Multiresolution Vector Quantization

Michelle Effros, *Senior Member, IEEE,* and Diego Dugatkin, *Member, IEEE*

***Abstract*—Multiresolution source codes are data compression algorithms yielding embedded source descriptions. The decoder of a multiresolution code can build a source reproduction by decoding the embedded bit stream in part or in whole. All decoding procedures start at the beginning of the binary source description and decode some fraction of that string. Decoding a small portion of the binary string gives a low-resolution reproduction; decoding more yields a higher resolution reproduction; and so on. Multiresolution vector quantizers are block multiresolution source codes. This paper introduces algorithms for designing fixed- and variable-rate multiresolution vector quantizers. Experiments on synthetic data demonstrate performance close to the theoretical performance limit. Experiments on natural images demonstrate performance improvements of up to 8 dB over tree-structured vector quantizers. Some of the lessons learned through multiresolution vector quantizer design lend insight into the design of more sophisticated multiresolution codes.**

***Index Terms*—Embedded source code design, fixed rate, multiuser, network, progressive transmission, successive refinement, variable rate.**

I. Introduction

We call a source code designed to be used at a single rate and reproduction fidelity (or resolution) a *single-resolution* source code. Vector quantizers are block single-resolution source codes, and distortion-rate theory is single-resolution source coding theory. Given a stationary source μ, the distortion-rate function $D_\mu(R)$ is an information-theoretic lower bound on the expected distortion achievable using *any* compression scheme with expected rate less than or equal to R. The nth-order operational distortion-rate function $\delta_\mu^{(n)}(R)$ is the expected distortion of the best dimension-n vector quantizer with the same expected rate. (All rates and distortions are measured per sample.) Since $\delta_\mu^{(n)}(R)$ converges to $D_\mu(R)$ as n grows without bound [1]–[4],[1] vector quantizers are asymptotically optimal single-resolution source codes.

Manuscript received October 20, 2000; revised November 22, 2002. The material in this paper was presented in part at the 1998 Data Compression Conference and the 1998 Asilomar Conference on Signals and Systems. This material is based on work supported in part by the National Science Foundation CAREER Award MIP-9501977, a grant from the Charles Lee Powell Foundation, donations through the Intel 2000 Program, and the Oringer Fellowship.

M. Effros is with the Department of Electrical Engineering, MC 136-93, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: effros@caltech.edu).

D. Dugatkin was with the Department of Electrical Engineering at the California Institute of Technology, Pasadena, CA 91125 USA. He is now with Ixia Corporation, Calabasas, CA 91302 USA (e-mail: diego@caltech.edu).

Communicated by P. A. Chou, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2004.838381

[1]If the stationary source is ergodic, then there exist both fixed- and variable-rate vector quantizers satisfying this property. If the stationary source is nonergodic, then variable-rate vector quantization is required to achieve performance arbitrarily close to $D_\mu(R)$.

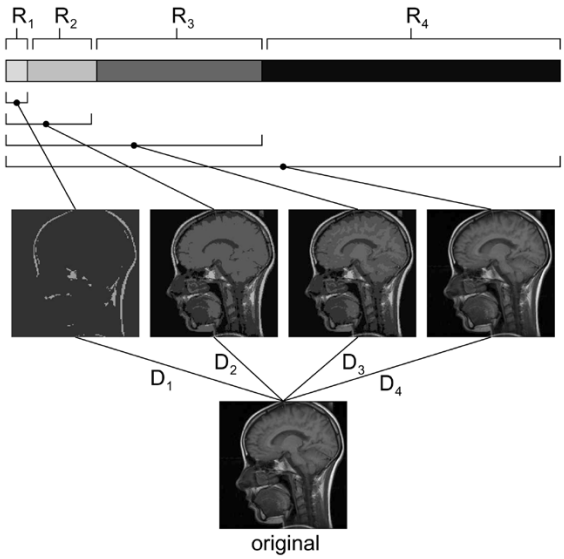


Fig. 1. A four-resolution image description. Decoding the first R_1 bits per symbol (bps) of the binary description yields a reproduction with distortion D_1. Decoding an additional R_2 bps (for a total description length of $R_1 + R_2$ bps) yields a reproduction of distortion $D_2 \leq D_1$. and so on.

Given a fixed coding dimension n and rate constraint R, an optimal vector quantizer is a vector quantizer that achieves distortion $\delta_\mu^{(n)}(R)$, the lowest expected distortion per sample over all n-dimensional vector quantizers that require an expected rate of no more than R bits per symbol (bps). Equivalently, given distortion constraint D, an optimal vector quantizer minimizes the expected rate over n-dimensional vector quantizers satisfying the rate constraint; or, given fixed Lagrangian parameter $\lambda > 0$, an optimal vector quantizer minimizes the Lagrangian performance $D+\lambda R$ over n-dimensional vector quantizers. These optimality criteria form the basis of single-resolution vector quantizer design [5], [6].

A multiresolution source code (also known as a progressive transmission, embedded, or successive refinement code) is a single compression system that describes data at a variety of rates and resolutions. A multiresolution source code creates a binary source description such that low-resolution descriptions of the given data set are embedded in higher-resolution descriptions of the same data set. Fig. 1 demonstrates the action of a four-resolution source code. Decoding only the first R_1 bps yields a low-resolution reproduction of the image with per sample distortion equal to D_1; decoding an additional R_2 bps (for a total description length of $R_1 + R_2$ bps) yields a higher resolution source reproduction with per sample distortion $D_2 \leq D_1$, and so on.

Multiresolution source coding is useful for applications where a single source must be described to a variety of different users or using an available rate that varies from system use

to system use. For example, a single file on a particular web site may be examined by thousands of web site visitors. Some may want to see the data at high reproduction fidelity, others may favor fast transmission speed over reproduction quality, and still others may wish to determine acceptable reproduction quality during the data transfer process. Since single-resolution source codes fix a single rate and reproduction quality, no such code can satisfy all of these needs simultaneously. In contrast, a multiresolution source code yields a single source description from which each user can decode to the rate or resolution most useful to him. Multiresolution codes are also useful in mobile communication systems, where the available communication rate may vary as a function of network traffic and physical location.

Multiresolution vector quantization is block multiresolution source coding. In this case, the encoder blocks an incoming data stream into contiguous blocks of dimension n and chooses for each n-block a collection of n-dimensional reproductions. In particular, in an L-resolution code, the encoder chooses L reproductions for each data block. The encoder describes the chosen collection of reproductions using a fixed- or variable-rate binary string such that the first portion of the binary string describes the resolution-1 reproduction, the second portion of the binary string describes the resolution-2 reproduction given the resolution-1 reproduction already described, and so on. The decoder decodes the desired portion of the binary string, updating its source reproduction as the binary descriptions for higher and higher resolution reconstructions become available.

For some sources, the flexibility afforded by multiresolution source coding comes at a price. In particular, there exist sources for which it is not possible to achieve distortions

$$D_\ell = \delta_\mu^{(n)}\left(\sum_{i=1}^{\ell} R_i\right), \qquad \text{for all } \ell \in \{1,\ldots,L\}$$

simultaneously [7]–[9]. While the rate loss for multiresolution coding is generally small [10]–[13], its existence makes the definition of optimality difficult. It is not clear, for example at rates (R_1, R_2), whether achieving

$$D_1 > \delta_\mu^{(n)}(R_1) \quad \text{and} \quad D_2 = \delta_\mu^{(n)}(R_1 + R_2)$$

is better or worse than achieving

$$D_1 = \delta_\mu^{(n)}(R_1) \quad \text{and} \quad D_2 > \delta_\mu^{(n)}(R_1 + R_2)$$

or whether achieving both

$$D_1 > \delta_\mu^{(n)}(R_1) \quad \text{and} \quad D_2 > \delta_\mu^{(n)}(R_1 + R_2)$$

can ever be optimal.

Multiresolution distortion-rate theory [8], [14]–[18] addresses the question of multiresolution source code optimality in the limit of high coding dimension n. Given a stationary source μ, let $\mathcal{R}_n^{\mathrm{fr},L}(\mu)$ and $\mathcal{R}_n^{\mathrm{vr},L}(\mu)$ be the closure of the set of rate-distortion vectors achievable on source μ by fixed- and variable-rate dimension-n multiresolution vector quantizers, respectively. Let $\mathcal{R}^L(\mu)$ denote the corresponding information theoretic bound on the L-resolution rate-distortion region [18]. Since $\mathcal{R}_n^{\mathrm{fr},L}(\mu)$ and $\mathcal{R}_n^{\mathrm{vr},L}(\mu)$ converge to $\mathcal{R}^L(\mu)$, multiresolution vector quantizers are asymptotically optimal multiresolution source codes [18].[2]

Given a fixed coding dimension n, an optimal fixed- or variable-rate multiresolution vector quantizer is an n-dimensional code that achieves rate-distortion performance on the lower boundary of $\mathcal{R}_n^{\mathrm{fr},L}(\mu)$ or $\mathcal{R}_n^{\mathrm{vr},L}(\mu)$. While $\mathcal{R}_n^{\mathrm{fr},L}$ and $\mathcal{R}_n^{\mathrm{vr},L}$ are not convex, Lagrangian methods may be used to find the lower convex hull of $\mathcal{R}_n^{\mathrm{fr},L}$ and $\mathcal{R}_n^{\mathrm{vr},L}$, given by

$$j_n^{(\mathrm{fr})}(a^L, b^L, \mu) = \min_{(R^L,D^L)\in\mathcal{R}_n^{\mathrm{fr},L}(\mu)} \sum_{\ell=1}^{L}[a_\ell D_\ell + b_\ell R_\ell]$$

$$j_n^{(\mathrm{vr})}(a^L, b^L, \mu) = \min_{(R^L,D^L)\in\mathcal{R}_n^{\mathrm{vr},L}(\mu)} \sum_{\ell=1}^{L}[a_\ell D_\ell + b_\ell R_\ell].$$

Here, (a^L, b^L) describes the direction of a hyper-plane supporting the convex hull of the L-resolution operational rate-distortion region at a single point [18]. We use minimization of

$$\sum_{\ell=1}^{L}[a_\ell D_\ell + b_\ell R_\ell] \tag{1}$$

as our optimality criterion for multiresolution vector quantizer design. Using this approach, a dimension-n fixed- or variable-rate vector quantizer is optimal if it lies on the lower convex hull of $\mathcal{R}_n^{\mathrm{fr},L}$ or $\mathcal{R}_n^{\mathrm{vr},L}$, respectively. This optimality criterion, first introduced in [19], can also be applied in other multiresolution codes [20], [21]. It differs critically from its predecessors, as described in Section III.

Rate-distortion and quantization theory lend valuable insight into the properties of optimal single- and multiresolution source codes. Some important outcomes in the single-resolution case are (locally) optimal design algorithms for fixed- and variable-rate vector quantizers [5], [6]. We here pursue similar iterative descent algorithms for fixed- and variable-rate multiresolution vector quantizer design.

The remainder of this paper is organized as follows. Section II establishes notation. Section III contains a brief look at previous vector quantizers for multiresolution source coding, comparing their (implicit) optimality criteria to the criterion given in (1). The optimal multiresolution vector quantizer design algorithm follows in Section IV. Sections V and VI treat code complexity and parameter choice, respectively. Section VII contains experimental results. A summary appears in Section VIII.

II. Preliminaries

A vector quantizer used for multiresolution coding comprises an encoder and a decoder. We describe the decoder as a tree in

[2]If the stationary source is ergodic, then

$$\lim_{n\to\infty} \mathcal{R}_n^{\mathrm{fr},L}(\mu) = \lim_{n\to\infty} \mathcal{R}_n^{\mathrm{vr},L}(\mu) = \mathcal{R}^L(\mu)$$

and the information-theoretic rate-distortion region characterizes the set of rates and distortions achievable through either fixed- or variable-rate L-resolution source coding [18, Theorems 1–4]. If the stationary source is nonergodic and the alphabet is Polish, then [18, Theorems 5–12]

$$\lim_{n\to\infty} \mathcal{R}_n^{\mathrm{fr},L}(\mu) \subseteq \lim_{n\to\infty} \mathcal{R}_n^{\mathrm{vr},L}(\mu) = \mathcal{R}^L(\mu).$$

which each node contains a single reproduction value or codeword. The encoder chooses for each data vector a path from the root to a leaf of the tree. The encoder's source description is a binary description of that path. That path description may be decoded in part or in whole. The resulting source reproduction is the codeword associated with the described node in the tree-structured codebook. A more precise description follows.

Consider an L-resolution source coding system. The encoder $\boldsymbol{\alpha}: \mathcal{X}^n \to \mathcal{S}$ maps a vector of source symbols to a binary string. The decoder $\boldsymbol{\beta}: \mathcal{S} \to \boldsymbol{\mathcal{Y}}^n$ maps a binary string to a collection of reproductions

$$\left(Y_{(1)}^n, \ldots, Y_{(L)}^n\right) \in \boldsymbol{\mathcal{Y}}^n = \mathcal{Y}_{(1)}^n \times \cdots \times \mathcal{Y}_{(L)}^n.$$

We break the encoding and decoding operations into quantization and lossless coding components, giving $\boldsymbol{\alpha} = \gamma \circ \alpha$ and $\boldsymbol{\beta} = \beta \circ \gamma^{-1}$, where α and β are the quantizer encoder and decoder, γ and γ^{-1} are the encoder and decoder for the embedded lossless code, and $\circ$ denotes composition. Before describing these components, we establish the notation used to describe the tree-structured codebook.

The tree-structured codebook for an L-resolution code is a depth-L tree. Each tree node has a unique label. For each $\ell \in \{0, \ldots, L\}$, we use $\mathcal{K}_\ell$ to denote all nodes at depth ℓ. For any $k_\ell \in \mathcal{K}_\ell$, we use $\mathcal{K}_{\ell+1}(k_\ell)$ to denote the subset of $\mathcal{K}_{\ell+1}$ that descends from node k_ℓ. Using these definitions, $\mathcal{K}_{\ell+1}(k_\ell) \cap \mathcal{K}_{\ell+1}(k'_\ell) = \phi$ for any distinct nodes $k_\ell, k'_\ell \in \mathcal{K}_\ell$, and $\mathcal{K}_{\ell+1} = \cup_{k_\ell \in \mathcal{K}_\ell} \mathcal{K}_{\ell+1}(k_\ell)$. The set

$$\mathcal{K} = \{(k_1, \ldots, k_L) \in \mathcal{K}_1 \times \cdots \times \mathcal{K}_L : k_\ell \in \mathcal{K}_\ell(k_{\ell-1}) \quad \forall\, \ell \in \{1, \ldots, L\}\}$$

describes all paths from the root to a leaf of the tree-structured codebook. (The root k_0 is unique and therefore is not specified in each path description $\boldsymbol{k} \in \mathcal{K}$.)

The quantizer encoder $\alpha: \mathcal{X}^n \to \mathcal{K}$, which is many-to-one and information lossy in general, maps each source vector $X^n \in \mathcal{X}^n$ to a distinct path $\boldsymbol{k} \in \mathcal{K}$ through the tree-structured codebook. We here use $\alpha_\ell(X^n) = k_\ell$ to specify the ℓth -step in the L-step path $\alpha(X^n)$. Note, however, that the full path $\alpha(X^n) = (\alpha_1(X^n), \ldots, \alpha_L(X^n))$ is chosen jointly rather than sequentially.

The embedded lossless encoder $\gamma: \mathcal{K} \to \mathcal{S}$, which is one-to-one and information lossless, maps each path index $\boldsymbol{k} \in \mathcal{K}$ to a channel codeword $\boldsymbol{s} \in \mathcal{S}$. Each channel codeword $\boldsymbol{s} \in \mathcal{S}$ gives a step-by-step description of the path $\boldsymbol{k}$ as described next. (For simplicity, we here assume that channel codewords are binary strings and use $(s_1, \ldots, s_\ell)$ to denote the concatenation of binary strings $s_1, \ldots, s_\ell$.) For each $\ell \in \{1, \ldots, L\}$ and each $k_{\ell-1} \in \mathcal{K}_{\ell-1}$, let $S_\ell(k_{\ell-1}) \subseteq \{0,1\}^*$ denote a prefix-free code of size $|\mathcal{K}_\ell(k_{\ell-1})|$. Define

$$\mathcal{S} = \{(s_1, \ldots, s_\ell) : \exists \boldsymbol{k} \in \mathcal{K} \text{ s.t. } s_\ell \in \mathcal{S}_\ell(k_{\ell-1})\ \forall\, \ell \in \{1, \ldots, L\}\}.$$

If $\gamma(\boldsymbol{k}) = (s_1, \ldots, s_L)$, then for each $\ell \in \{1, \ldots, L\}$, $s^\ell = (s_1, \ldots, s_\ell)$ gives a description of the first ℓ steps $k^\ell = (k_1, \ldots, k_\ell)$ in the path $\boldsymbol{k} = (k_1, \ldots, k_L)$. Since $S_\ell(k_{\ell-1})$ and $S_\ell(k'_{\ell-1})$ satisfy the prefix property separately but need not satisfy it together, channel codeword s_ℓ may not be uniquely decodable on its own. However, channel codeword s_ℓ is instantaneously decodable given $(s_1, \ldots, s_{\ell-1})$. If $\gamma(\boldsymbol{k}) = (s_1, \ldots, s_L)$, we use $\gamma_\ell(k^\ell) = s_\ell$ to denote the ℓth increment of the channel codeword. Note that for each $\ell \in \{1, \ldots, L\}$, the ℓth increment of the channel codeword depends only on $k_1, \ldots, k_\ell$ (and not on $k_{\ell+1}, \ldots, k_L$).

The encoder sends its binary description of the data to the decoder. Given a sequence of data vectors

$$X_{(1)}^n, X_{(2)}^n, \ldots, X_{(M)}^n$$

with binary descriptions

$$\boldsymbol{\alpha}\left(X_{(m)}^n\right) = \left(s_1^{(m)}, \ldots, s_L^{(m)}\right)$$

the path descriptions are generally ordered as

$$s_1^{(1)}, s_1^{(2)}, \ldots, s_1^{(M)}, s_2^{(1)}, \ldots, s_2^{(M)}, \ldots, s_L^{(1)}, \ldots, s_L^{(M)}$$

so that the decoder can build a first-resolution reproduction of the entire data sequence and then update that reproduction to get reproductions of higher and higher quality.

The lossless decoder $\gamma^{-1}: \mathcal{S} \to \mathcal{K}$ reverses the operation of the embedded lossless encoder γ. Like γ, γ^{-1} can be used sequentially, with $\gamma_\ell^{-1}(s^\ell) = k_\ell$ for each $\ell \in \{1, \ldots, L\}$.

The quantizer decoder $\beta: \mathcal{K} \to (\mathcal{Y}_{(1)}^n \times \cdots \times \mathcal{Y}_{(L)}^n)$ maps the path description $\boldsymbol{k}$ to the L reproduction vectors that lie along path $\boldsymbol{k}$ from the root to a leaf in the tree-structured codebook. More precisely, $\beta = (\beta_1, \ldots, \beta_L)$, where $\beta_\ell(k^\ell) \in \mathcal{Y}_{(\ell)}^n$, and $\beta(\boldsymbol{k}) = (\beta_1(k^1), \ldots, \beta_L(k^L))$. For any $\ell \in \{1, \ldots, L\}$, the reconstruction of the ℓth -resolution reproduction $Y_{(\ell)}^n \in \mathcal{Y}_{(\ell)}^n$ relies only on the first ℓ steps $k^\ell = (k_1, \ldots, k_\ell)$ of $\boldsymbol{k}$. The L reproduction alphabets $\mathcal{Y}_{(1)}, \ldots, \mathcal{Y}_{(L)}$ are usually, but not necessarily, equal both to each other and to the source alphabet $\mathcal{X}$.[3]

Together, the encoder $\boldsymbol{\alpha}$ and decoder $\boldsymbol{\beta}$ allow the data to be encoded and decoded at any of L resolutions. In particular, $\alpha^\ell(X^n) = (\alpha_1(X^n), \ldots, \alpha_\ell(X^n))$ gives the source's binary description at resolution ℓ, and $\beta_\ell(\alpha^\ell(X^n))$ gives the highest resolution reproduction available after seeing the first ℓ components of the source description.

Let $Q^{L,n} = (\alpha, \beta, \gamma)$ denote an L-resolution block source code of dimension n with quantizer encoder α, quantizer decoder β, and lossless code γ. If γ uses a fixed-rate channel codebook, then $Q^{L,n}$ is called a fixed-rate block L-resolution source code. If γ uses a variable-rate channel codebook, then $Q^{L,n}$ is called a variable-rate block L-resolution source code. For any $L \geq 1$, let $\mathcal{Q}^{\text{fr}}(L,n)$ and $\mathcal{Q}^{\text{vr}}(L,n)$ be the classes of fixed- and variable-rate L-resolution dimension-n block source codes respectively. Notice that $\mathcal{Q}^{\text{fr}}(L,n) \subseteq \mathcal{Q}^{\text{vr}}(L,n)$.

Given distortion measure $\rho^{(\ell)}: \mathcal{X} \times \mathcal{Y}_{(\ell)} \to [0, \infty)$, the expected distortion of $Q^{L,n}$ is

$$D_\ell = \frac{1}{n} E_\mu \rho_n^{(\ell)}\left(X^n, \beta_\ell(\alpha^\ell(X^n))\right)$$

[3]The source and reproduction alphabets need not be finite or even countable. We later assume alphabets for which the ergodic decomposition holds. (This condition covers almost any situation encountered in practice; see, for example, [22].)

in resolution ℓ, where

$$\rho_n^{(\ell)}(x^n, y^n) = \sum_{i=1}^{n} \rho^{(\ell)}(x_i, y_i).$$

The expected (incremental) rate is

$$R_\ell = \frac{1}{n} E_\mu |\gamma_\ell(\alpha^\ell(X^n))|$$

where $|s|$ denotes the length of s for any $s \in \mathcal{S}_\ell$.

We wish to find multiresolution codes from $\mathcal{Q}^{\rm fr}(L,n)$ and $\mathcal{Q}^{\rm vr}(L,n)$ that achieve performance (R^L, D^L) at extremal points on the convex hull of $\mathcal{R}_n^{{\rm fr},L}$ and $\mathcal{R}_n^{{\rm vr},L}$.[4] We use an iterative descent technique analogous to the generalized Lloyd algorithm [5] and the entropy constrained vector quantization (ECVQ) design algorithm [6] to explicitly minimize

$$j(Q^{L,n}) = \sum_{\ell=1}^{L} \frac{1}{n} E_\mu \Big[a_\ell \rho_n^{(\ell)} \left(X^n, \beta_\ell(\alpha^\ell(X^n)) \right) + b_\ell |\gamma_\ell(\alpha^\ell(X^n))| \Big]. \quad (2)$$

The resulting technique accomplishes (locally) optimal fixed- and variable-rate code design. Before describing our algorithm, we compare (2) with the optimality criteria implicit in prior algorithms.

III. Optimality Criteria for Code Design

By examining the design strategies and encoder definitions for multiresolution codes, we can learn something about the definitions of optimality that they (implicitly) employ. Codes that can be used for multiresolution source description (whether or not that was the initial intention of their design) include tree-structured scalar and vector quantizers [23]–[25], multistage or residual vector quantizers [26], [27], multiresolution transform codes (see, for example, [28], and the references therein), multiresolution trellis source codes [29], and multiresolution source codes combining wavelets or other frequency decompositions with zero-trees or other embedded codes [30]–[32]. We here focus on only the most closely related vector quantizers.

The following four categories give examples of strategies found in prior multiresolution codes using tree-structured vector quantizer codebooks. In the first category, exemplified by tree-structured vector quantization (TSVQ) [23], [25], the encoder chooses a path through the tree-structured codebook in a top-down greedy fashion. That is, for each source vector, the encoder starts at the root of the tree and chooses the best first-resolution reproduction, then chooses the best second-resolution reproduction of all second-resolution reproductions descending from the chosen first-resolution reproduction, and so on. Since each step in the path choice affects future resolutions but is made without regard to that impact, each low-resolution reproduction is effectively considered to be infinitely more important than all higher resolution reproductions that build from it.

[4]We can achieve arbitrary points on these convex hulls using time sharing, but time sharing is typically unnecessary in practice due to the richness of the space of available points.

In the second category, exemplified by the progressive vector quantizer (progressive VQ) of [33], the encoder considers only the highest level reproduction in choosing its path through the tree-structured codebook. That is, each data vector is mapped to the path ending in the leaf that gives the best reproduction. In this case, the highest resolution description is considered to be infinitely more important than any other description. Codes of this type in some sense reverse the priorities of TSVQ.

A third category of tree-structured codes, exemplified by pruned TSVQ (PTSVQ) [24], implicitly incorporates priorities somewhere between those of TSVQ and those of progressive VQ. PTSVQ combines the top-down greedy tree-design algorithm and encoder of TSVQ with a pruning algorithm. The pruning algorithm defines the resolutions of the code so that the rate-distortion performance at each resolution sits on the convex hull of all rate-distortion points achievable using subtrees of some initial tree-structured codebook. Since PTSVQ combines greedy codebook design and encoding with a globally optimal pruning procedure, PTSVQ's priority function tempers the low-rate priorities of TSVQ to some degree.

The final category of tree-structured codes, here called "weighted" codes, defines optimality in terms of the minimization of a weighted sum of the performances at the different resolutions. Typically, the weighting represents an expectation over the resolutions. For example, optimality in [34]–[36] corresponds to minimization of an expected distortion $\sum_\ell p_\ell D_\ell$ with respect to distribution $\{p_\ell\}$ over the resolutions in a fixed-rate code. Similarly, [36], [37] designs scalar quantizers that minimize the expected distortion $\sum_\ell p_\ell D_\ell$ with respect to a constraint on the expected rate $\sum_\ell p_\ell R_\ell$, giving optimization criterion $\sum_\ell p_\ell D_\ell + \lambda \sum_\ell p_\ell R_\ell$.

Given appropriate choices of the parameters (a^L, b^L) in the Lagrangian performance measure $j(Q^{L,n})$ described in (2), $j(Q^{L,n})$ can mimic the optimality criteria used in fixed-rate VQ, TSVQ, progressive VQ, and weighted design and in the entropy-constrained variations of these algorithms. For example, let $\mathcal{Q}^{\rm fr}(L,n,R^L) \subset \mathcal{Q}^{\rm fr}(L,n)$ be the class of fixed-rate n-block codes $Q^{L,n} = (\alpha, \beta, \gamma)$ that satisfy

$$(1/n)|\gamma_\ell(\alpha^\ell(x^n))| = R_\ell$$

for all $\ell \in \{1, \ldots, L\}$ and all $x^n \in \mathcal{X}^n$. Then

$$\arg \min_{Q^{L,n} \in \mathcal{Q}^{\rm fr}(L,n,R^L)} j(Q^{L,n}) = \arg \min_{Q^{L,n} \in \mathcal{Q}^{\rm fr}(L,n,R^L)} \sum_{\ell=1}^{L} E_\mu \left[a_\ell \rho_n^{(\ell)} \left(X^n, \beta_\ell(\alpha^\ell(X^n)) \right) \right]$$

and for any fixed incremental rate vector R^L, only the value of a^L affects the code choice. For any $\ell \in \{1, \ldots, L\}$ setting $a_\ell = 1$ and $a_j = 0$ for all $j \neq \ell$ gives an optimization equivalent to the generalized Lloyd algorithm. Setting $a_\ell \gg a_{\ell+1}$ for all $\ell \in \{1, \ldots, L-1\}$ and $a_L > 0$ gives an optimization equivalent to the TSVQ optimization; setting $a_\ell \ll a_{\ell+1}$ for all $\ell \in \{1, \ldots, L-1\}$ and $a_1 > 0$ gives an optimization equivalent to the progressive VQ optimization; and setting $a_\ell = p_\ell$ for some probability distribution over the L resolutions gives an optimization of the criterion $\sum_{\ell=1}^{L} a_\ell D_\ell$ used in weighted codes.

For variable-rate n-block codes $Q^{L,n} = (\alpha, \beta, \gamma) \in \mathcal{Q}^{\mathrm{vr}}(L, n)$, if $a_\ell = 1$ for some fixed $\ell \in \{1, \ldots, L\}$ and zero otherwise and $b_\ell = \lambda$ for all $j \le \ell$ and zero otherwise, then the above optimization criterion is equivalent to the single-resolution Lagrangian optimization $D_\ell + \lambda \left(\sum_{i=1}^{\ell} R_i\right)$.[5] Thus, our design algorithm reduces to the ECVQ design algorithm under these conditions. Setting $a_\ell \gg a_{\ell+1}$ for all $\ell \in \{1, \ldots, L-1\}$, $a_L > 0$, and $b_\ell = a_\ell \sum_{i=\ell}^{L} \lambda_i$ for all $\ell \in \{1, \ldots, L\}$ is equivalent to an entropy-constrained TSVQ design with slope λ_ℓ at resolution ℓ. Using $a_\ell \ll a_{\ell+1}$ for all $\ell \in \{1, \ldots, L-1\}$, $a_1 > 0$, and $b_\ell = a_\ell \sum_{i=\ell}^{L} \lambda_i$ for all $\ell \in \{1, \ldots, L\}$ is equivalent to a progressive ECVQ design with slope λ_ℓ at resolution ℓ. The optimization used in weighted codes is achieved by setting $a_\ell = p_\ell$ and $b_\ell = p_\ell \sum_{i=\ell}^{L} \lambda = p_\ell (L - \ell + 1)\lambda$ for all $\ell \in \{1, \ldots, L\}$.

While (2) can be used to mimic the optimality criteria used in fixed-rate VQ, TSVQ, progressive VQ, and weighted code design and in their entropy-constrained variations, the proposed optimality criterion is not equivalent to any of the earlier alternatives since there exist extremal points on the lower convex hull of the operational rate-distortion region that cannot be achieved through any of the prior methods. The optimality criterion used in weighted codes comes closest. In [34], [35], the weighted criterion $\sum_{\ell=1}^{L} p_\ell D_\ell$ is used for scalar quantizer design. This same criterion, when extended to the vector $(n > 1)$ case, allows the design of codes achieving any point on the convex hull of $\mathcal{R}_n^{\mathrm{fr},L}(\mu)$. Unfortunately, the weighted criterion fails for the variable-rate case since there exist points on the convex hull of $\mathcal{R}_n^{\mathrm{vr},L}(\mu)$ that cannot be achieved by optimization with respect to either the above weighted distortion measure or its entropy constrained variation

$$\sum_{\ell=1}^{L} p_\ell D_\ell + \lambda \sum_{\ell=1}^{L} p_\ell R_\ell = \sum_{\ell=1}^{L} (p_\ell D_\ell + \lambda p_\ell R_\ell)$$

[36], [37]. By forcing $b_\ell / a_\ell = \lambda$ across all resolutions, the weighted criterion removes $L - 1$ of the $2L - 1$ degrees of freedom from the optimization procedure in (2) and severely restricts the class of supporting hyperplanes, as shown experimentally in Section VII.

IV. The MRVQ Design Algorithm

We here consider the joint optimization of a quantizer encoder α, quantizer decoder β, and lossless code γ. We call the jointly optimized code a multiresolution vector quantizer (MRVQ).[6] Given $a^L, b^L \ge 0$, our design objective is to minimize (2) over all possible fixed- or variable-rate multiresolution source codes. Tracing out the entire convex hull requires a separate design for a variety of (a^L, b^L) values. We consider the choice of (a^L, b^L) to match target rate or distortion constraints in Section VI.

The design algorithm is an iterative descent technique. We initialize the design with an L-resolution tree-structured codebook. For each $\ell \in \{1, \ldots, L\}$, depth ℓ in the tree contains all resolution-ℓ reproductions. Those reproductions are set arbitrarily. For any $\ell < L$, the number of branches descending from each node at depth ℓ is an integer greater than or equal to one. For a fixed-rate code, the tree's branching rate is uniquely determined by the desired rate at each resolution. In the variable-rate case, branching rate initialization trades off a desire to exceed the optimal number of branches (as in ECVQ, the iterative design removes excess branches but cannot increase the number of branches) and the need to keep computational complexity in check. As a rule of thumb, a branching rate of $2^{c(nR_\ell)}$ in resolution ℓ seems to achieve good experimental performance; here c is a constant between 2 and 3. In both fixed- and variable-rate code design, we initialize the lossless code $\gamma = (\gamma_1, \ldots, \gamma_L)$ as follows. For each $\ell \in \{1, \ldots, L\}$, let

$$b_\ell = \max_{k_{\ell-1} \in \mathcal{K}_{\ell-1}} \lceil \log |\mathcal{K}_\ell(k_{\ell-1})| \rceil.$$

Then for each $k_{\ell-1} \in \mathcal{K}_{\ell-1}$, give all indices $k_\ell \in \mathcal{K}_\ell(k_{\ell-1})$ distinct fixed-rate binary descriptions of length b_ℓ.

Each iteration requires three steps that sequentially optimize the quantizer encoder α for the given quantizer decoder β and lossless code γ, the quantizer decoder β for the given quantizer encoder α and lossless code γ, and the lossless code γ for the given quantizer encoder α and quantizer decoder β. The algorithm is run to convergence. The three steps required in each iteration are enumerated as follows.

1) *Nearest Neighbor Encoding:* Given a fixed β and γ, the mapping $\alpha : \mathcal{X}^n \to \mathcal{K}$ that minimizes (2) is the mapping that minimizes

$$\sum_{\ell=1}^{L} \frac{1}{n} \left(a_\ell \rho_n^{(\ell)}(x^n, \beta_\ell(\alpha^\ell(x^n))) \right) + b_\ell |\gamma_\ell(\alpha^\ell(x^n))|)$$

pointwise, giving

$$\alpha^\star(x^n) = \arg \min_{\boldsymbol{k} \in \mathcal{K}} \sum_{\ell=1}^{L} \left(a_\ell \rho_n^{(\ell)}(x^n, \beta_\ell(k^\ell)) + b_\ell |\gamma_\ell(k^\ell)| \right). \quad (3)$$

The function $\alpha^\star$ is not unique (e.g., ties may be broken arbitrarily). Encoder $\alpha^\star$ is analogous to the nearest neighbor encoder in single-resolution source coding.

2) *Decoding to the Centroid:* Given a fixed α and γ, the mapping $\beta : \mathcal{K} \to \mathcal{Y}_{(1)}^n \times \cdots \times \mathcal{Y}_{(L)}^n$ that minimizes (2) is the mapping that minimizes

$$\frac{1}{n} E_\mu \left[a_\ell \rho_n^{(\ell)} \left(X^n, \beta_\ell(k^\ell) \right) + b_\ell |\gamma_\ell(k^\ell)|) | \alpha^\ell(X^n) = k^\ell \right]$$

for each $\ell \in \{1, \ldots, L\}$ and each k^ℓ describing a partial path through the tree-structured codebook, giving

$$\beta_\ell^\star(k^\ell) = \arg \min_{y^n \in \mathcal{Y}_{(\ell)}^n} E_\mu \left[\rho_n^{(\ell)}(X^n, y^n) \middle| \alpha^\ell(X^n) = k^\ell \right]. \quad (4)$$

Thus, each reproduction $\beta_\ell^\star(k^\ell)$ lies at the centroid of the region $\mathcal{X}^n$ of values that map to k^ℓ in their first ℓ resolution descriptions. If $\rho_\ell(x, y) = (x - y)^2$ (the squared

[5] Recall that R_i denotes an incremental rate; the total rate used in the resolution-ℓ description is $\sum_{i=1}^{\ell} R_i$.

[6] The same code was called a multiresolution TSVQ on its introduction in [19].

error distortion measure) then the centroid is the conditional expectation $\beta_\ell^\star(k^\ell) = E_\mu\left[X^n \middle| \alpha^\ell(X^n) = k^\ell\right]$. If $\rho_\ell(x,\hat{x}) = |x - \hat{x}|$ (the absolute difference distortion measure) then the centroid is the conditional median of the probability density function in that region $\beta_\ell^\star(k^\ell) = \mathcal{M}\left[X^n \middle| \alpha^\ell(X^n) = k^\ell\right]$ [38]. Centroids for a variety of other common distortion measures appear in [23].

3) *Optimizing the Prefix Code:* For fixed-rate codes, this step causes no change. For variable-rate codes, given a fixed α and β, the optimal prefix-code $\gamma^\star : \mathcal{K} \to \mathcal{S}$ to minimize (2) is an entropy code matched to the probabilities

$$\Pr(\alpha_\ell(X^n) = k_\ell | \alpha^{\ell-1}(X^n) = k^{\ell-1}).$$

Thus, the ideal codeword lengths are

$$|\gamma_\ell^\star(k_\ell)| = -\log \Pr\left(\alpha_\ell(X^n) = k_\ell \left| \alpha^{\ell-1}(X^n) = k^{\ell-1}\right.\right) \quad (5)$$

for each $\ell \in \{1,\ldots,L\}$ and each k^ℓ describing a partial path through the tree-structured codebook.

The above algorithm jointly optimizes the full code rather than designing the code one resolution at a time. At each step in each iteration, the Lagrangian functional (2) cannot increase. Since the functional is bounded below by 0, the algorithm is guaranteed to converge. Since each step produces a global minimum of (2) relative to the fixed source coding components, the algorithm is guaranteed to converge to a local optimum.

For many practical coding applications, the underlying source distribution μ is not known. In this case, expectations may be taken with respect to an empirical distribution $\hat{\mu}$ on $\mathcal{X}^n$ derived from a representative training set.

V. Complexity and Fast Approximations

The MRVQ code implementation uses an explicit minimization of (3) in the quantizer encoder; the lossless code and quantizer decoder are implemented using table lookup. As a result, the run-time complexity of MRVQ, like that of VQ, is dominated by the complexity of the quantizer encoder. MRVQ encoding complexity is roughly equivalent to (nearest neighbor) VQ encoding complexity given a codebook size equal to the number of nodes in the multiresolution tree. For example, for a binary tree, the MRVQ encoder has complexity roughly twice that of a VQ encoder with the same (maximal) rate since the number of nodes in a binary tree is roughly twice the number of leaves in the tree. The key difference between the MRVQ and TSVQ encoders is that the MRVQ encoder compares all paths through the tree (which requires 2^L n-dimensional distortion calculations in a depth-L binary tree) while the TSVQ algorithm optimizes for one layer at a time (which requires $2L$ n-dimensional distortion calculations in the same tree). Thus, for a binary tree of depth L, MRVQ complexity is exponential in L while TSVQ complexity is linear in L. We next introduce an algorithm that simultaneously approximates MRVQ performance and TSVQ complexity.

A linear complexity MRVQ (LMRVQ) combines an MRVQ decoder with an encoder that tracks, at each resolution, the N best paths, thereby obtaining complexity that is linear in the tree depth L. The performance measure at resolution ℓ is

$$\sum_{i=1}^{\ell} E_\mu\left[a_i \rho_n^{(i)}\left(X^n, \beta_i(\alpha^i(X^n))\right) + b_i |\gamma_i(\alpha^i(X^n))|\right].$$

Setting $N = 1$ yields a TSVQ encoder. Section VII gives experimental results for the LMRVQ. On the example image data set, the LMRVQ with $N = 2$ (which requires twice the complexity of the corresponding TSVQ) achieves performance comparable to that of the MRVQ. The MRVQ for the same example requires complexity roughly 28 times that of the corresponding TSVQ. As implemented, the LMRVQ uses an MRVQ codebook and therefore requires design complexity identical to that for MRVQ.[7]

An alternative approach to low-complexity encoding would be to replace the optimal encoder with a sequence of table lookups using the hierarchical approach of [39]–[41].

VI. Choosing (a^L, b^L)

We next consider the problem of how to choose (a^L, b^L) to meet functional constraints on the desired code. For example, suppose that proportion p_ℓ of all users want to receive information at rate $\sum_{\ell=1}^{L} R_\ell$. A system designer wishes to design the variable-rate code that minimizes $\sum_{\ell=1}^{L} p_\ell D_\ell$ subject to the constraint that the total rate in resolution ℓ must be less than or equal to $\sum_{i=1}^{\ell} R_i$ for each $\ell \in \{1,\ldots,L\}$.[8] We next describe a systematic search algorithm for finding the parameters (a^L, b^L) for use in (2).

To incorporate the given priorities in our Lagrangian parameters and maintain the symmetry between rate and distortion, we set $a_\ell = (1-c)p_\ell$ and $b_\ell = cq_\ell$ for some $c \in [0,1]$ and $q^L \geq 0$ with $\sum_{\ell=1}^{L} q_\ell = 1$. There is no loss of generality in this choice since only the relative values of these parameters are meaningful [18, Lemma 4].

The following argument allows us to further restrict the space of q^L vectors over which we must search. The time-sharing argument used to prove the convexity of the space of achievable (incremental) rates and distortions (R^L, D^L) [18, Lemma 2] can also be applied to prove the convexity of the space of achievable total rates and distortions $\left(R_T^L, D^L\right)$, where $R_{T,\ell} = \sum_{i=1}^{\ell} R_i$ for all $\ell \in \{1,\ldots,L\}$. The corresponding Lagrangian is

$$\begin{aligned} J &= \sum_{\ell=1}^{L} \left[a_\ell D_\ell + \lambda_\ell R_{T,\ell}\right] \\ &= \sum_{\ell=1}^{L} \left[a_\ell D_\ell + \lambda_\ell \sum_{i=1}^{\ell} R_i\right] \\ &= \sum_{\ell=1}^{L} \left[a_\ell D_\ell + \left(\sum_{i=\ell}^{L} \lambda_i\right) R_\ell\right]. \end{aligned} \quad (6)$$

[7]Use of the LMRVQ encoder in code design would improve the design complexity, perhaps at the cost of some performance degradation.

[8]Alternative problem formulations (e.g., matching distortion constraints given priorities over the rates) can be handled similarly.

Thus, $b_\ell = cq_\ell = \sum_{i=\ell}^{L} \lambda_i$ and $\lambda_i \geq 0$ for all i imply that we need only consider $q^L \in \mathcal{P}_L$, where

$$\mathcal{P}_L = \left\{ q^L : \sum_{\ell=1}^{L} q_\ell = 1 \wedge q_1 \geq \cdots \geq q_L \geq 0 \right\}.$$

(As in ECVQ, the restriction $\lambda_i \geq 0$ results from the Lagrangian formulation; intuitively, $\lambda_i < 0$ would force the rate term to grow without bound.)

Since p^L is given, it remains only to choose the value of $(c, q^L) \in [0,1] \times \mathcal{P}_L$. A wide variety of techniques can be applied to search for an optimal $(c^\star, (q^L)^\star) \in [0,1] \times \mathcal{P}_L$. The simple method that follows takes a bisection-style approach. Gradient descent techniques are also possible.

We set $(\underline{c}_0, \overline{c}_0) = (0,1)$ and choose an initial value $\left(c_0, q_0^L\right)$ in some central location in the allowed region $[\underline{c}_0, \overline{c}_0] \times \mathcal{P}_L$. For example, when $L = 3$

$$\mathcal{P}_L = \left\{ q^3 : \sum_{\ell=1}^{3} q_\ell = 1 \wedge q_1 \geq q_2 \geq q_3 \geq 0 \right\}$$

and we choose $\left(c_0, q_0^3\right) = (1/2, (11/18, 5/18, 2/18))$. At each time $t \geq 0$, we design an MRVQ for Lagrangian parameters $(a^L, b^L) = \left((1-c_t)p^L, c_t q_t^L\right)$, calculate the resulting performance (R^L, D^L), and find $\left(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q_{t+1}^L\right)$ according to the following rules.

At time t, define

$$A = \{\ell : R_\ell < R_\ell^\star - \epsilon\} \quad \text{and} \quad B = \{\ell : R_\ell > R_\ell^\star + \epsilon\}$$

where $\epsilon > 0$ describes a target margin of error (i.e., we are aiming for $R_\ell \in [R_\ell^\star - \epsilon, R_\ell^\star + \epsilon]$).[9] Then any code for which B is empty meets our design criteria, but codes for which both A and B are empty use the full available rate and thus have the potential to achieve a lower value of $\sum_{\ell=1} a_\ell D_\ell$. We therefore run the following iterative search procedure until at least B is empty.[10]

- If A and B are both empty, then the procedure stops.
- Otherwise, if B is empty, then

$$\left(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q_{t+1}^L\right) = \left(\underline{c}_t, c_t, (\underline{c}_t + c_t)/2, q_t^L\right).$$

- Otherwise, if A is empty, then

$$\left(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q_{t+1}^L\right) = \left(c_t, \overline{c}_t, (c_t + \overline{c}_t)/2, q_t^L\right).$$

- Otherwise, we leave $\underline{c}$, $\overline{c}$, and c unchanged (giving $(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}) = (\underline{c}_t, \overline{c}_t, c_t)$) and search the space $\mathcal{P}_L$ of allowed q^L vectors using the iterative approach described below. This procedure outputs a modified vector $q_{t+1}^L \neq q_t^L$ such that at least one of A and B is empty for parameters $\left(\underline{c}_{t+1}, \overline{c}_{t+1}, c_{t+1}, q_{t+1}^L\right)$.

[9]Asymmetrical error margins ($R_\ell \in [R_\ell^\star - \epsilon, R_\ell^\star]$) and multiplicative error margins ($R_\ell \in [R_\ell^\star(1-\epsilon), R_\ell^\star(1+\epsilon)]$) can be handled similarly.

[10]In theory, it will not always be possible to find a code with both A and B empty since we are restricting our attention to codes whose performance lies on the lower convex hull of achievable (R^L, D^L) vectors. Experimentally, the set of points on the lower convex hull seems to be extremely rich for the sources considered here, and thus this problem has not been observed in practice.

Given a fixed c and some initial $q_t^L \in \mathcal{P}_L$ for which sets A and B are both nonempty, the procedure for searching the space $\mathcal{P}_L$ of allowed q^L vectors likewise uses an iterative approach. Since both A and B are nonempty, we rule out the subspace

$$\left\{ q^L \in \mathcal{P}_L : [q_i > q_{t,i} \; \forall \, i \in A] \wedge [q_j < q_{t,j} \; \forall \, j \in B] \right\}$$

choose a central point in the region that remains, test the resulting rates, and continue the iterative procedure until achieving a point for which at least one of A and B is empty. In the procedure used for the experimental results section, the choice of a tentative value for q_{t+1}^L given q_t^L maintains the ratios $q_{t+1,i}/q_{t+1,j} = q_{t,i}/q_{t,j}$ for all $(i,j) \in (A \times A) \cup (B \times B)$, giving

$$q_{t+1,\ell} = \begin{cases} a q_{t,\ell}, & \text{for all } \ell \in A \\ \left(1 + \frac{(1-a)q(A)}{q(B)}\right) q_{t,\ell}, & \text{for all } \ell \in B \\ q_{t,\ell}, & \text{for all } \ell \in A^c \cap B^c \end{cases}$$

where $q(A) = \sum_{i \in A} q_{t,i}$, $q(B) = \sum_{j \in B} q_{t,j}$, and $a < 1$ is the midpoint of the segment of values for which the resulting $q_{t+1,\ell}$ falls in the unsearched remaining region of q^L. The procedure is monotonic. By shrinking the subspace of values that must be searched at each step, the algorithm narrows its way to a solution.

VII. Experimental Results

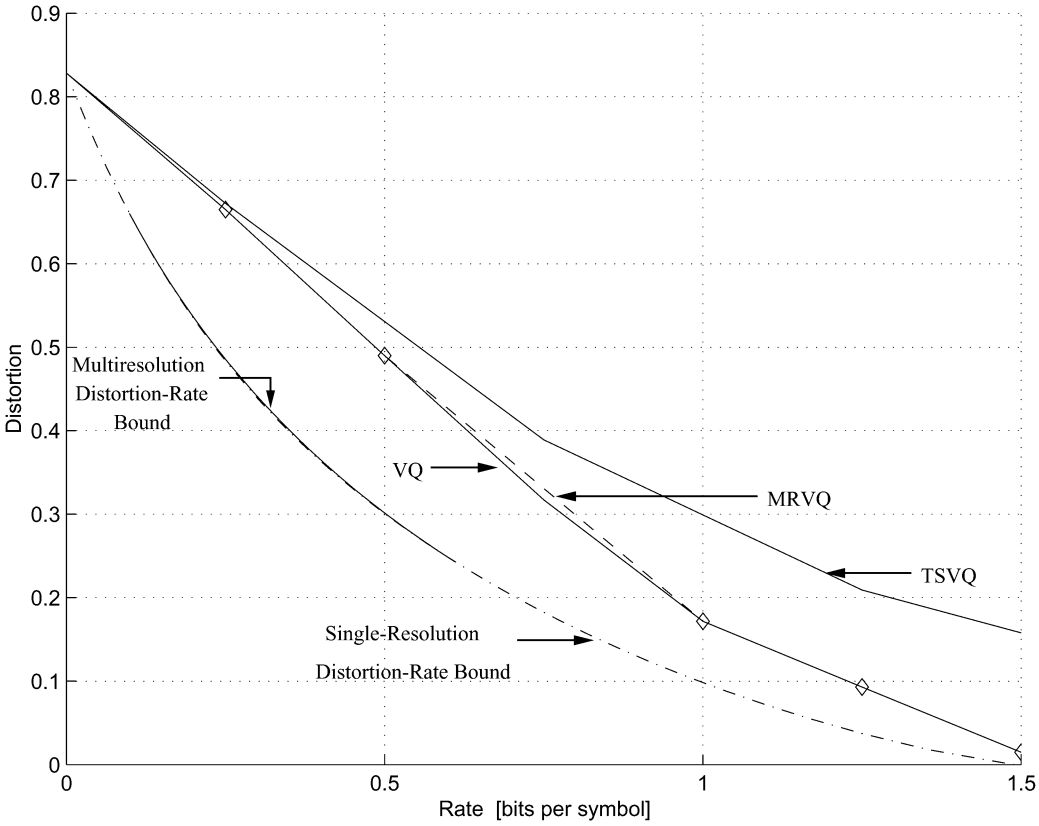
We next examine the empirical performance of fixed- and variable-rate multiresolution codes designed using the MRVQ design algorithm. We compare their performance to both the theoretically optimal performance (where available) and the performance of alternative single- and multiresolution vector quantizers of the same dimension n. We also investigate MRVQ convergence properties for growing vector dimension n and LMRVQ performance for increasing search complexity N.

A. Synthetic Data

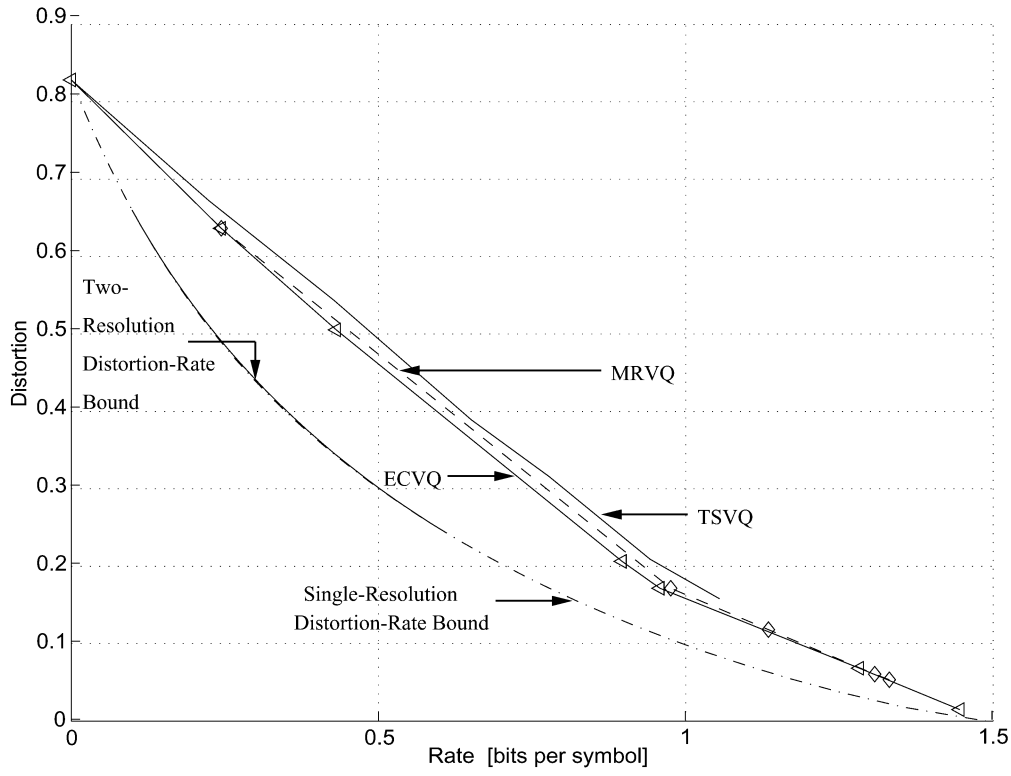
The synthetic data set consists of 2^{20} independent and identically distributed (i.i.d.) samples drawn according to the distribution $\mu = \{(1-p)/2, p, (1-p)/2\}$ on alphabet $\{1,2,3\}$ with $p = 0.171$. This Gerrish distribution is treated in [7], [8], [18]. We use half of the data samples for training and report results on the remaining half. The distortion measure is $d(x, \hat{x}) = |x - \hat{x}|$.

While the theoretical results of [18] demonstrate that the penalty associated with using a multiresolution code on the given three-symbol source is very small, those results treat only the asymptotic case, where the coding dimension n is allowed to be arbitrarily large. The results of Fig. 2 give empirical evidence suggesting that similar statements hold on this source even at very small coding dimensions. Fig. 2 shows the performance of a) fixed-rate and b) variable-rate MRVQ of dimension $n = 4$. In both the fixed- and the variable-rate examples, the MRVQ achieves second-resolution performance very near to the performance of the best single-resolution code of the same dimension and gives better performance than a TSVQ of the same dimension.

Fig. 3 demonstrates the performance improvements associated with increasing the coding dimension n. The performance

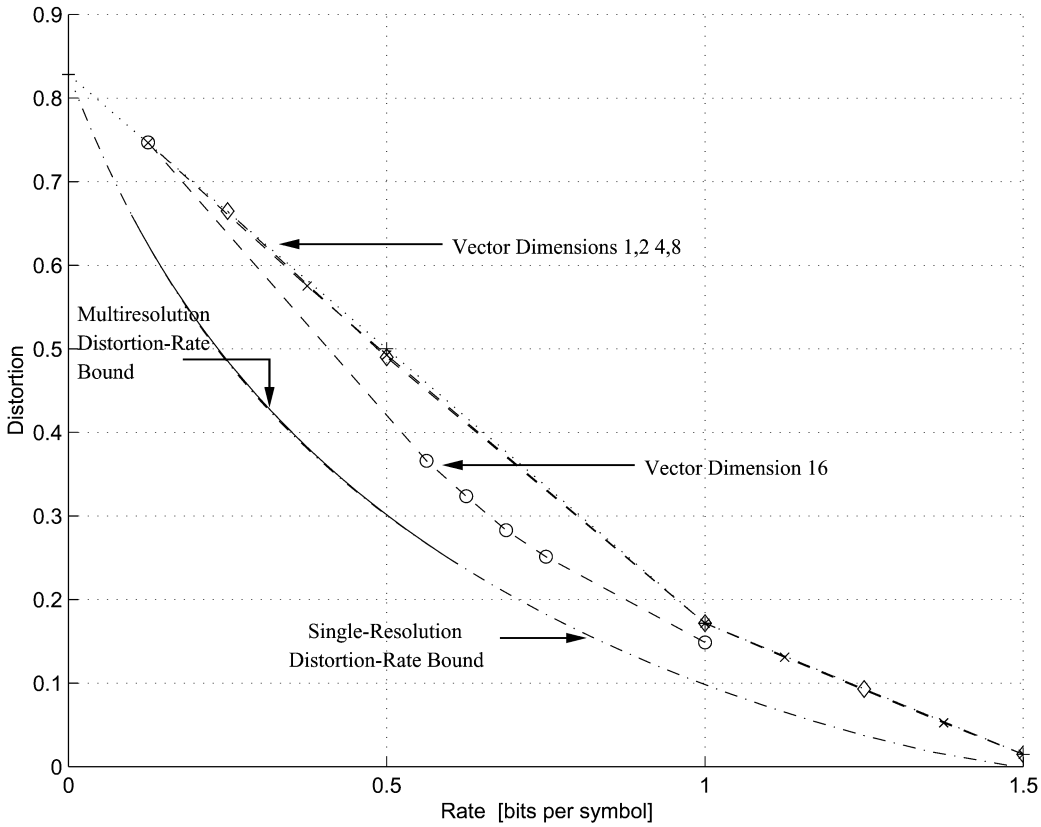


(a) Fixed-rate codes

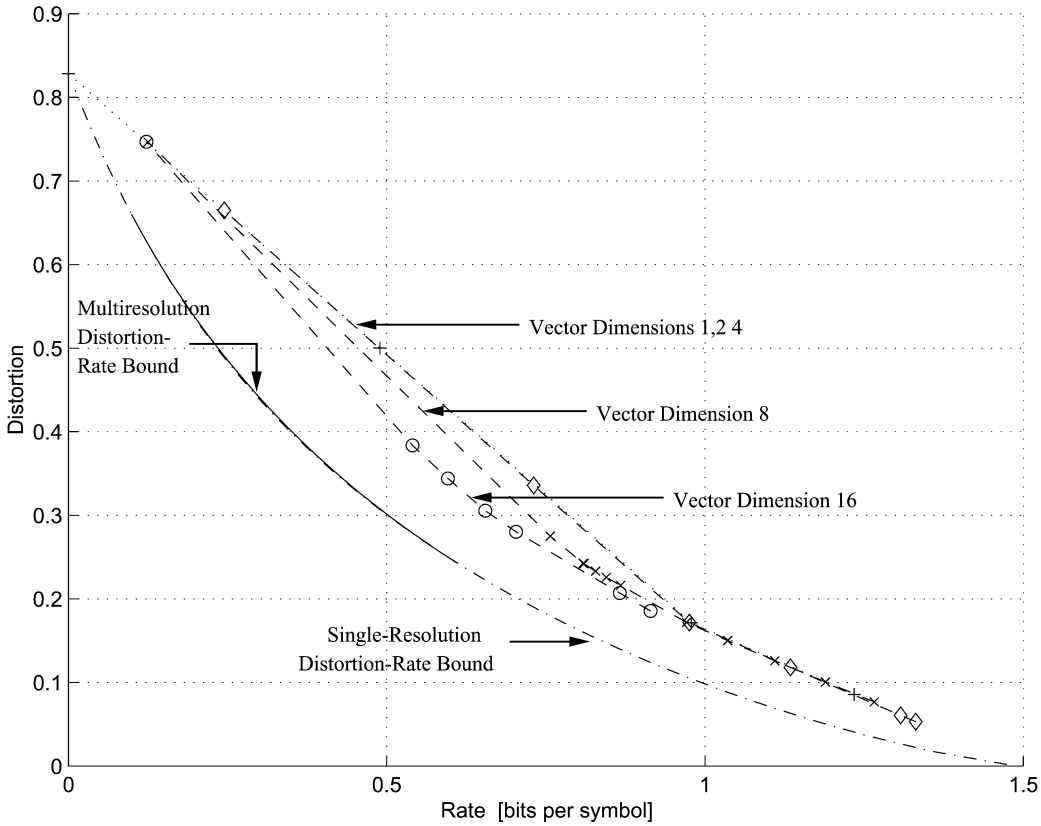


(b) Variable-rate codes

Fig. 2. Performance comparison for fixed- and variable-rate MRVQ, fixed- and variable-rate TSVQ (single-resolution) fixed-rate VQ and ECVQ, and the theoretical bound. Results are given for $n = 4$ and the synthetic data set. The MRVQ curve gives the second-resolution performances of MRVQs with first-resolution performance identical to that of the best rate-0.25 VQ in the fixed-rate case and the best rate-0.246 ECVQ in the variable-rate case. The multiresolution distortion-rate bound shows several curves; the curves are so similar that they are indistinguishable. Each curve shows the distortion-rate performance in resolution 2 when the first resolution performance sits on the distortion-rate curve at a single fixed rate R_1; the curves given correspond to a variety of different values of R_1.

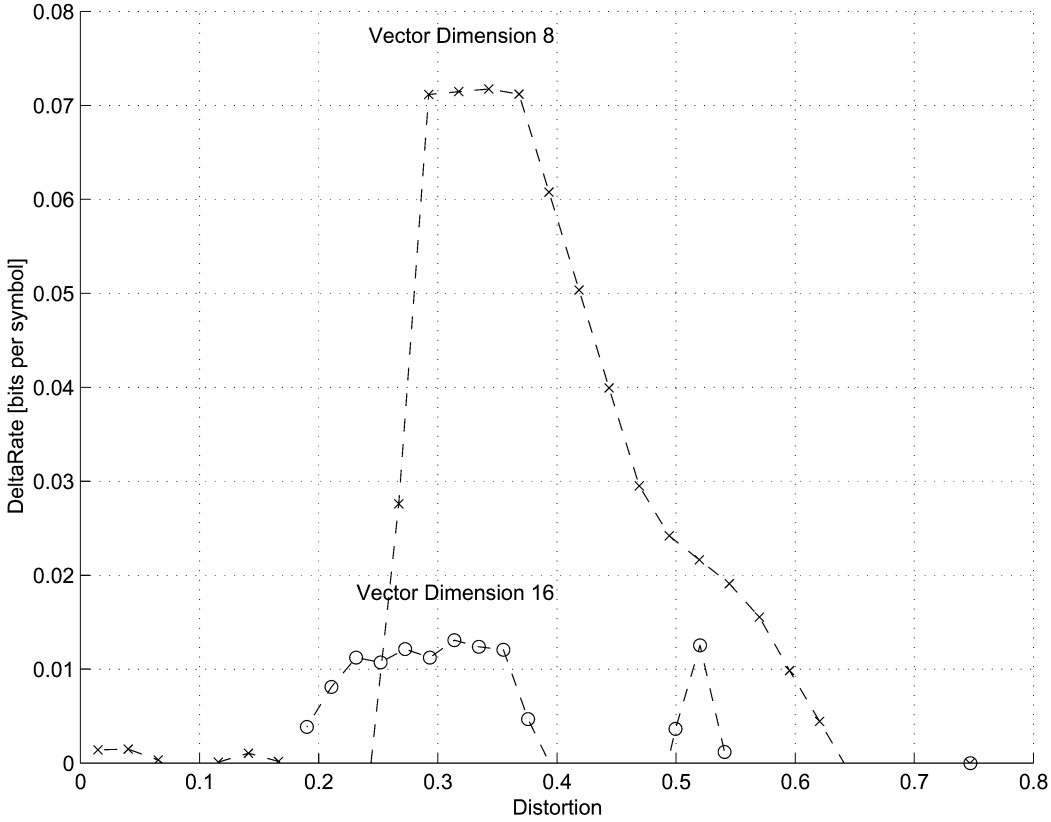


(a) Fixed-rate codes

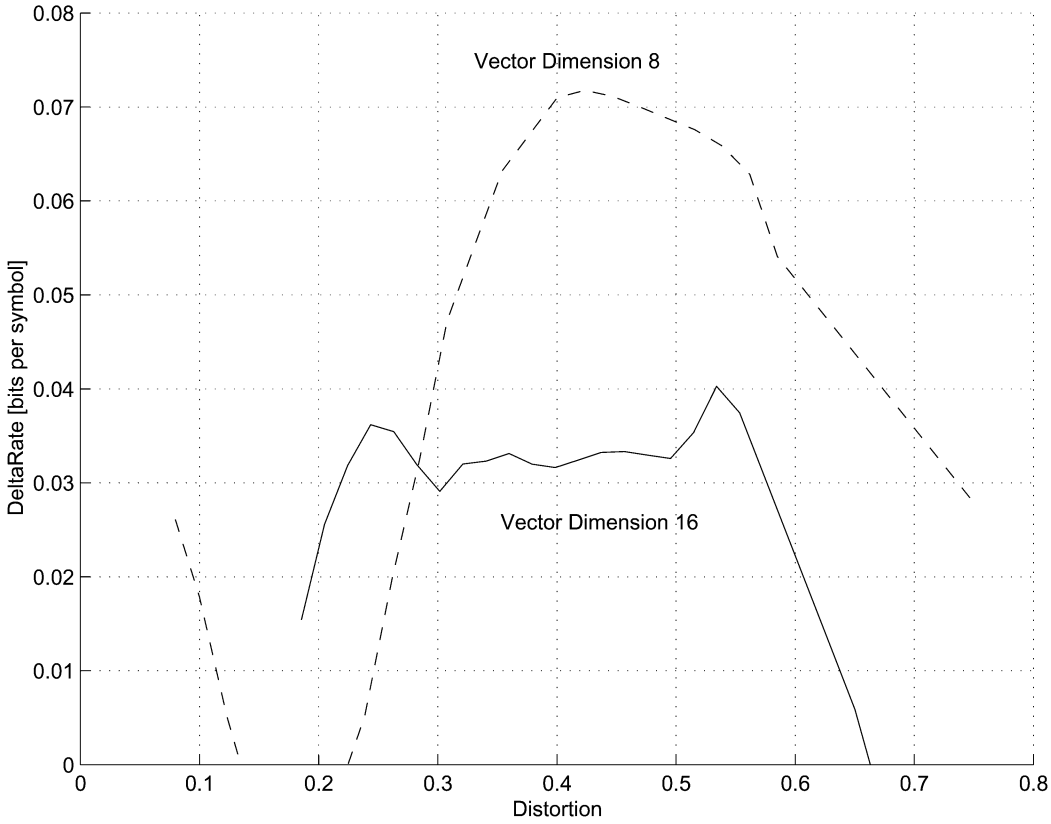


(b) Variable-rate codes

Fig. 3. MRVQ performance in resolution 2 of a code with $L = 2$ for vector dimensions $1, 2, 4, 8$, and 16 on the synthetic data set. For each multiresolution code, the graphs show the rate-distortion performance in the second resolution when the first-resolution performance is constrained to be the best available at rate 0.125 bps.



(a) Fixed-rate codes



(b) Variable-rate codes

Fig. 4. The second-resolution rate penalty for 8- and 16-dimensional MRVQ on the synthetic data set. For each distortion value, the graphs show the difference between the rate required to achieve that distortion in the second resolution of an MRVQ and the rate required to achieve that distortion with a single-resolution code. In each case, the first resolution of the MRVQ is constrained to achieve performance identical to that of the best corresponding single-resolution code at rate $\simeq 0.125$ bps.

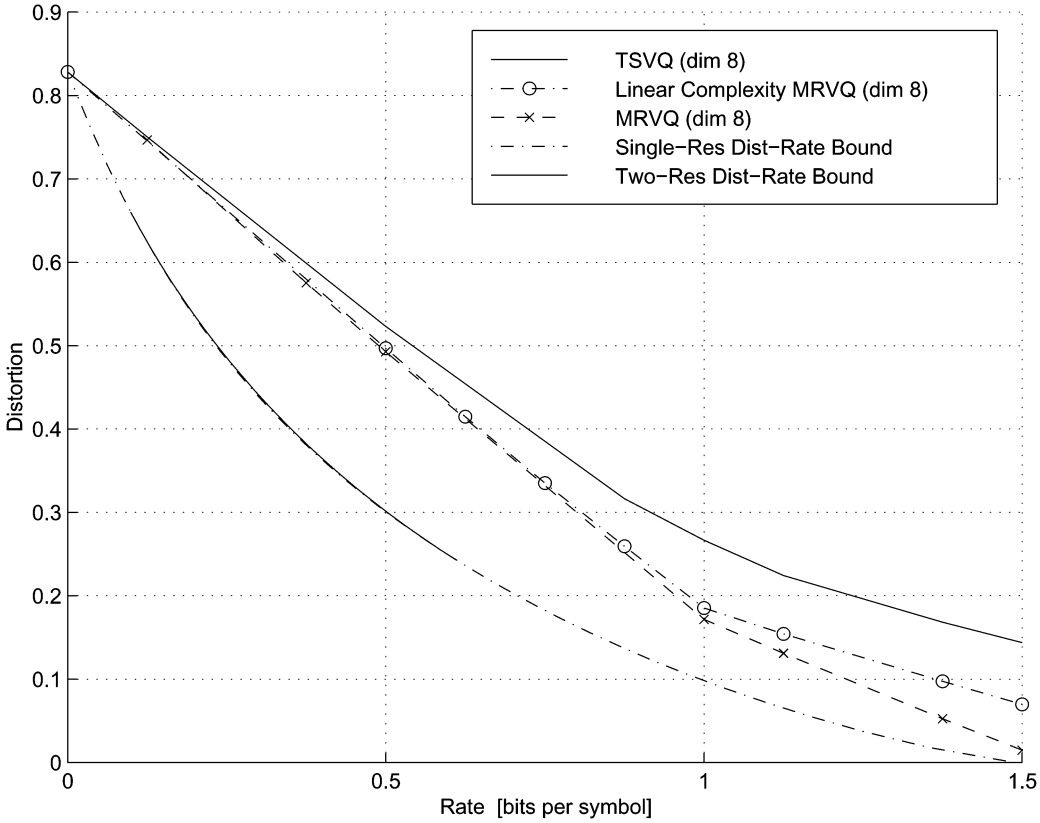


Fig. 5. Distortion-rate performance of LMRVQ ($N = 1$), TSVQ, MRVQ, and the theoretically optimal performance on the synthetic data set. The LMRVQ approximates MRVQ performance using TSVQ complexity. All codes are fixed rate with $n = 8$.

is not necessarily monotonic in n, and in these experiments it changes very little for the first few dimensions. This behavior likely results from the independence of the source samples and the low dimension of the codes. (The vector quantization advantage is generally attributed to the ability of high-dimensional codes to take advantage of correlation between data samples and to the economies of scale that come with large coding dimensions [42].) Increasing n to 16 gives significant performance improvement. Further improvement could be obtained by increasing the dimension even more. The penalty for this improvement is an increase in computational complexity, as discussed in Section V.

Fig. 4 characterizes, for several dimensions, the second-resolution rate penalty associated with multiresolution coding. These low-dimension experimental results are analogous to the asymptotic theoretical results of [18]. The rate penalty varies as a function of both the first-resolution rate and the coding dimension, never exceeding 0.08 bps in this example.

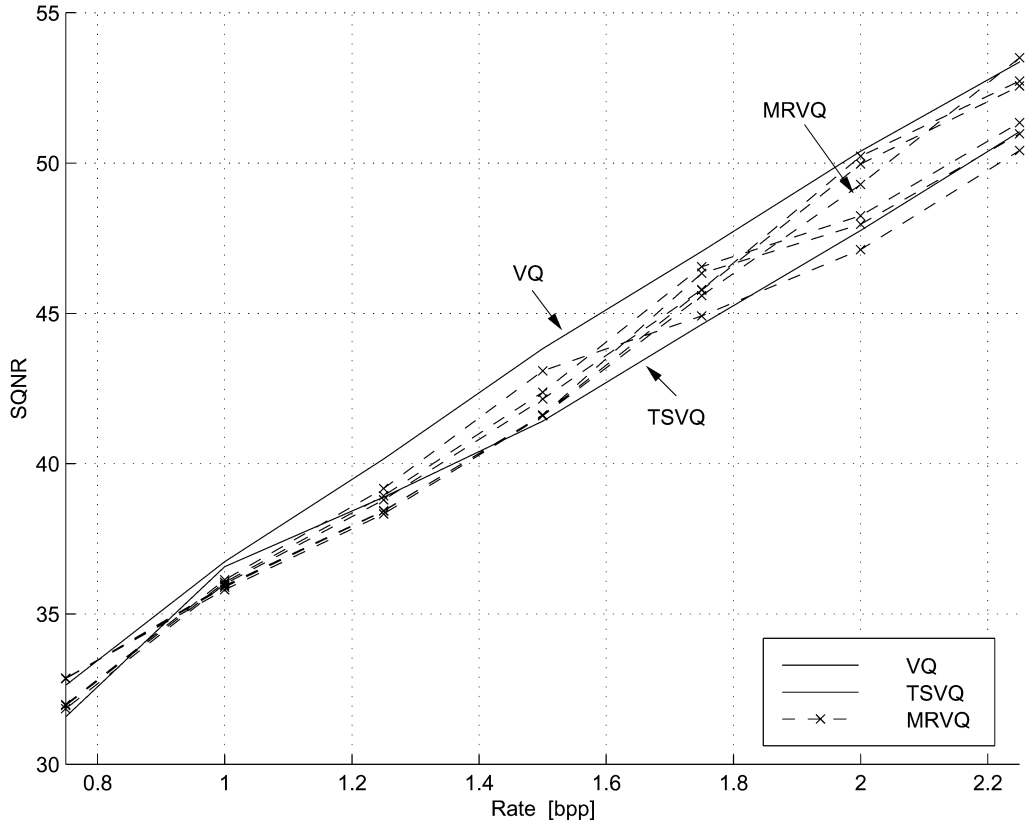
Fig. 5 compares the performances of fixed-rate LMRVQ ($N = 1$), MRVQ, and TSVQ. The LMRVQ approximates MRVQ performance using TSVQ complexity.

B. Natural Data

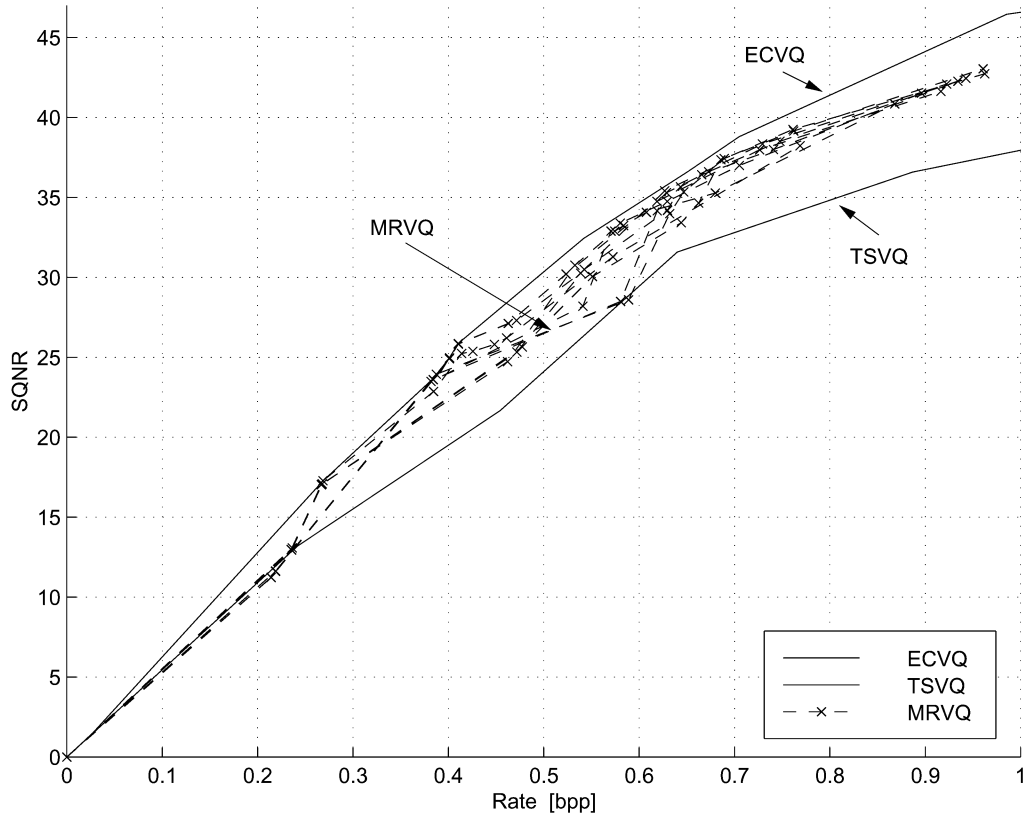
The natural data set consists of 25 256×256 medical brain scans: 20 training images and 5 test images. All experiments use $d(x, \hat{x}) = (x - \hat{x})^2$ and $n = 4$.

Fig. 6 compares the a) fixed-rate and b) entropy-constrained performances of MRVQ, VQ, and TSVQ on the natural data set. Distortion is given as signal to quantization noise ratio (SQNR). MRVQ performance can be made identical to VQ or ECVQ performance at a single resolution if the priority at that resolution is sufficiently high. The potential expense of this choice is a degradation of the performance of the code at another resolution. Fig. 6 includes examples both of cases where the MRVQ performance is set equal to the corresponding VQ or ECVQ performance at a given resolution—giving the best possible performance at the given resolution but causing performance degradation at other resolutions—and examples where the MRVQ performance is everywhere near but nowhere equal to the performance of the best single-resolution code. The MRVQ exceeds the performance of the TSVQ except, occasionally, at the lowest rates where the TSVQ's "greedy" strategy can give good performance. Fig. 7 shows examples of compressed images from single- and multiresolution codes.

To investigate the choice of a^L and b^L from functional constraints and to compare the performance of codes designed with MRVQ's Lagrangian performance measure (2) with that of codes designed with the weighted performance measure, we consider the following example. We wish to design a 3-resolution code for the natural data set with incremental rates $R_1 = R_2 = R_3 = 0.25$ bps (giving total rates $0.25, 0.5, 0.75$) and priorities $p_1 = p_2 = p_3 = 1/3$ on the three distortions. The method for choosing a^L and b^L described in Section VI gives $a^3 = [1\ 1\ 1]$ and $b^3 = [16100\ 1560\ 80]$. The resulting performance is shown in Fig. 8. The performances of two codes designed using the weighted performance are shown in the same figure. Due to the smaller number of degrees of freedom



(a) Fixed-rate codes



(b) Variable-rate codes

Fig. 6. SQNR versus rate results for fixed- and variable-rate MRVQ (single-resolution) VQs and ECVQs, and fixed- and variable-rate TSVQ on the medical image data set.

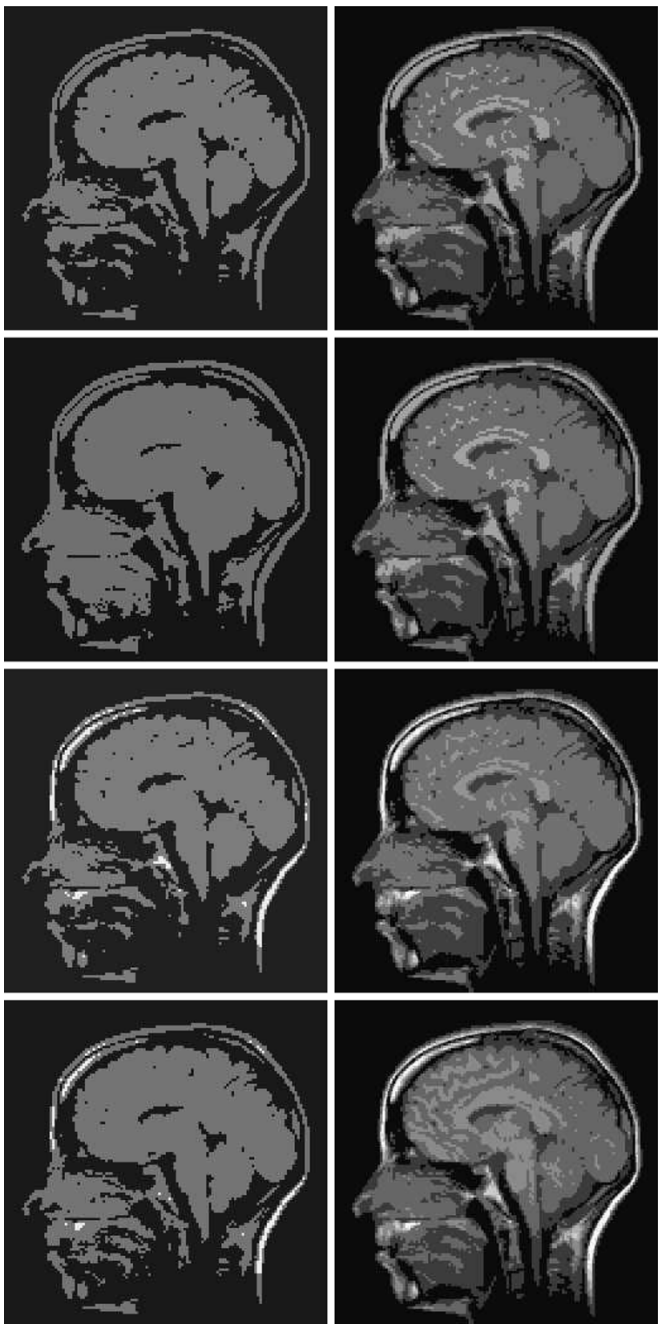


Fig. 7. Sample compressed images from single- and multiresolution codes. In each case, the same image from the test set is shown. Row 1: Fixed-rate MRVQ, resolutions 1 (rate 0.25, distortion 878.28) and 2 (rate 0.50, distortion 321.96). Row 2: Fixed-rate VQ (two independent single-resolution codes), codes 1 (Rate 0.25, distortion 761.95) and 2 (rate 0.50, distortion 316.95). Row 3: Variable-rate MRVQ, resolutions 1 (rate 0.25, distortion 791.96) and 2 (rate 0.50, distortion 220.81). Row 4: ECVQ (two independent single-resolution codes), codes 1 (rate 0.25, distortion 705.87) and 2 (rate 0.50, distortion 207.95).

available in weighted code design, designing a code that approximates the first rate bound of 0.25 yields a code for which all three resolutions give identical performance; here setting λ large enough to achieve rate 0.25 in resolution 1 makes the rate constraint too tight in resolutions 2 and 3. Designing a code that approximates the third rate bound of 0.75 yields a code that achieves good performance in resolution 3 but violates the first two rate constraints; in this case, choosing λ small enough to achieve rate 0.75 in resolution 3 makes the rate constraints too loose in resolutions 1 and 2.

Fig. 9 compares the performance of fixed-rate MRVQ and TSVQ to that of fixed-rate LMRVQ with $N \in \{2, 4\}$. All codes use depth-9, binary tree-structured codebooks with $n = 4$. The codebooks for the MRVQ and LMRVQ results are identical, and the encoders for these codes use the same values of a^L and b^L.

VIII. Summary

This paper introduces optimal vector quantizers for multiresolution source coding and presents the MRVQ algorithms for

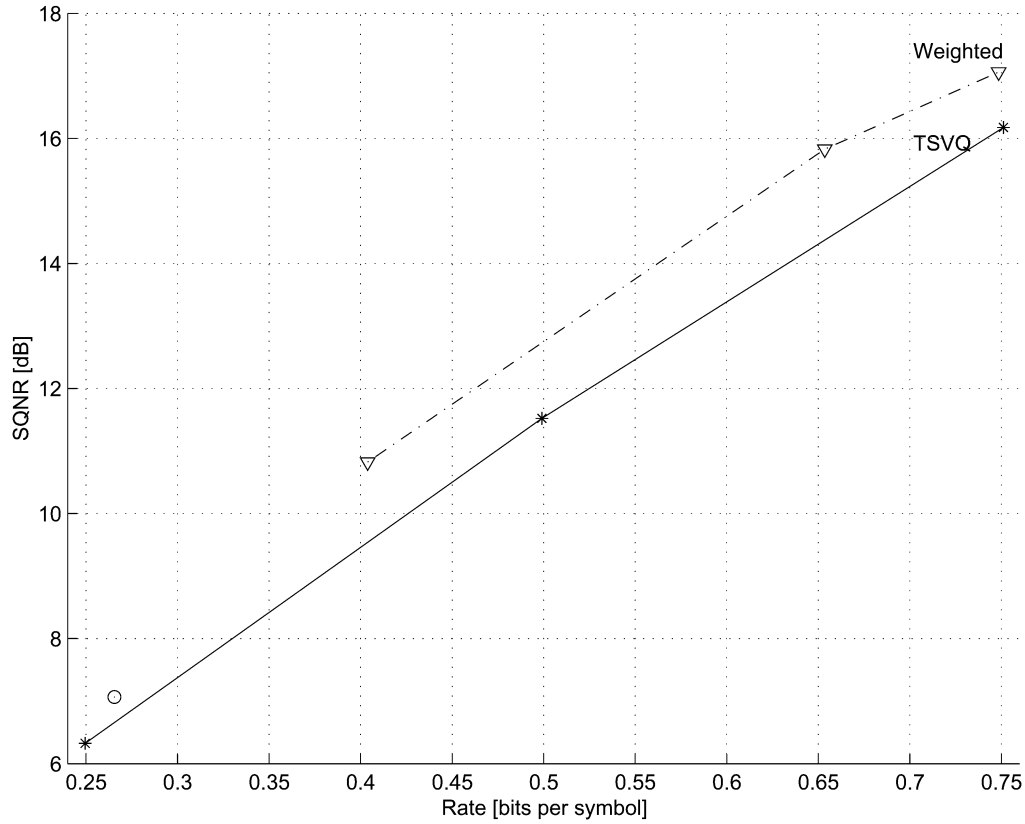


Fig. 8. SQNR versus rate results for three 3-resolution codes designed to achieve priorities $[1/3, 1/3, 1/3]$ and total rates $[0.25, 0.5, 0.75]$ on the natural data set with $n = 4$. The code designed with the Lagrangian performance (asterisks) achieves the target rates precisely. Two codes are designed with the weighted approach. The first (circle) approximates the first-resolution rate bound but achieves identical performance in all resolutions. The second (triangles) achieves high SQNR in the third resolution but violates the first- and second-resolution rate constraints.

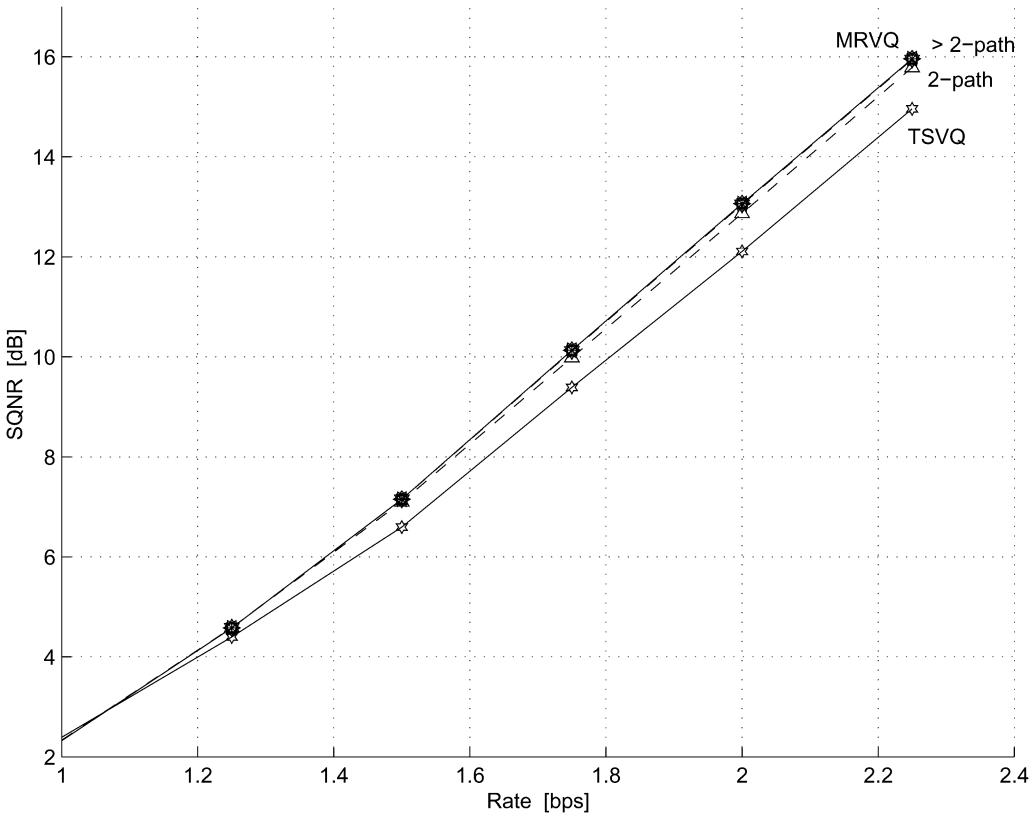


Fig. 9. SQNR versus rate performance for fixed-rate LMRVQ ($N \in \{2, 4\}$), MRVQ, and TSVQ on the natural data set with $n = 4$. The performance of the LMRVQ for $N = 4$ is almost identical to that of the MRVQ.

locally optimal multiresolution vector quantizer design. The MRVQ design algorithm is an iterative descent technique on a Lagrangian performance measure (2). The algorithm guarantees convergence to a locally optimal solution. The family of codes achievable through the MRVQ design algorithm is parameterized by the corresponding Lagrangian parameters. Graphically, the Lagrangian parameters describe a hyperplane tangent to the convex hull of the space of achievable rate-distortion vectors at a single point; choice of the Lagrangian parameters is equivalent to choice of that point. We propose a simple technique for choosing the appropriate parameters for an arbitrary set of functional constraints.

The relationship between the MRVQ design algorithm and the iterative descent algorithm for multiresolution scalar quantizers described in [35]–[37] appears initially to be parallel to the relationship of the generalized Lloyd algorithm for VQ design to the Lloyd algorithm [43] for scalar quantizer design; in both cases, there is a generalization from a description based on thresholds and boundaries in the scalar problem to a description based on codewords and encoding regions in the vector case. This parallel is, however, misleading for several reasons. First, the difference between (2) and the weighted optimality criterion of the earlier work is critical for the variable-rate case. Second, the nearest neighbor encoder of (3) allows nonconvex encoding regions that cannot be conveniently represented (and are, in practice, typically disallowed) in algorithms that rely on the threshold/boundary model used in the scalar coding case; allowing nonconvex encoding regions is critical for optimality in multiresolution code design by [44].

We give experimental results comparing MRVQ performance with both the theoretically optimal performance and the performance of a variety of single- and multiresolution vector quantizers. The MRVQ achieves performance improvements over prior codes at the expense of increased computational complexity. By replacing the exhaustive search of an MRVQ encoder with an N-path search, we approximate MRVQ performance with lower complexity. Performance results comparing MRVQ and LMRVQ to VQ and TSVQ demonstrate the codes' good performance with and without complexity constraints.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers and the Associate Editor for suggestions that improved the quality of this paper. In particular, the generalization from greedy search ($N = 1$) to N-path search ($N \geq 1$) in the LMRVQ was proposed by a reviewer.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.
[2] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[3] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[4] M. Effros, P. A. Chou, and R. M. Gray, "Variable-rate source coding theorems for stationary nonergodic sources," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1920–1925, Nov. 1994.
[5] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, Jan. 1980.
[6] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 31–42, Jan. 1989.
[7] W. H. R. Equitz, "Successive refinement of information," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1989.
[8] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory*, vol. 37, pp. 269–275, Mar. 1991.
[9] J. Chow and T. Berger, "Failure of successive refinement for symmetrical Gaussian mixtures," *IEEE Trans. Inform. Theory*, vol. 43, pp. 350–352, Jan. 1997.
[10] L. Lastras and T. Berger, "All sources are nearly successively refinable," in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, June 2000, p. 127.
[11] ——, "All sources are nearly successively refinable," *IEEE Trans. Inform. Theory*, vol. 47, pp. 918–926, Mar. 2001.
[12] H. Feng and M. Effros, "Improved bounds for the rate loss of multiresolution source codes," in *Proc. IEEE Int. Symp. Information Theory*, Washington, DC, June 2001, p. 193.
[13] ——, "Improved bounds for the rate loss of multi-resolution source codes," *IEEE Trans. Inform. Theory*, vol. 49, pp. 809–821, Apr. 2003.
[14] R. M. Gray and A. D. Wyner, "Source coding for a simple network," *Bell Syst. Tech. J.*, vol. 53, no. 9, pp. 1681–1721, Nov. 1974.
[15] I. Csiszar and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems* . New York: Academic, 1981.
[16] B. Rimoldi, "Successive refinement of information: Characterization of achievable rates," *IEEE Trans. Inform. Theory*, vol. 40, pp. 253–259, Jan. 1994.
[17] M. Effros, "Multi-resolution source coding theorems," in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 226.
[18] ——, "Distortion-rate bounds for fixed- and variable-rate multiresolution source codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1887–1910, Sept. 1999.
[19] ——, "Practical multi-resolution source coding: TSVQ revisited," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 53–62.
[20] D. Dugatkin and M. Effros, "Setting priorities: A new SPIHT-compatible algorithm for image compression," in *Proc. SPIE Int. Symp. Optical Science and Technology*, vol. 4119, San Diego, CA, July 2000, pp. 799–805.
[21] D. Dugatkin, H. Zhou, T. Chan, and M. Effros, "Lagrangian optimization of the group testing for wavelets algorithm," in *Proc. Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2002.
[22] R. M. Gray, *Probability, Random Processes, and Ergodic Properties*. New York: Springer-Verlag, 1988.
[23] A. Buzo, A. H. Gray Jr, R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 562–574, Oct. 1980.
[24] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree structured source coding and modeling," *IEEE Trans. Inform. Theory*, vol. 35, pp. 299–315, Mar. 1989.
[25] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Trans. Signal Processing*, vol. 39, pp. 2500–2507, Nov. 1991.
[26] B.-H. Juang and J. A. H. Gray, "Multiple stage vector quantization for speech coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, Paris, France, 1982, pp. 597–600.
[27] C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi, "Advances in residual vector quantization: A review," *IEEE Trans. Image Processing*, vol. 5, pp. 226–262, Feb. 1996.
[28] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation, Compression, and Standards*, 2nd ed. New York/London: Plenum, 1995.
[29] H. Brunk and N. Farvardin, "Embedded entropy-constrained trellis coded quantization," in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 274.
[30] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.
[31] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technol.*, vol. 6, pp. 243–250, June 1996.
[32] E. Ordentlich, M. Weinberger, and G. Seroussi, "A low-complexity modeling approach for embedded coding of wavelet coefficients," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 408–417.
[33] E. A. Riskin, R. Ladner, R. Wang, and L. E. Atlas, "Index assignment for progressive transmission of full-search vector quantization," *IEEE Trans. Image Processing*, vol. 3, pp. 307–312, May 1994.

[34] S. Herman and K. Zeger, "Variable fanout trimmed tree-structured vector quantization for multirate channels," in *Proc. IEEE Int. Symp. Information Theory and Its Applic.*, vol. 1, Victoria, BC, Canada, Sept. 1996, pp. 417–421.
[35] H. Brunk and N. Farvardin, "Fixed-rate successively refinable scalar quantizers," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 1996, pp. 250–259.
[36] H. Brunk, H. Jafarkhani, and N. Farvardin, "Design of successively refinable scalar quantizers," preprint, Apr. 1998.
[37] H. Jafarkhani, H. Brunk, and N. Farvardin, "Entropy-constrained successively refinable scalar quantization," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1997, pp. 337–346.
[38] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
[39] P. C. Chang, J. May, and R. M. Gray, "Hierarchical vector quantizers with table-lookup encoders," *Proc. Int. Conf. Communications*, vol. 3, pp. 1452–1455, June 1985.
[40] M. Vishwanath and P. Chou, "An efficient algorithm for hierarchical compression of video," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Austin, TX, Nov. 1994, pp. 275–279.
[41] N. Chaddha, P. A. Chou, and R. M. Gray, "Constrained and recursive hierarchical table lookup vector quantization," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1996, pp. 220–229.
[42] T. D. Lookabaugh and R. M. Gray, "High resolution quantization theory and the vector quantization advantage," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1020–1033, Sept. 1989.
[43] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–136, Mar. 1982.
[44] M. Effros and D. Muresan, "Codecell contiguity in optimal scalar quantizers," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2002, pp. 312–321.