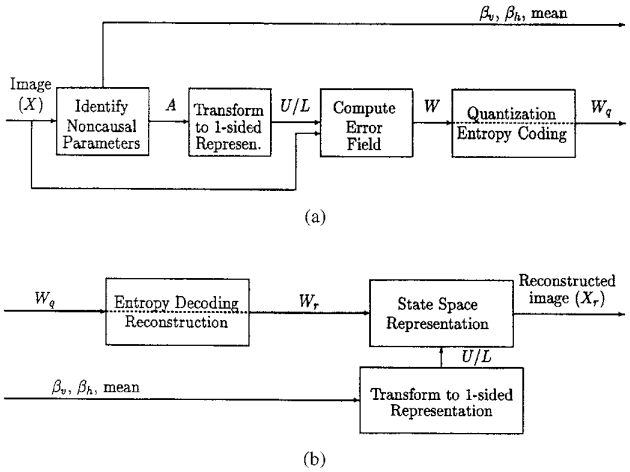


Fig. 2. Codec block diagram. (a) Transmitter. (b) Receiver.

This differs from assuming from the start a causal or unilateral model for the image. The noncausal model is parsimonious, i.e., the number of its free parameters is small and the model is derived from first principles. On the other hand, the equivalent unilateral model that is derived from it is of high order, i.e., has a large number of parameters. These parameters are highly constrained, the number of their degrees of freedom being the number of parameters in the original noncausal model.

From the unilateral representation, we derive a state variable model for the image to which we apply Kalman–Bucy filtering-type (KBF-type) recursive algorithms. Our approach makes use of KBF without compromising the basic noncausal or bidirectional nature of the two-dimensinal (2-D) spatial data.

II. Recursive Framework

The major theoretical ideas underlining the derivation of a recursive framework that is equivalent to the original noncausal model are presented in [5]. The approach is statistical. The image is modeled as a Gauss–Markov random field (GMRF). Markov fields have a local structure that is captured by the notion of neighborhood. Fig. 3 shows the neighborhood sets up to order five for pixel marked "o," where the neighbors for a given order include all the neighbors of lower order.

For the most part in this section, for the sake of clarity, we succinctly describe the noncausal prediction step in the simpler context of first-order spatially invariant models on square finite lattices with zero boundary conditions (bc) of the Dirichlet type. This will be made clear below.

A. MMSE Representation

The image[1] is modeled as a noncausal GMRF. Basically, this says that the image is represented by a bilateral autoregressive linear model driven by a correlated input noise. This is the minimum mean square error (MMSE) representation derived by Woods [6].

[1] Recall that by image we mean the signal obtained after the preprocessing stage of mean removal.

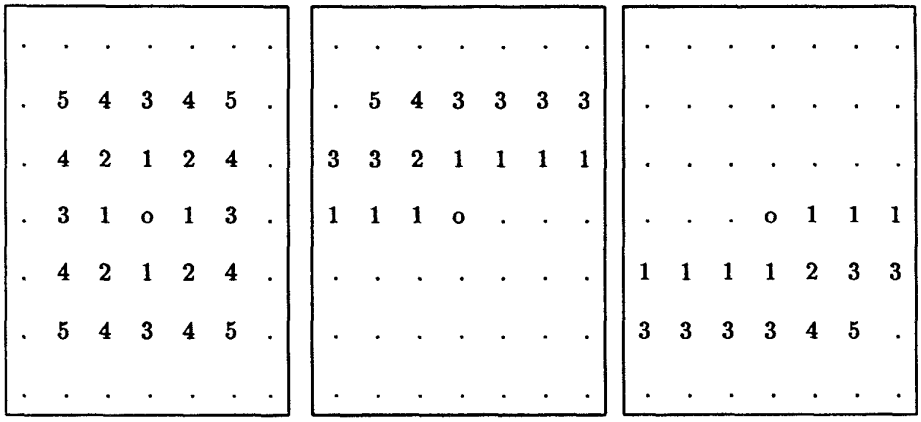


Fig. 3. Equivalent neighborhood structures (for the pixel marked by "o"). (a) Noncausal of order 1–5, each subsumes the lower order neighbors, i.e., the ith order includes all sites labeled "i" or lower. (b) Equivalent forward causal models. (c) Equivalent backward causal models.

To be specific, we introduce needed notation. The image is defined on a $N \times M$ square lattice $\mathcal{L}$

$$\boldsymbol{X} = \{x_{ij}\colon (i,j) \in \mathcal{L}\}$$

where x_{ij} is the value of the image attribute in pixel (i,j). We restrict attention to a single image attribute, the gray intensity level. As customary in image processing [1], we stack the intensity levels of the pixels in row i in the column vector $\boldsymbol{x}^i$ and then stack these in a NM-dimensional vector $\boldsymbol{x}^T$

$$\boldsymbol{x}^i = \left[x x_{i,1} \vdots x_{i,M}\right]^T \quad \text{and} \quad \boldsymbol{x} = \left[\boldsymbol{x}^{1^T} \cdots \boldsymbol{x}^{N^T}\right]^T. \tag{2}$$

Woods' MMSE approach represents a first-order image by

$$x_{i,j} + \beta_v(x_{i-1,j} + x_{i+1,j}) + \beta_h(x_{i,j-1} + x_{i,j+1}) = \epsilon_{i,j} \tag{3}$$

where β_v and β_h are the model parameters and $\epsilon_{i,j}$ is a zero mean correlated noise with

$$E[\epsilon_{i,j}\epsilon_{k,l}] = \begin{cases} \sigma_\epsilon^2 & \text{if } k = i, l = j \\ \sigma_\epsilon^2\beta_v & \text{if } k = i \pm 1, l = j \\ \sigma_\epsilon^2\beta_h & \text{if } k = i, l = j \pm 1 \\ 0 & \text{otherwise.} \end{cases}$$

On finite lattices, as in the problem under study, (3) is completed with bc's. We choose bc's that match the order of the model and stack all N equations corresponding to the N rows of the finite lattice. Using vector notation and collecting in an $NM \times NM$ matrix A the coefficients β_v and β_h, the MMSE representation of the finite lattice GMRF is written as

$$A\boldsymbol{x} = \epsilon \tag{4}$$

where $\epsilon = [(\epsilon^1)^T \cdots (\epsilon^N)^T]^T$ is the NM-dimensional vector of the random noise samples $\epsilon_{i,j}$.

B. Noncausal Canonical Representation

In [5], the following theorem is proved. We state the theorem in its simplest form of first-order spatially invariant (homogeneous), nondegenerate (strictly positive definite covariance matrix) GMRF's with zero Dirichlet bc's. See [5] for the full details.

Theorem 2.1: Equation (4) represents a zero-mean, first-order, nondegenerate, noncausal GMRF with zero Dirichlet bc's iff

$$A = A^T = \begin{bmatrix} B & C & \underline{0} & \cdot & \cdot & \cdot \\ C & B & C & \underline{0} & \cdot & \cdot \\ \underline{0} & C & B & C & \underline{0} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \underline{0} \\ \cdot & \cdot & \underline{0} & C & B & C \\ \cdot & \cdot & \cdot & \underline{0} & C & B \end{bmatrix} > 0$$

$$B = \begin{bmatrix} 1 & \beta_h & 0 & \cdot & \cdot & \\ \beta_h & 1 & \beta_h & 0 & \cdot & \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \beta_h & 1 & \beta_h & \\ \cdot & \cdot & 0 & \beta_h & 1 & \end{bmatrix} \tag{5}$$

$$C = \beta_v I, \quad \text{Cov}\,\epsilon = \sigma_\epsilon^2 A, \quad \text{and} \quad E\boldsymbol{x}\epsilon^T = \sigma_\epsilon^2 I. \tag{6}$$

In (5), $\underline{0}$ are zero matrix blocks. Further, the covariance $\Sigma_{\boldsymbol{x}}$ of the field $\boldsymbol{x}$ is

$$\Sigma_{\boldsymbol{x}} = \sigma_\epsilon^2 A^{-1}. \tag{7}$$

Theorem 2.1 provides for first-order fields a simple, albeit powerful, representation in terms of three parameters $\{\sigma_\epsilon^2, \beta_h, \beta_v\}$. The quantities β_h and β_v are the interaction or potential parameters. The matrix A is highly structured, sparse, block tridiagonal. The Toeplitz structure is a consequence of the field homogeneity and of the Dirichlet bc. For other bc's, A is a tridiagonal block slightly perturbed Toeplitz matrix. Matrix A is referred to as the potential matrix. In contrast with one-dimensional (1-D) signal processing, where the covariance matrix is usually the description of choice, the theorem states that with 2-D data the appropriate object to work with is the potential matrix A, not the field covariance $\Sigma_{\boldsymbol{x}}$.

For models of order p, the number of interaction or potential parameters $\{\beta_\tau\}$ increases. For example, with a general (2-D) model there will be two additional diagonal interaction parameters β_{d_1} and β_{d_2}; see Fig. 3 for the structure of a

second-order field. We indicate the interaction parameters of a pth-order model by $\{\beta_\tau : \tau \in \eta_p\}$ where η_p is the neighborhood set.

C. Equivalent Unilateral Representation

The next theorem introduces the recursive or unilateral equivalent representation.

Theorem 2.2: The following are two equivalent representations for a zero-mean, first-order, nondegenerate, noncausal GMRF with zero Dirichlet bc.

1) Noncausal representation

$$A\boldsymbol{x} = \epsilon \tag{8}$$

where A, ϵ are given in Theorem 2.1.

2) Causal representation

$$U^T\boldsymbol{x} = \boldsymbol{w} \tag{9}$$

where

$$U = \begin{bmatrix} U_1 & \Theta_1 & \underline{0} & \cdot & \cdot \\ \underline{0} & U_2 & \Theta_2 & \underline{0} & \cdot \\ \cdot & \underline{0} & U_{N-1} & \Theta_{N-1} & \underline{0} \\ \cdot & \cdot & \underline{0} & U_N & \cdot \end{bmatrix} \tag{10}$$

$$U_i^T U_i = S_i \quad \text{and} \quad U_i^T \Theta_i = C \tag{11}$$

and $\underline{0}$ are zero matrix blocks of appropriate dimensions.

The S_i are obtained through the successive iterates of the following Riccati[2] iterative scheme.

$$S_1 = B, S_i = B - C^T S_{i-1}^{-1} C \qquad 2 \le i \le N. \tag{12}$$

The noise vector is white

$$\boldsymbol{w} = [(\boldsymbol{w}^1)^T \cdots (\boldsymbol{w}^N)^T]^T \quad with \quad \text{Cov}\ \boldsymbol{w} = \sigma_\epsilon^2 I. \tag{13}$$

The first part of Theorem 2.2 follows by Cholesky decomposition of A. What is important to note is that the Cholesky upper triangular factor U in (9) is block diagonal, where the blocks U_i and Θ_i are themselves upper and triangular matrices, respectively.

Expand the two representations (8) and (9) in terms of the row intensity vectors. The noncausal model in (8) and the causal model of (9) lead to

$$C\boldsymbol{x}^{i-1} + B\boldsymbol{x}^i + C\boldsymbol{x}^{i+1} = \epsilon^i \tag{14}$$

$$U_i\boldsymbol{x}^i + \Theta_i\boldsymbol{x}^{i+1} = \boldsymbol{x}^i \tag{15}$$

respectively. Equation (14) expresses the vector $\boldsymbol{x}^i$, i.e., the intensities of the ith row, as a linear combination of the ("past") vector $\boldsymbol{x}^{i-1}$ and the ("future") vector $\boldsymbol{x}^{i+1}$. We refer to it as a noncausal or bilateral regressor. On the other hand, (15) is row-wise unilateral or (anti-)causal, expressing $\boldsymbol{x}^i$ simply in terms of $\boldsymbol{x}^{i+1}$. We refer to it as a causal or unilateral regressor. Further, the matrices U_i and Θ_i are lower triangular. This means that the pixel-wise regressor equations in (15) are Nth-order unilateral AR models. This contrasts with the AR Markov mesh model in Fig. 1, which is of much lower order. However, the N coefficients of the AR models (15) have only two degrees of freedom (β_h and β_v).

The causal regressor is spatially varying, i.e., the matrices U_i and Θ_i are different from row to row. In [7], it is proved that, under suitable conditions, the Riccati iteration (12) converges, $S_i \to S_\infty$, where S_∞ is a solution of the steady-state equation

$$S_\infty = B - C^T S_\infty^{-1} C \tag{16}$$

$$\|S_i - S_\infty\|_s \le K_0 a^i, \quad \forall i \ge 2 \tag{17}$$

where $\|\cdot\|_s$ is the spectral norm and K_0 and $0 < a < 1$ are constants. Equation (17) shows that the convergence is at a geometric rate.

From the point of view of computer implementation, the geometric rate of convergence of the iteration is highly significant. It says that after a small number n of steps, $n \ll N$, it can be safely assumed that

$$i \ge n: \quad S_i \approx S_\infty = U_\infty^T U_\infty, \quad U_i \approx U_\infty$$
$$\Theta_i \approx \Theta_\infty, \quad U_\infty^T \Theta_\infty = C. \tag{18}$$

D. State Variable Framework

In order to enable the use of Kalman–Bucy-type recursive algorithms, we provide a state variable framework for the 2-D image. We derive from (15) the following state variable model for the image.

Backward State Variable Representation:

$$\boldsymbol{x}^N = G_N^b \boldsymbol{w}^N,$$
$$\boldsymbol{x}^i = F_i^b \boldsymbol{x}^{i+1} + G_i^b \boldsymbol{w}^i, \qquad 1 \le i \le N-1. \tag{19}$$

The model matrices for the state space representation are given by

$$G_i^b = U_i^{-1}, \quad \text{and} \quad F_i^b = -U_i^{-1}\Theta_i = -G_i^b (G_i^b)^T C. \tag{20}$$

From Theorem 2.2, the statistics of the driving noise are

$$E(\boldsymbol{w}^i \boldsymbol{w}^{j^T}) = \sigma_\epsilon^2 I_M \delta_{i,j},$$
$$E(\boldsymbol{x}^i \boldsymbol{w}^{i^T}) = \sigma_\epsilon^2 U_i^{-1} \quad \text{for} \quad 1 \le i \le N \tag{21}$$

where $\delta_{i,j}$ is the Kronecker symbol. In other words, the driving noise $w_{i,j}$ is white and uncorrelated with the image variables that lie ahead of it (in a backward sense) in the lexicographic ordering.

There is a second equivalent state variable model, a forward model, that is derived from the lower triangular Cholesky factorization of A.

Equation (19) provides a quick way of generating sample functions of a GMRF for given values of the interaction parameters. We determine the model matrices F_i^b and G_i^b as given by (20), where the U_i and Θ_i are found from Theorem 2.2, and drive the state variable model with a Gaussian sequence of zero-mean independent, identically distributed (i.i.d.) random variables w_{ij} of variance σ_ϵ^2.

[2] We refer to (12) as a Riccati iteration due to its analogy with the algebraic quadratic equation in Kalman–Bucy filtering.

III. Model Fitting

In predictive coding, the first stage fits a parsimonious model to the image. By describing the image as a GMRF, this corresponds to determining from the image of interest the parameters $\theta = \{\beta_\tau, \tau \in \eta_p\}$ as well as the noise power level σ_ϵ^2. We apply maximum likelihood (ML) estimation. ML estimation maximizes the (negative) log-likelihood function (LLF). Fast ML estimation of the field parameters has been considered in [7]. The expression of the LLF follows from the Gaussian assumption on the field

$$L(X/\theta, \sigma_\epsilon^2) = \frac{1}{2}\ln\sigma_\epsilon^2 - \frac{1}{2NM}\ln|A_p(\theta)| + \frac{1}{2\sigma_\epsilon^2 NM}\vec{X}^T A_p(\theta)\vec{X} \tag{22}$$

where A_p is the potential matrix of a pth-order model. The second term in (22) is the "model" term. It corresponds to the normalizing constant of the multivariate Gaussian probability density function. It is the partition function in statistical mechanic's models. The third term is the "data"-driven term.

In general, to maximize $L(X/\theta, \sigma_\epsilon^2)$ is no trivial task, involving a large number of evaluations of the determinant $|A_p|$ of the potential operator A_p, an $N^2 \times M^2$ matrix. Further, the parameters β_τ are constrained to a bounded region in the parameter space Θ where A_p is positive definite.

To search efficiently the parameter space, we use the conjugate gradient method, e.g., [8]. In conjugate gradient (or any other gradient-based procedure), evaluation of the gradient $\nabla L(\cdot)$ is required at several points in the search space. We compute ∇L for arbitrary order noncausal GMRF's by exploiting the recursive structure of these models as explained below.

Arbitrary-Order Models: Reworking (22), we obtain

$$L(\theta, \sigma_\epsilon^2) = \frac{1}{2}\ln\sigma_\epsilon^2 - \frac{1}{NM}\sum_i^N \ln|U_i| + \frac{1}{2\sigma_\epsilon^2}\left[S_x - 2\sum_{\tau\in\eta_p}\beta_\tau \mathcal{X}_\tau\right] \tag{23}$$

where the U_i's are the blocks on the diagonal of the Cholesky factor (10), S_x is the sample power, and $\mathcal{X}_\tau$ is the sample correlation of lag τ. The last two quantities are computed from the image by (34) and (35) in Appendix A, respectively.

The noise power level σ_ϵ^2 is explicitly determined in terms of the other quantities by differentiating the LLF with respect to it and equating to zero. We get

$$\sigma_\epsilon^2(\theta) = S_x - 2\sum_{\tau\in\eta_p}\beta_\tau \mathcal{X}_\tau. \tag{24}$$

Given a set of values $\{\beta_\tau, \tau \in \eta_\tau\}$, the U_i's in (23) are computed by the Riccati iteration (11) and (12). The geometric convergence of the Riccati iteration significantly reduces the cost of evaluating $L(\cdot)$. We now present a novel way of computing the gradient.

Result 3.1: The gradient of the LLF with respect to the interaction parameters is given by

$$\nabla L = [\nabla L_\tau : \tau \in \eta_p] \tag{25}$$

$$\forall \tau \in \eta_p: \quad \nabla L_\tau = \frac{E\{\mathcal{X}_\tau\} - \mathcal{X}_\tau}{\sigma_\epsilon^2}. \tag{26}$$

The proof is in [7]. To compute the gradient in an efficient way, we use this result and exploit the recursive structure of the GMRF; see Appendix A for details.

For first-order and some higher order fields, we are able to parameterize the function $L(\cdot)$ and its gradient explicitly, thereby avoiding the need for using the algorithm described in Appendix A.

First-Order Models: First-order GMRF's have only two interaction parameters, β_h and β_v, the horizontal and vertical nearest neighbor interactions (see Fig. 3). In [7], we compute analytically the partition function for this case (and certain higher order models). The likelihood function is

$$L(X/\beta_h, \beta_v, \sigma_\epsilon^2) = \frac{1}{2}\ln\sigma_\epsilon^2 - \frac{1}{2NM}\sum_{i=1}^N\sum_{j=1}^M \ln(1 - \beta_v\lambda_i(\mathcal{S}_N) - \beta_h\lambda_j(\mathcal{S}_M)) + \frac{1}{2\sigma_\epsilon^2}[S_x - 2\beta_h\mathcal{X}_h - 2\beta_v\mathcal{X}_v] \tag{27}$$

with S_x the sample power, and $\mathcal{X}_h$ and $\mathcal{X}_v$ the horizontal and vertical sample correlations given by (34) and (36) in Appendix A, respectively.

The matrices $\mathcal{S}_K (K = N, M)$ depend on the choice of bc's. The eigenvalues of these matrices are computed prior to the conjugate gradient search using the functional forms derived in [7] for each bc. For example, for zero (free or Dirichlet) bc, the eigenvalues of $\mathcal{S}_K$ are given by

$$\lambda_k(\mathcal{S}_K) = 2\cos\frac{k\pi}{K+1} \qquad 1 \le k \le K. \tag{28}$$

The conjugate gradient search uses these expressions to compute explicitly the gradient ∇L [see (39) and (40) in Appendix A].

To alleviate the computational effort, we indicate an approximate parameter estimate that we have used without noticeable degradation in performance.

Approximate Estimates: For many images, the first-order GMRF with zero bc performs satisfactorily. For square (i.e., $N = M$) images that are locally smooth, there is a simple approximate relationship that provides directly an approximate minimum point at which the LLF is within a few percent of its minimum value, without requiring an iterative search of the parameter space.

For Dirichlet boundary conditions, the parameter space is defined by the inequality

$$|\beta_v| + |\beta_h| < \frac{1}{2\cos\frac{\pi}{M+1}}. \tag{29}$$

For numerical stability, we use a tolerance κ on the right-hand side of (29), and define the boundary of the parameter search

space as

$$\varsigma = \frac{1}{2\cos\frac{\pi}{M+1}} - \kappa. \tag{30}$$

The approximate parameter estimates are given by

$$\beta_h = \frac{\varsigma \mathcal{X}_h}{|\mathcal{X}_h| + |\mathcal{X}_v|} \quad \text{and} \quad \beta_v = \frac{\varsigma \mathcal{X}_v}{|\mathcal{X}_h| + |\mathcal{X}_v|} \tag{31}$$

where $\mathcal{X}_h$ and $\mathcal{X}_v$ are the horizontal and vertical sample correlations as introduced before. These can be derived via a Taylor series approximation. They reflect the fact that, intuitively, the ratio of the interaction coefficients should be equal to the ratio of the sample correlations and the fact that for fields that are smooth (as most images are) the minimum lies very close to the parameter space boundary. This approximation is used in the image shown in Fig. 6(b).

IV. Algorithm

The codec is described in detail using the equations summarized in the previous two sections. The coder is considered in Subsection IV-B and the decoder in Subsection IV-A. Specialized tasks are presented in pseudocode in Appendix B. The dependence of the codec on the model order is through these blocks. At the block level in Subsections IV-A and IV-B, the procedure is independent of the model order.

A. Coder

We detail the structure of the coder. The coder takes the given image X as input and produces a compressed representation W_q plus overhead information (m, θ) pertaining to the image model. The specialized subroutines referred to below are in Appendix B.

1) *Parameter Estimation.* The global mean (m) of the image is subtracted out, and the noncausal model parameters θ are estimated for the zero-mean image. There are two steps, the setup and the iterative search. For a first-order model with zero (also known as free or Dirichlet) bc's, for practical reasons the iterative search may be replaced by the fast approximation outlined in Section III. In this case, Step 2 is replaced by the subroutine *approxest* $(\cdot)$ given in Appendix A.
   a) *Step 1: Setup.* Compute: i) global image mean $m = 1/NM \sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j}$; ii) zero mean image $X := X - m$; iii) sample power S_x using (34), and sample correlations $\{\mathcal{X}_\tau, \tau \in \eta_p\}$ using (35) for arbitrary order models or (36) for first order.
   b) *Step 2: Iterative Search.* The parameter space of θ is searched iteratively along a sequence of conjugate line directions $(D^{(i)})$ chosen according to the Polak–Ribiere conjugate gradient method; see, for example, [8, Sec. 10.6]. Each iteration i finds the point $\theta^{(i+1)}$ at which the (negative log) likelihood function $L(\cdot)$ is minimized on the line specified by the previous minimum point $\theta^{(i)}$ and conjugate direction $D^{(i)}$. The iteration proceeds until the change in function value between two consecutive minimum points is below a user-specified tolerance ϵ (see Appendix A for details).
2) *Transformation.* Given the noncausal parameter estimate vector $\hat{\theta}_{ML}$ (output of the previous stage), the subroutine riccati$(\cdot)$ computes the equivalent recursive representation matrices $\{U_i, \Theta_i\}$.
3) *Generation.* The uncorrelated error image W is generated recursively one row at a time, starting with the last $(N\text{th})$ row, using the equivalent "backward" innovations row process (15). The innovations process matrices $\{U_i, \Theta_i\}$ were computed in the previous stage.
4) *Quantization.* Any quantizer may be used to quantize the error image (W). In order to obtain bit rates of 0.5 and lower (corresponding to compression ratios of 16 or better), we use a VQ employing the Linde–Buzo–Gray (LBG) algorithm [2]. The quantizer output may be entropy coded (in our experiments reported in Section V we have not done this) to obtain additional lossless compression. The output of this stage is the quantized and entropy coded field W_q. Details on the VQ actually used in the experiments of Section V are given in that section.

Output of Coder: The output is a compressed representation of the image consisting of the quantized entropy-coded error field W_q. The noncausal model parameters (θ, m) are overhead information, negligible in comparison with the quantized error image. For example, for a first-order model, there are only three scalar overhead items, the parameters β_h and β_v, and the mean m.

B. Decoder

The decoder reconstructs the image (X_r) from the coded error image W_q and the model parameters (θ, m). The decoder is essentially the reverse of the coder, except it has no parameter estimation step.

1) *Reconstruction of Error Image.* The coded error image W_q is entropy-decoded and reconstructed according to the rules of the quantizer, producing the reconstructed error image W_r.
2) *Transformation.* Given the noncausal model parameter vector θ, the subroutine riccati$(\cdot)$ computes the equivalent recursive representation matrices $\{U_i, \Theta_i\}$.
3) *Reconstruction.* The "backward" state space model (19) is used to reconstruct the image recursively, one row at a time, starting from the last $(N\text{th})$ row, with the rows from the reconstructed error image W_r as inputs. The $\{G_i^b, F_i^b\}$ are obtained from $\{U_i, \Theta_i\}$ using (20). The global mean m is added to the reconstructed image, $X_r := X_r + m$.

C. Architecture Implementation

The architecture of the noncausal codec is simplified by taking into consideration the following observations where, for simplicity, we assume square images $(N = M)$.

1) Storing the $2N$ triangular matrices U_i and Θ_i requires $O(N^3)$ memory size. We observed that the coefficients of the equivalent unilateral representations decay ex-

TABLE I
PARAMETERS FOR MODELS USED IN FIG. 4. GLOBAL MEAN OF ORIGINAL (LENNA) IMAGE = 110.396667

Model	parameters
noncausal	$\widehat{\beta_h} = 0.103827, \widehat{\beta_v} = 0.394309, \widehat{\sigma_\epsilon^2} = 130.448801$
Approx. noncausal	$\widehat{\beta_h} = 0.246140, \widehat{\beta_v} = 0.251996, \widehat{\sigma_\epsilon^2} = 150.675292$
causal	$\widehat{\beta_h^c} = 0.666224, \widehat{\beta_v^c} = 0.863320, \widehat{\beta_d^c} = -0.556740, \widehat{\sigma_c^2} = 118.731850$

ponentially fast. By keeping only $l \ll N$ of these coefficients, the matrices U_i and Θ_i become upper and lower triangular banded matrices with band l, reducing the memory requirements to $O(N^2 l)$. Because most operations are $O(N^3)$ or $O(N^4)$, they are now reduced to being $O(N^2 l)$ or $O(N^2 p^2)$. A band of $l = 8$ for images of $N = 256$ was found not to degrade significantly the performance.

2) The Riccati iteration converges at a geometric rate. By stopping at iteration $n \ll N$, we save computation time and memory space. The memory requirements become $O(Nnl)$ and the number of operations $O(Nnl)$ or $O(Nnl^2)$.
3) For many images, see the Lenna image at .375 b/pixel in Fig. 6(b), the simple approximate estimates of (31) provide good performance, avoiding the costly ML estimation step.

V. EXPERIMENTAL RESULTS

The experiments presented here were designed to make two major points. First, we seek to emphasize the importance of using noncausal image models versus causal ones. This is the motivation behind the first set of experiments in which we couple predictive coding with a simplified setting for quantization (scalar quantizer) to contrast the effects of noncausality versus causality in the image model. In the second set of experiments, we show that good-quality results may be obtained at very low bit rates when we combine noncausal predictors with vector quantization as in NCP-VQ. These results are superior to those obtained at similar bit rates using causal predictors with VQ, VQ alone, or the DCT-based JPEG baseline system. We discuss below these results in terms of visual quality, entropy, and PSNR.

The VQ used in our experimental studies is the LBG algorithm in [2] and has been trained with the image it is quantizing. The JPEG implementation used here is from [9]. The JPEG results are after a final stage of lossless coding, not included with the other methods, the compression ratio indicated for JPEG already factoring in this additional gain.

We report results with one of the images from the USC-SIPI database [10], the 128×128 Lenna image at a resolution of 8 b/pixel. The Lenna image combines relatively homogeneous and smooth patches with highly detailed regions like the eyebrows and strands of hair.

Experiment 1: The Lenna image is compressed to 2 b/pixel using NCP followed by scalar quantization. The image was modeled as a noncausal first-order Dirichlet field and its parameters estimated using the ML procedure in Section III. The estimated values are given in the first entry of Table I. For simplicity, the uncorrelated error field was quantized to four levels or 2 b/pixel using a scalar quantizer, the optimal Lloyd–Max quantizer assuming a Gaussian density, e.g., [1]. The reconstructed image is shown in Fig. 4(b). We compared this compressed image with the result of compressing it with a causal predictor followed by scalar quantization. The causal predictor is designed assuming the image is modeled by a causal image model, the (three-neighbor) third-order Markov mesh displayed in Fig. 1. The prediction error field is generated recursively using a causal third-order AR [see (1)]. This error field is then quantized using the Lloyd–Max quantizer to the same bit rate of 2 b/pixel as above. The Markov mesh parameters were obtained using least-squares estimation (see the third entry in Table I). The reconstructed Lenna using NCP and DPCM followed by scalar quantizer are shown in Fig. 4(b) and (c), respectively.

Comparing the two reconstructed images, we see that the one produced by the causal image model exhibits considerable streaking while the one that used NCP does not. This emphasizes the superiority of noncausal image models vis-a-vis causal ones. This is true in spite of the fact that the third-order Markov mesh causal model has more free parameters to adjust than the first-order noncausal model.

Experiment 2: We coded the Lenna image with the NCP-VQ codec described in Section IV. Neglecting the small overhead due to the noncausal model parameters, the image is coded to .375 b/pixel. The corresponding compression ratio is 21.3. This bit rate is achieved with a vector quantizer based on the LBG algorithm [2] with block size 4×4 and 64 codevectors trained with the Lenna image error field.

We tested two noncausal predictor models in the NCP-VQ. In the first, the parameters of the noncausal predictor are estimated by applying the ML algorithm of Section III. They are, as before, shown by the first entry in Table I. The image quantized with this optimal ML NCP-VQ is displayed in Fig. 5(b). In the second noncausal predictor, the parameters are the estimates given by the (31) (see second entry in Table I). The results of the quantized image are displayed in Fig. 5(c).

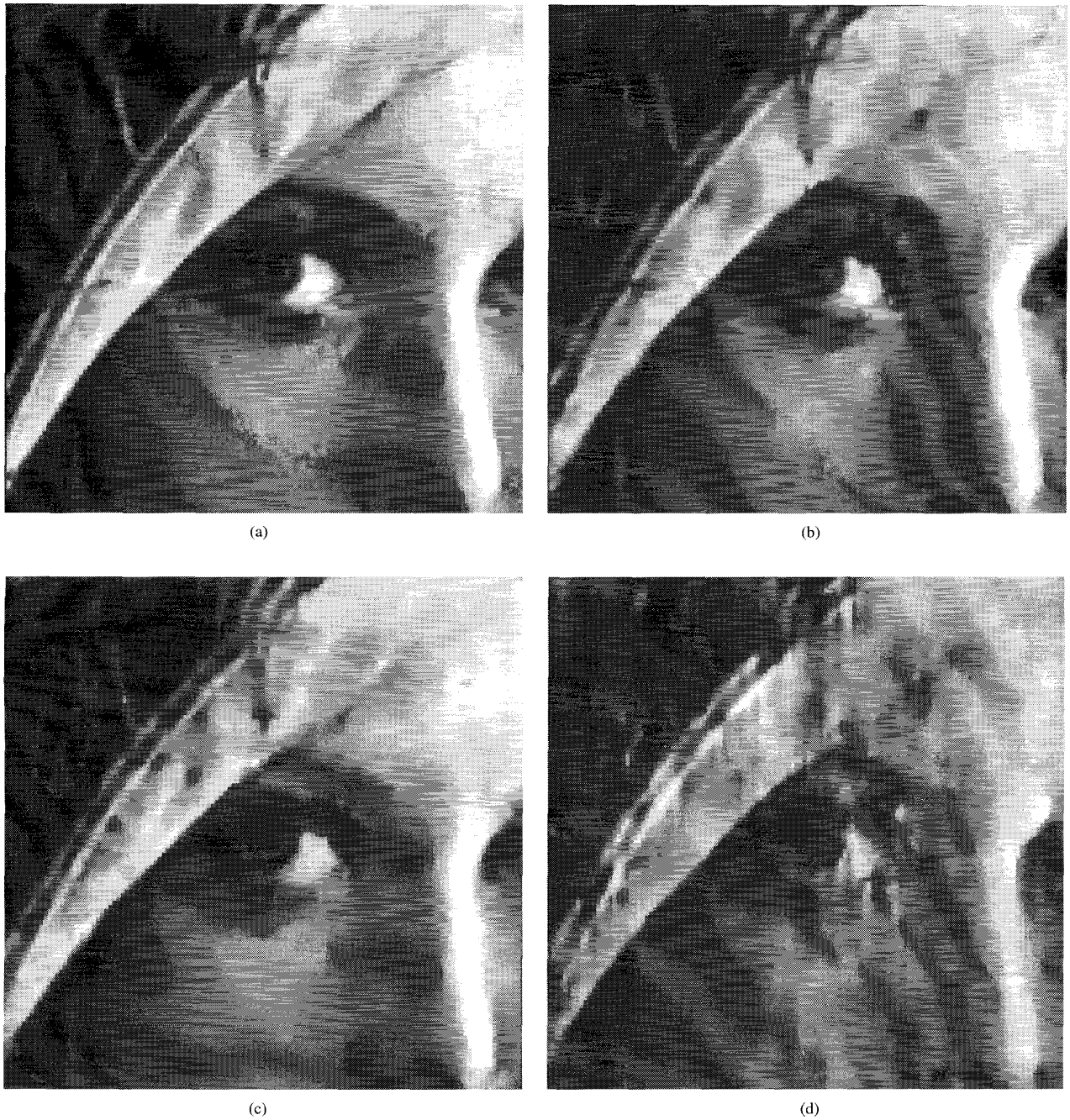
We compared these two NCP-VQ quantized Lennas with the Lenna quantized using three competing codecs. The first is causal prediction with the same implementation of VQ as in NCP-VQ, trained now on the causal error Lenna image. The

Fig. 4. (a) Original 128×128 Lenna image. (b)–(c) Reconstructed image after compression to 2 b/pixel using predictive coding and the Lloyd–Max quantizer with four levels. (b) used (noncausal) first-order Dirichlet field as image model, and (c) used (causal) third-order Markov mesh as image model. The model parameters are given in Table I.

second is VQ alone trained on the Lenna image. The third is JPEG as available from [9]. All these codecs compressed Lenna to a similar .375 b/pixel compression.

The causal-VQ result is shown in Fig. 5(d). The result of VQ alone (after subtraction of the global mean) is displayed in Fig. 6(c). The JPEG reconstructed image is provided in Fig. 6(d). The bit rates reported for JPEG include entropy coding, while for NCP-VQ they are prior to entropy coding. We expect entropy coding to provide an additional reduction of 10–15% of the bit rate reported for this experiment (0.375 b/pixel).

The results in Figs. 5 and 6 show that the image compressed with our technique [see Fig. 5(b) and (c) and Fig. 6(b)] displays good quality as well as fidelity especially considering the



(a) (b) (c) (d)

Fig. 5. (a) Original 128×128 Lenna. (b)–(d) Comparison of reconstructed Lenna after compression to 0.375 b/pixel using: (b) NCP-VQ with optimal ML parameters from Table I; (c) NCP-VQ with approximated parameters from Table I; (d) third-order Markov mesh (causal prediction) with optimal parameters from Table I.

low bit rate (0.375 b/pixel). The reconstructed image retains acceptable visual quality and most of the detail of the original, e.g., the structure of the eye and eyebrows. Moreover, there is little blocking visible in the image despite the fact that VQ is prone to introducing blocking at low bit rates. This remains true even when we use the approximate simple estimates of (31) [see Fig. 5(c)]. The causal-VQ codec exhibits severe degradation and streaking at these bit rates [see Fig. 5(d)]. VQ exhibits blocking effects as illustrated in Fig. 6(c) (VQ alone). The result obtained with JPEG, see Fig. 6(d), displays the blocking effects known to occur with this method at low bit rates. This blocking or *tiling* occurs because each 8×8 block in the image is coded independently, and therefore the reconstruction error for each block is independent. At low bit rates, the difference between the reconstruction errors of neighboring blocks causes the tiling.

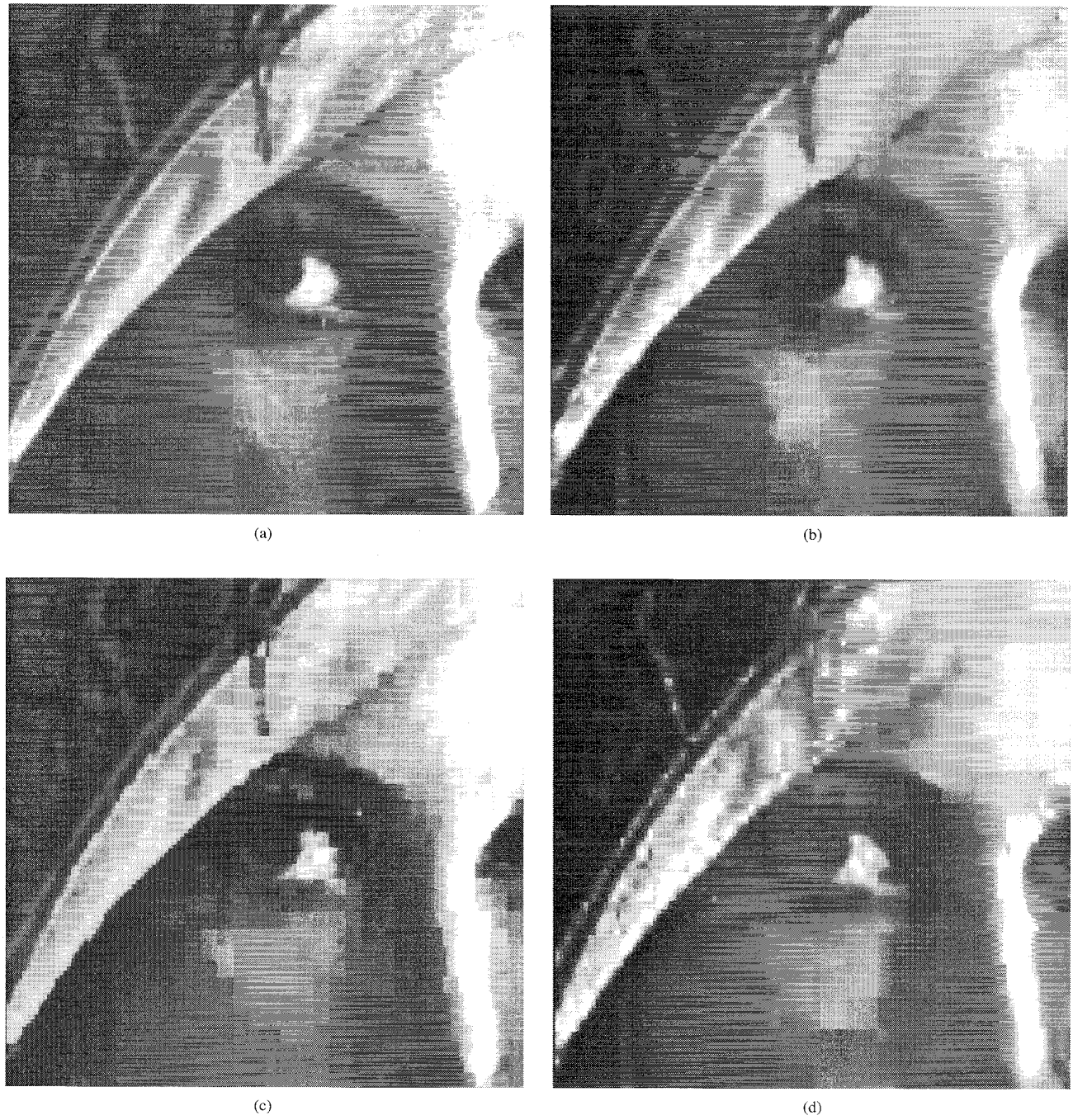


Fig. 6. (a) Original 128×128 Lenna. (b)–(d) Comparison of reconstructed Lenna after compression: (b) To 0.375 b/pixel using NCP-VQ with a first-order Dirichlet field as the noncausal image model and VQ (4×4 blocks, 64 codevectors) to quantize the error field; (c) to 0.375 b/pixel using VQ (4×4 blocks, 64 codevectors) to quantize the image with the global mean removed; (d) to 0.3765 b/pixel using JPEG (8×8 DCT) plus entropy coding.

Entropy and SNR Comparisons: Table II summarizes the entropy, MMSE, and PSNR values for the several techniques discussed before when applied to the Lenna image. In Table II, the entropies of the images compressed with the noncausal predictor and causal predictor are computed for the error images. For the JPEG compressed image, the entropy value is the entropy of the DCT transformed image.

The MSE and PSNR are evaluated according to the following expressions:

$$\mathrm{MSE} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (X(i,j) - \tilde{X}(i,j))^2 \tag{32}$$

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\mathrm{MSE}} \tag{33}$$

TABLE II
MSE, PSNR, AND ENTROPY VALUES

Images	MSE	PSNR(db)	Entropy	Description
Figure 4-(a)	–	–	7.6016	Original Image
Figure 4-(b)	278	23.7	4.07	Noncausal codec (2bpp)
Figure 4-(c)	689	19.5	4.14	Causal codec (2bpp)
Figure 5-(b)	132.5	27	4.07	ML NCP-VQ (.375 bpp)
Figure 5-(c)	144.4	26.54	4.23	Approx. NCP-VQ (.375 bpp)
Figure 5-(d)	612.2	20.26	4.14	Causal codec with VQ (.375 bpp)
Figure 6-(c)	113.7	27.6	7.18	VQ alone (.375 bpp)
Figure 6-(d)	112.4	27.62	5.05	JPEG (.3765 bpp)

where $X(i,j)$ is the original image and $\tilde{X}(i,j)$ is the reconstructed image.

The PSNR values for the noncausal predictor exceed from 4–7 dB the corresponding values for the causal predictor. A second important characteristic of the noncausal predictor is that its PSNR increases 3 dB when we replace the scalar quantizer with the VQ. This is not the case with the causal predictor for which the MSE and PSNR values are practically the same with scalar quantization and with VQ. Compared with JPEG and VQ alone, NCP-VQ exhibits a slight loss of .6 dB in PSNR, 27.6 dB for the first two, and 27 dB for NCP-VQ.

In terms of entropy, the noncausal and causal predictors with an entropy of 4 are capable of reducing more significantly the correlation of the image than JPEG with an entropy of 5 or VQ alone with an entropy of 7.

Feedback Versus Feedforward: Vector quantizers with predictive coding are usually implemented as feedforward schemes [11]. The quantizer is cascaded with the predictor instead of being in the predictor feedback loop as is usually done with scalar quantization schemes. A major strength of the noncausal predictive component is that it is robust enough to produce good-quality output at low bit rates, as demonstrated by Fig. 5(b) or Fig. 6(b). In contrast, the traditional causal predictive model is very sensitive to errors and produces very poor-quality output, as demonstrated by Fig. 5(c) if the quantizer is cascaded instead of being in the feedback loop.

VI. SUMMARY

NCP-VQ, the algorithm for image compression described in this paper, has been demonstrated to provide significant compression at a low level of degradation. It combines a new noncausal prediction model component with vector quantization. The noncausal component whitens the original image (after removal of the global mean), generating an error image that is vector quantized. The paper demonstrates how to implement the noncausal predictor component recursively making it structurally equivalent to a causal predictor, but without compromising its optimality. Even at compression ratios exceeding 20, NCP-VQ does not show any major objectionable artifacts such as the streaking that is caused by causal prediction models or the blocking caused by VQ alone or JPEG.

Besides the better visual quality results of the NCP-VQ, it leads also to better quantitative results. Its PSNR value, comparable to that of JPEG and VQ alone, exceeds in about 7 dB the PSNR of the causal predictor with VQ. In terms of entropy, the noncausal predictor reduces the entropy of the original from 7.6 to 4. This is better than the entropy of 5 of the JPEG DCT transformed image, or the entropy of 7.2 of the VQ alone.

NCP-VQ is a flexible and easily extensible algorithm. Its novel component, the NCP stage, can be coupled (as a preprocessor) with new high-quality VQ codecs to produce even better results.

APPENDIX
SAMPLE CORRELATIONS AND GRADIENT COMPUTATIONS

Sample Correlations: The sample power, S_x, and the sample correlation of lag $\tau, \mathcal{X}_\tau$, are given by

$$S_x = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}^2 \tag{34}$$

$$\mathcal{X}_\tau = \begin{cases} \dfrac{1}{NM}\displaystyle\sum_{r=1}^{N-i}\sum_{s=1}^{M} x_{r,s}x_{r+i,s} & \text{for } \tau = v_i \\ \dfrac{1}{NM}\displaystyle\sum_{r=1}^{N}\sum_{s=1}^{M-j} x_{r,s}x_{r,s+j} & \text{for } \tau = h_j \\ \dfrac{1}{NM}\displaystyle\sum_{r=1}^{N-i}\sum_{s=1}^{M-j} x_{r,s}x_{r+i,s+j} & \text{for } \tau = d_{l_{ij}} \\ \dfrac{1}{NM}\displaystyle\sum_{r=1}^{N-i}\sum_{s=j+1}^{M} x_{r,s}x_{r+i,s-j} & \text{for } \tau = d_{r_{ij}}. \end{cases} \tag{35}$$

In (35), $\tau = v_i, h_j, d_{l_{ij}}, d_{r_{ij}}$ represent the directions of the interaction parameters vertical, horizontal, left diagonal (northwest, southeast), or right diagonal (northeast, southwest), respectively. For example, $\beta_{d_{r_{ij}}}$ represents the interaction parameter along the right diagonal direction for pixel (i,j). For first-order fields, there are only two sample correlations, the horizontal $\tau = h$ and vertical $\tau = v$

$$\mathcal{X}_h = \frac{1}{NM}\left(\sum_{i=1}^{N}\sum_{j=1}^{M-1} x_{i,j}x_{i,j+1} + X_{h_1}^b\right),$$
$$\mathcal{X}_v = \frac{1}{NM}\left(\sum_{i=1}^{N-1}\sum_{j=1}^{M} x_{i,j}x_{i+1,j} + X_{v_1}^b\right) \tag{36}$$

with $\mathcal{X}_{h_1}^b$ and $\mathcal{X}_{v_1}^b$ being correction terms that depend on the choice of boundary conditions. For example, for zero Dirichlet boundary conditions, they are both zero.

Computation of the Gradient: We evaluate the gradient of the LLF at a point $(\theta^{(i)}, \sigma_\epsilon^{2(i)})$ using Result 3.1. First compute the expectation $E\{\mathcal{X}_\tau\}$. Evaluate this by Monte Carlo via the fast recursive procedure in Section II. Using the recursive representation (15), we generate a large number (K) of

samples of a field with parameters $(\theta^{(i)}, \sigma_\epsilon^{2(i)})$. For each $\tau \in \eta_p$, we estimate $E\{\mathcal{X}_\tau\}$ from these samples as

$$\hat{E}\{\mathcal{X}_\tau\} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{X}_\tau^{(k)} \tag{37}$$

where $\mathcal{X}_\tau^{(k)}$ is the correlation of lag τ computed from field sample $\#k$. The algorithm is summarized below. Given the potential matrix $A_p(\theta^{(i)})$:

- *Step 1*: Compute the matrices $\{U_i, \Theta_i\}$ using the Riccati iteration (12) and (11).
- *Step 2*: Use the ("backward") state-space representation (19) to generate recursively K samples of the field, with a Gaussian random number generator providing the white noise input: For $k = 1$ to K

$$\begin{aligned} \boldsymbol{x}_N^{(k)} &= G_N^b \boldsymbol{w}_{(k)}^N, \\ \boldsymbol{x}_i^{(k)} &= F_i^b \boldsymbol{x}_{i+1}^{(k)} + G_i^b \boldsymbol{w}_{(k)}^i \qquad 1 \le i \le N-1 \end{aligned} \tag{38}$$

  with model matrices defined in (20). The sample correlation $\mathcal{X}_\tau^{(k)}$ is computed for sample $\#k$.
- *Step 3*: Substitute the sample correlations $\{\mathcal{X}_\tau^{(k)}\}_{k=1}^K$ in (37) and the results in (26) to obtain an estimate for the gradient at the given parameter space point $(\theta^{(i)}, \sigma_\epsilon^{2(i)})$.

For first-order fields, using the explicit equation for the log-likelihood function in (27), we obtain the following explicit equations for the gradient over the 2-D space (β_h, β_v), $\nabla L = [\nabla L_h, \nabla L_v]$ where ∇L_h, and ∇L_v, the partial derivatives of $L(\cdot)$ with respect to β_h and β_v, respectively, are given by

$$\nabla \mathcal{L}_{h_1} = \frac{1}{2NM}\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{\lambda_j(\mathcal{S}_M)}{(1-\beta_v\lambda_i(\mathcal{S}_N) - \beta_h\lambda_j(\mathcal{S}_M))} - \frac{\mathcal{X}_h}{\sigma_\epsilon^2} \tag{39}$$

$$\nabla \mathcal{L}_{v_1} = \frac{1}{2NM}\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{\lambda_i(\mathcal{S}_N)}{(1-\beta_v\lambda_i(\mathcal{S}_N) - \beta_h\lambda_j(\mathcal{S}_M))} - \frac{\mathcal{X}_v}{\sigma_\epsilon^2}. \tag{40}$$

Coder—Iterative Search: The iterative search step in the coder proceeds along the following lines.

1) *Initialization*: Choose search tolerance ϵ, the starting point $\theta^{(0)}$ (e.g., the origin), and the initial direction $D^{(0)}$ (e.g., the negative gradient as suggested in the Polak–Ribiere method)

$$\theta^{(0)} := 0, \quad \gamma_0 := 0, \quad i := 0, \quad D^{(0)} := -\nabla L(\theta^{(0)}) \tag{41}$$

   where $\nabla L(\cdot)$, the gradient of $L(\cdot)$, is computed by the subroutine $funcgradient(\cdot)$.
2) *Line minimizations*: Iterate through line minimizations until a stop condition is satisfied. The minimization along line i specified by the point and direction pair $(\theta^{(i)}, D^{(i)})$ proceeds as follows:
   a) The line minimum is bracketed by finding scalars a, b, c such that $a < b < c$ and

$$L(\theta^{(i)} + bD^{(i)}) < \min(L(\theta^{(i)} + aD^{(i)}), L(\theta^{(i)} + cD^{(i)})).$$

   The bracketing is done by subroutine $bracket(\cdot)$. The value of $L(\cdot)$ at any location in the parameter space is provided by subroutine $funcvalue(\cdot)$.
   b) Given bracket points a, b, c, subroutine $dbrent(\cdot)$ uses parabolic interpolation (Brent's method), to find the scalar $\alpha_{\min}$ that minimizes the value of the function along line i as in

$$\alpha_{\min} = \arg\min_{\alpha} L(\theta^{(i)} + \alpha D^{(i)}).$$

   c) The present value of the minimum point is updated as follows:

$$\theta^{(i+1)} := \theta^{(i)} + \alpha_{\min} D^{(i)}.$$

3) *Stop Condition*: The function value at $(\theta^{(i+1)})$ is compared with that at the previous estimate of the minimum $(\theta^{(i)})$. If the change is below tolerance ϵ, the parameter estimation is terminated; otherwise, the new conjugate direction is chosen according to Polak–Ribiere's update formula [8] as follows:

IF

$$\frac{2|L(\theta^{(i+1)}) - L(\theta^{(i)})|}{|L(\theta^{(i+1)})| + |L(\theta^{(i)})|} > \epsilon$$

then

$$\gamma_i = \frac{(\nabla L(\theta^{(i+1)}) - \nabla L(\theta^{(i)}))^T \nabla L(\theta^{(i+1)})}{\nabla L(\theta^{(i)})^T \nabla L(\theta^{(i)})}$$

$$D^{(i+1)} := -\nabla L(\theta^{(i+1)}) + \gamma_i D^{(i)}$$

$$i := i + 1$$

Go to beginning of line minimization

else

$$\hat{\theta}_{ML} := \theta^{(i)}$$

Exit the iteration.

APPENDIX B
SUBROUTINES

The coder and decoder use the following subroutines: $riccati(\cdot)$, $funcvalue(\cdot)$, $funcgradient(\cdot)$, $bracket(\cdot)$, $dbrent(\cdot)$, and $approxest(\cdot)$. Some subroutines use different procedures when the eigenvalues of matrix A_p are known (first- and special second-order models). For these models, explicit expressions are available for the evaluation of the likelihood function and its gradient, in, respectively, $funcvalue(\cdot)$ and $funcgradient(\cdot)$, and an exact description of the parameter space is available for use in the bracketing routine, $bracket(\cdot)$. We describe here the essential procedure for these models, as well as the one for arbitrary order.

Tolerances for the Algorithm: The user specifies three tolerances κ, ϵ, and δ. The first two are used only in the coder parameter estimation step, the third one in both the coder and the decoder. Tolerance κ is used in bracketing for first-order fields only, representing the minimum distance from the parameter space boundary that is acceptable from the point of view of numerical stability [see (30)]. Tolerance ϵ is the stopping condition for the iterative search in the parameter space. Finally, tolerance δ is the stopping criterion for convergence of the Riccati iteration in subroutine $riccati(\cdot)$.

1. riccati(·)

Inputs: θ, δ.

Outputs: $n_\delta, \{U_i\}_{i=1}^{n_\delta}, \{\Theta_i\}_{i=1}^{n_\delta}$ (optional, see comments below).

Procedure:

- Blocks B and C in the potential matrix A given in (5) are set up using the parameter vector θ. The structure of B and C is dependent on the model order and boundary conditions.
- The Riccati iteration in (12) is continued until convergence to specific tolerance δ. One possible metric for convergence using the easily computed matrix infinity norm $\|\cdot\|_\infty$ is

$$\|S_{i-1} - S_i\|_\infty \leq \delta \tag{42}$$

i.e., convergence is said to have occurred at iteration $i = n_\delta$, when the condition in (42) is satisfied. Due to the geometric convergence, this occurs in most cases for $n_\delta \ll N$. At each iteration i, S_i is tested for positivity. If nonpositive (indicating that the parameters are outside the valid parameter space of the model), the Boolean variable "valid" is set to zero, and the iteration is aborted. The Cholesky factor matrices $\{U_i, \Theta_i\}_{i=1}^{n_\delta}$ are computed using (11).

The number of iterations until convergence, n_δ, and the matrices $\{U_i\}_{i=1}^{n_\delta}$ are the output of this block. The set $\{\Theta_i\}_{i=1}^{n_\delta}$ may be returned as output or computed elsewhere, when required, from (11), depending on whether computational cost or memory usage is the primary constraint of the implementation.

2. **funcvalue**$(\cdot)$

Inputs: $\theta^{(i)}, D^{(i)}, \alpha, S_x, \{\mathcal{X}_\tau \in \eta_p\}, \{\lambda_k(\mathcal{S}_K)\}$.

Outputs: *valid* (Boolean), $L(\theta^{(i)} + \alpha D^{(i)})$.

Procedure for First-Order Fields: The (negative log) likelihood function is evaluated at the given location $(\theta^{(i)} + \alpha D^{(i)})$ using the expression given in (27), with σ^2 substituted from (24).

Procedure for Arbitrary-Order Fields: The function is evaluated at the given location $(\theta^{(i)} + \alpha D^{(i)})$ using (23), with σ^2 substituted from (24). The subroutine $riccati(\cdot)$ computes the blocks U_i up to convergence with tolerance δ. For convergence at n_δ (see procedure for $riccati(\cdot)$), the function value is

$$L(\theta, \sigma^2) = \frac{1}{2}\ln\sigma_\epsilon^2 - \frac{1}{NM}\left[\sum_i^{n_\delta} \ln|U_i| + (N - n_\delta)\ln|U_{n_\delta}|\right] + \frac{1}{2\sigma_\epsilon^2}\left[S_x - 2\sum_{\tau\in\eta_p}\beta_\tau X_\tau\right] \tag{43}$$

with σ^2 substituted from (24). If $riccati(\cdot)$ returns $valid = 0$, exit the procedure with $valid = 0$.

3. **funcgradient**$(\cdot)$

Inputs: $\theta^{(i)}, D^{(i)}, \alpha, S_x, \{\mathcal{X}_\tau \in \eta_p\}, \{\lambda_k(\mathcal{S}_K)\}$.

Outputs: $\nabla L(\theta^{(i)} + \alpha D^{(i)})$.

Procedure for First-Order Fields: The gradient is evaluated at the given location $(\theta^{(i)} + \alpha D^{(i)})$ using the expressions given in (39)–(40) with σ^2 substituted from (24).

Procedure for Arbitrary-Order Fields: The gradient is evaluated at the given location $(\theta^{(i)} + \alpha D^{(i)})$ using the expressions given in (25) and (26), with σ^2 substituted from (24). The statistical quantities in (26) $(E\{\mathcal{X}_\tau\})$ are estimated from samples of a GMRF with parameter vector $\theta = \theta^{(i)} + \alpha D^{(i)}$. These samples are synthesized using the recursive representation of the GMRF. The procedure is as follows.

1) Given θ, the subroutine $riccati(\cdot)$ computes the equivalent recursive representation matrices $\{U_i, \Theta_i\}$.
2) The ("backward") state-space representation given by (38) is used to generate recursively (one row at a time) K samples of the field. The sample correlation $\mathcal{X}_\tau^{(k)}$ is computed for sample $\#k$ by (35).
3) The sample correlations $\{\mathcal{X}_\tau^{(k)}\}_{k=1}^K$ are substituted into (37), and the results into (26) to obtain an estimate for the gradient at the given point.

4. **bracket**$(\cdot)$

Inputs: $\theta^{(i)}, D^{(i)}, \alpha, S_x, \{\mathcal{X}_\tau \in \eta_p\}, \{\lambda_k(\mathcal{S}_K)\}$, $funcvalue(\cdot)$.

Outputs: a, b, c

Procedure for First-Order Fields:

1) $a := 0$.
2) Set c equal to the smallest $k > 0$ such that $\theta^{(i)} + kD^{(i)}$ is in the parameter space. For example, for first-order models, zero bc, and $N = M$, the parameter space is given by (29). For numerical stability we use a boundary cushion (matrix A is undefined on the boundary) using (30) on the right-hand side of (29) and an equality instead of the inequality. For $k = k_1, k_2, k_3, k_4$, the point $\theta^{(i)} + kD^{(i)}$ is placed on one of the four boundary lines given by (29) [with (30) as the right-hand side], the negative values for $k_i, i = 1, 2, 3, 4$, are set to zero, and $c := \min(k_1, k_2, k_3, k_4)$.
3) **If**
   $L(\theta^{(i)} + aD^{(i)}) > L(\theta^{(i)} + cD^{(i)})$
   then
   Swap a and c
4) Repeat until exit condition is satisfied:
   - $b = a + c/2$
   - **If**
     $L(\theta^{(i)} + bD^{(i)}) < \min(L(\theta^{(i)} + aD^{(i)}), L(\theta^{(i)} + cD^{(i)}))$
     then
     Exit
     else
     $c := b$
     Go back to beginning of loop.

Procedure for Arbitrary-Order Fields:

1) $a := 0$.
2) Find middle point: starting with a large initial value for b, evaluate function at b using $funcvalue(\cdot)$, continue to decrease b (e.g., $b := b/2$) until a valid point is obtained.
3) Find outer point: starting at b step along line away from a and evaluate function. Repeat with decreasing step

sizes until a valid point c is found at which the function value is larger than at b.

5. dbrent$(\cdot)$

Inputs: $\theta^{(i)}, D^{(i)}, a, b, c, funcvalue(\cdot), funcgradient(\cdot)$
Outputs: $\alpha_{\min}$
Procedure: This subroutine uses Brent's method for parabolic interpolation to obtain the minimum along a line. The procedure begins with three points that bracket the line minimum. The routine is from [8].

6. approxest$(\cdot)$

Inputs: $\mathcal{X}_h, \mathcal{X}_v, \kappa$.
Outputs: $\theta = [\beta_h, \beta_v]$
Procedure: Compute approximate estimates β_h and β using (31).

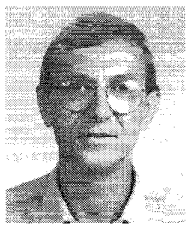
REFERENCES

[1] Anil K. Jain. *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[2] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commnun.*, vol. COM-28, no. 1, pp. 84–95, Jan. 1980.
[3] A. Asif and J. M. F. Moura, "Image codec with noncausal prediction, residual mean removal, and cascaded VQ," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 42–55, Feb. 1996.
[4] D. J. Vaisey and A. Gersho, "Variable block-size image coding," in *Int. Conf. Acoust., Speech, Signal Processing*, pp. 1051–1054, 1987.
[5] J. M. F. Moura and Nikhil Balram, "Recursive structure of noncausal Gauss–Markov random fields," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 334–354, Mar. 1992.
[6] J. W. Woods, "Two-dimensional discrete Markovian fields," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 232–240, 1972.
[7] N. Balram and J. M. F. Moura. Noncausal Gauss–Markov random fields: parameter structure and estimation," *IEEE Trans Inform. Theory*, vol. 39, no. 4, pp. 1333–1355, July 1993.
[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge, MA: Cambridge Univ. Press, 1989.
[9] A. C. Hung. (1993, June). *PVRG-JPEG codec 1.1* (3rd ed.), Port. Video Research Grp., Stanford Univ.
[10] USC–SIPI image data base. Signal Image Processing Inst., Univ. South. Calif., Los Angeles, CA.
[11] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.

Nikhil Balram was born in Bombay, India, in 1963. He received the B.S. degree with University Honors in 1986, the M.S. degree in 1988, and the Ph.D. degree in January 1992, all in electrical engineering, from Carnegie Mellon University, Pittsburgh, PA.

He was with the Graphics/Video VLSI department at IBM, Boca Raton, FL, from March 1992 until June 1994, where he worked on the XGA multimedia subsystem, memory controller design, hardware graphics algorithms, and a new highly parallel graphics coprocessor architecture. He was also a member of the Workshop 2000 task force and the Video/Graphics product strategy team. From June 1994 until January 1995, he worked for the multimedia division of Rexon/Tecmar as an architect of a new distance learning system. Since January 1995, he has been with Kaiser Electronics, San Jose, CA, where he has been working on graphics hardware architectures, new algorithms for video and graphics, modeling and characterization of flat panel displays, objective image quality metrics, and video architecture for helmet-mounted displays. Dr. Balram's current research interests include digital signal/image/video processing, image compression, video coding, computer graphics algorithms, video and graphics hardware, flat panel displays, helmet-mounted displays, image quality metrics, and visual perception. He has authored or coauthored more than 20 technical publications, including an invited chapter in the *Handbook of Statistics 10: Signal Processing and its Applications*, N. K. Bose and C. R. Rao, Eds. He has invented new algorithms in image compression, computer graphics, and video processing that are in the process of being patented.

Dr. Balram received the Patterson Prize in 1985 and the E. M. Williams award in 1986. He is a member of Tau Beta Pi, ACM, and TiE.



José M. F. Moura (S'71–M'75–SM'90–F'94) received the engenheiro electrotécnico degree in 1969 from Instituto Superior Tecnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and the D.Sc. in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1973 and 1975, respectively.

He is presently a Professor of Electrical and Computer Engineering at Carnegie Mellon University, Pittsburgh, which he joined in 1986. Prior to this, he was on the faculty of IST where he was an Assistant Professor (1975), Professor Agregado (1978), and Professor Catedratico (1979). He has had visiting appointments at several institutions, including MIT (Genrad Associate Professor of Electrical Engineering and Computer Science, 1984–1986, also associated with LIDS) and the University of Southern California, Los Angeles (research scholar, Department of Aerospace Engineering, summers 1978–1981). His research interests lie in statistical signal processing (one and two dimensional), image and video processing, array processing, underwater accoustics, and multiresolution techniques. He has had over 160 technical contributions, including invited ones, published in international journals and conference proceedings, and is coeditor of *Nonlinear Stochastic Problems* (Reidal, 1983) and *Accoustic Signal Processing for Ocean Exploration* (Kluwer, 1993).

Dr. Moura is currently the Editor-in-Chief for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He was a member of the IEEE Press Board from 1991–1995 and a technical Associate Editor for the IEEE SIGNAL PROCESSING LETTERS from 1993–1995. He is a member of the Underwater Accoustics Technical Committee of the Signal Processing Society. He was an Associate Editor for the SIGNAL PROCESSING TRANSACTIONS (Sonar and Radar) from 1988–1992, a member of the technical committee of the IEEE International Symposium on Information Theory (ISIT'1993), and a member of the technical committee of the IEEE International Conference on Image Processing (ICIP'95). He was elected Fellow of the IEEE in November, 1993, and corresponding member of the Academy of Sciences of Portugal (Section of Sciences) in July 1992. He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM. He has organized and codirected two international scientific meetings on signal processing theory and applications.