Advances in Residual Vector Quantization: A Review

Christopher F. Barnes, *Member, IEEE*, Syed A. Rizvi, *Student Member, IEEE*, and Nasser M. Nasrabadi, *Senior Member, IEEE*

Invited Paper

***Abstract*— Advances in residual vector quantization (RVQ) are surveyed. Definitions of joint encoder optimality and joint decoder optimality are discussed. Design techniques for RVQ's with large numbers of stages and generally different encoder and decoder codebooks are elaborated and extended. Fixed-rate RVQ's, and variable-rate RVQ's that employ entropy coding are examined. Predictive and finite state RVQ's designed and integrated into neural-network based source coding structures are revisited. Successive approximation RVQ's that achieve embedded and refinable coding are reviewed. A new type of successive approximation RVQ that varies the instantaneous block rate by using different numbers of stages on different blocks is introduced and applied to image waveforms, and a scalar version of the new residual quantizer is applied to image subbands in an embedded wavelet transform coding system.**

I. Introduction

VECTOR quantization (VQ) theory seeks to determine the highest obtainable VQ performance as a function of rate and dimension, but the application of vector quantization is concerned with obtaining a high level of VQ performance at an affordable cost [1], [2]. The computation $\mathcal{C}_{VQ}$ and memory $\mathcal{M}_{VQ}$ expenditures required for VQ implementation generally depend on the VQ output rate r, the vector dimension k, and the constraints imposed on the quantizer's structure. Structurally unconstrained, exhaustive search vector quantizers (ESVQ's) have exponentially dependent costs proportional to $\mathcal{C}_{ESVQ} \approx \mathcal{M}_{ESVQ} \approx 2^{rk}$. A high ESVQ performance level can only be achieved as the vector dimension becomes large, but the exponential dependence of ESVQ cost limits the size of k that is affordable in practice. Fortunately, the imposition of carefully selected structural constraints can reduce the cost of VQ. A P-level, binary, tree-structured vector quantizer (TSVQ), for example, requires only $\mathcal{C}_{TSVQ} \approx 2 \times P$ computations for codebook searches but, unfortunately, consumes twice the memory of unconstrained quantizers $\mathcal{M}_{TSVQ} \approx 2 \times \mathcal{M}_{ESVQ}$.

A class of structural constraints that simultaneously reduces both computation and memory is the class of product code VQ structures [1]. A product code VQ is a structured vector quantizer where different components of the VQ system quantize different attributes or features of the source outputs [3]. Examples include mean-residual VQ [4], gain-shape VQ [5], and mean-gain-shape VQ [6]. Often implicit in the phrase "product code VQ" is the assumption that the components of the product code are sequentially searched in a predetermined order to find the best codevector matches for each source vector attribute. However, since some product codes have search procedures that are not sequential (e.g., exhaustive, iterative), the former codes are more appropriately referred to as "sequential search product codes" [7]. Thus, the imposition of a "product code" structure on a vector quantizer reduces only the memory requirements of VQ; the imposition of a composite "sequential search product code" structure is necessary for the simultaneous reduction of both computation and memory.

A simple type of product code VQ is a VQ with a direct sum codebook structure [8]. When the direct sum product code structure is combined with a sequential search procedure, the resulting quantizer has a sequence of encoder stages, where each stage encodes the residual (error) vector of the prior stage. Such sequential search product code VQ's are called *residual vector quantizers* (RVQ's). Residual vector quantizers are also called multiple stage VQ's [9], multiple step VQ's [10], cascaded VQ's [11], and summation product code VQ's [3]. The terminology "residual quantization" is also used to describe multiple stage VQ's, where the residual error is iteratively fed back into a single codebook [12]. Such quantizers are more appropriately referred to as recursive quantizers [13].

A. Performance Potential and Limitations

Although all structurally constrained quantizers are incapable of providing performance as good as that of unstructured quantizers for a given rate and dimension, experience has shown that structured quantizers sometimes provide superior performance for a given cost. This is because structured quantizers more efficiently implement codes with larger vector dimension, and since VQ performance generally improves with increasing dimensionality, with careful design, structured VQ's achieve better performance.

When investigating new product code structures that are sequentially searched, the respective limitations placed on the performance potential of the structured VQ by both the product code constraint and the sequential search constraint should be understood. One way this can be done is to first evaluate the performance of the VQ while subject only to the product code constraint (exhaustive search encoding over all

Manuscript received July 21, 1995; revised August 1, 1995.

C. F. Barnes is with the Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, GA 30332-0852 USA.

S. A. Rizvi and N. M. Nasrabadi are with the Department of Electrical and Computer Engineering, State University of New York at Buffalo, Amherst, NY 14260 USA.

Publisher Item Identifier S 1057-7149(96)01304-8.

possible product code outputs is tolerated), and then, evaluate the performance of the VQ while subject to both constraints. If necessary conditions for the optimality of the exhaustive search version of the product code can be determined, "optimal" product code VQ's can be designed and compared with "optimal" unstructured VQ's. In the first instance, the word "optimal" describes codes that satisfy necessary conditions for minimum distortion where all codes in the code search domain are structurally constrained; in the second instance, "optimal" describes codes that satisfy conditions for minimum distortion with no structural restrictions. Once the rate-distortion versus memory cost tradeoffs of the exhaustive search product codes are fully investigated and understood (computation tradeoffs are nonexistent at this point), then the imposition of the sequential search constraint can proceed if the rate-distortion performance loss due to the isolated product code constraint is acceptably small, and hopefully, the imposition of the additional sequential search constraint does not result in a fatal reduction of the rate-distortion performance of the structured vector quantizer.

Ideally, in the second part of this investigative exercise, one would hope to employ "optimal" sequential search rules that are derived from the defining properties of the sequential search structural constraint. However, analytic difficulties often arise when seeking necessary conditions for the optimality of VQ's subject to both product code and sequential search constraints. In most cases, this optimization problem cannot be formulated to permit a tractable analysis. An alternative approach that is often used is to find a more amenable definition of sequential search "optimality" that is not directly related to the sequential search structural constraint. An "optimal" sequential search encoder is usually defined as one that implements a partition of the VQ input space that is equivalent to the partition generated by an exhaustive search encoder. In some cases, this approach has led to a tractable analysis, where sequential search encoders have been identified that are equivalent to exhaustive search encoders. Such "optimal" sequential search encoders have been identified for gain-shape VQ [5], mean-gain-shape VQ [6], and RVQ's [8], but alas, in the latter case, these optimal, sequential search RVQ encoders are most often, but not always, also equivalent to exhaustive search encoders in complexity. Since there is no guarantee that sequential search rules that collectively implement exhaustive search partitions will have efficient realizations, the code designer must often resort to other design alternatives to achieve the goal of simultaneous good performance and low cost.

One alternative is the imposition of additional structure on the product code to make the code more submissive to sequential searches. Multiple-stage VQ's with stage codebooks comprised of lattice VQ's [14], [15] and direct sum codebooks subject to symmetry constraints are examples of this option [16], [17]. Another option is to accept relatively small increases in sequential search encoder complexity to permit better approximations of exhaustive search encoding decisions. Multiple path search RVQ's are an example of this latter option. In fact, prior research [8] has shown that for some information sources the performance penalty associated with the simultaneous imposition of a direct sum product code constraint and a multiple path sequential search constraint on a vector quantizer is acceptably small. Thus, with careful design, the structural constraints of these RVQ's are sufficiently nonrestrictive that performances superior to that of same-cost ESVQ's can be obtained.

B. RVQ's with Many Stages

In this paper, the problem of effective RVQ design is revisited while paying particular attention to the sequential search constraint of RVQ and to RVQ's with many stages. RVQ's with many stages are important in applications where the amount of available training data is limited (limited training data necessitates the use of small stage codebook sizes), applications that require large vector sizes, and applications where RVQ stage codebooks are transmitted as overhead information. Image-adapted RVQ's [18] and nearest-neighbor data classifiers that use RVQ pattern templates [19] are examples where training data is limited. Applications requiring large vectors sizes include variable dimension VQ [20] and transform VQ [21]. Multispectral imagery and video coding applications also benefit from the use of large vector sizes; with multispectral imagery, RVQ's with large vector sizes can jointly encode both spatial and spectral information [22], and with video data, such RVQ's can jointly encode both spatial and temporal information [23]. Furthermore, RVQ's with many stages are of interest in their own right since stage codebook sizes of two or four codevectors minimize the memory and computation costs of RVQ (for a given rate and dimension).

The design of RVQ's with many stages has not been previously given much attention in the literature. Nearly all RVQ literature considers the two-stage case and then inductively reasons that two-stage results can be generalized to RVQ's with many stages. Although reasonable, there are problems that arise when this approach is adopted; design methods developed for two-stage RVQ's may not be practical nor have satisfactory generalizations to many-stage RVQ's due to unforeseen difficulties. These difficulties include loss of design stability, catastrophic design failure, and probably the most important problem of all (for RVQ's with high rates or large vector sizes), substandard performance for an increased number of stages (with rate and dimension held fixed). An alternative research approach, which is used in this paper, is to initially focus on the design of RVQ's with large number of stages and, then, when necessary, specialize the results to RVQ's with only a few stages. Solutions to the design problems of RVQ's with many stages are sure to be valid for RVQ's with only two stages since many of the challenges of RVQ design are not as evident nor as problematic in the two-stage case. Specifically, this paper presents an improved, nongreedy, joint design method for RVQ that is very stable and provides RVQ codebooks with good performance, even when many stages are generated or the amount of training data is extremely limited.

Although there is a rich theoretical basis for many of the results of this paper, due to the difficulties involved with

designing sequential search product codes (as explained in this and the next two sections), it is currently necessary to rely on (justifiable) heuristics for certain parts of the sequential search RVQ encoder design process presented in this paper. The introduction of heuristics into any process makes control of the process less of a science and more of an art; the design of effective and efficient RVQ's is no exception.

This paper also presents a new design method for a successive approximation RVQ that uses different numbers of stages on different input vectors. This latter design process generates a sequence of "significance thresholds" as a byproduct of the stage codebook designs that are shown to be effective in subband coding schemes such as embedded wavelet coding systems [24] and in pixel-based coding systems that allocate rate on the basis of intrablock pixel variance. In addition to these new design methods, an alternative RVQ codebook design approach [25], [26] that is based on a multilayer competitive neural network that minimizes the causal error of a sequential search encoder while subject to a constraint on the error due to subsequent stages is also presented and discussed.

This paper has a three-fold purpose. First, it provides a tutorial description of RVQ. Second, it presents new advances in the design and application of RVQ. Third, it provides a survey of RVQ research results and various hybrid-RVQ compression schemes that have appeared in the literature. Although an attempt has been made to be exhaustive in our presentation of recent advances in residual vector quantization, any overt omissions that may have occurred are due to unintentional oversight. The reader is also cautioned that many of the topics given emphasis in this paper relate more to the realm of experience of the authors and is not always intended to be indicative of relative research or application value.

This paper is outlined as follows. Section II provides an overview of RVQ and is basically an extension of this introductory section. Sequential and joint design methods that were initially proposed in the literature are reviewed. Issues related to the merger of sequential search procedures and direct sum codebook constraints are elaborated. Section III presents new and improved design methods for RVQ's after illustrating the effect of RVQ codebook entanglement and reviewing various entanglement-permissive RVQ encoders. Section IV discusses advances in the area of variable rate RVQ's. Entropy coded, entropy constrained, and conditional entropy coded/constrained RVQ's are visited. Section V surveys advances in the area of predictive and finite state RVQ's, including the use of neural net state predictors. Section VI reviews advances in the use of RVQ's as successive approximation codes, including the use of direct sum structures as successive approximation quantizers in image waveform and subband coding systems. A new type of successive approximation RVQ structure that uses different numbers of stages on different input vectors is introduced. Section VIII concludes the paper with a summary statement.

II. Residual Vector Quantizers

The structure and operation of a residual vector quantizer are described. Joint optimality is defined and previously proposed sequential and joint design methods are revisited. The approximation process often used for RVQ encoder design is exposed and elaborated.

A. Defining Structural Constraints of RVQ

A residual vector quantizer is defined by the imposition of the following three structural constraints:

1) the encoder direct sum codebook constraint
2) the encoder sequential search constraint
3) the decoder direct sum codebook constraint.

The purpose of the direct sum codebook constraint (imposed at both the encoder and decoder) is to reduce memory requirements. The purpose of the sequential search constraint is to reduce computation requirements. The sequential search constraint determines a permutation of the encoder stage codebooks that may, or may not, be in effect at the decoder. Applications that require progressive transmission or successive approximation, for example, access decoder codebooks in the same order used by the sequential search encoder. Other applications access decoder stage codebooks simultaneously. Block diagrams of the RVQ encoder and decoder are shown in Fig. 1.

If an RVQ consists of P stages with N_p codevectors in the pth stage codebook, then the memory and computation costs of RVQ are proportional to ΣN_p. If the stage codebook sizes are identical, then $N_p = 2^{kr/P}$, and the costs are proportional to $P \times 2^{kr/P}$. Although costs go down as the number of stages goes up, early empirical evidence suggested that RVQ's with many stages deliver disappointing performances [1], [11], [27]. Poor performance results have proven, at least for some sources, to be more a result of the design method used to generate stage codebooks and not necessarily an unavoidable consequence of the RVQ structural constraints [8], [28]. The following section describes the sequential method first used to design RVQ codebooks.

B. Sequential Design Methods

Juang and Gray [9] first proposed the RVQ structure and suggested that RVQ stages be designed by sequential application of the generalized Lloyd algorithm (GLA) [29], [30]. Although sequential use of the GLA is nearly optimum for two-stage RVQ's with moderated to high output rates (i.e., large stage codebook sizes) [31], [32], this design method is increasingly unsatisfactory as the number of stages grows beyond two [11].

The main shortcoming of the sequential GLA design method is that each stage codebook is generated while considering only the error due to previous stages (the *causal* error); the error due to subsequent stages (the *anticausal* error) is ignored. A joint design approach, on the other hand, takes into account both the causal and anticausal errors to reduce the *overall* error. Various techniques have been proposed for joint optimization, but before these techniques are described in the following sections, let us introduce mathematical notation and define exactly what is meant by "joint optimality."

The pth stage of an RVQ is a k-dimensional vector quantizer defined by the mapping $\boldsymbol{Q}_p: \mathbb{R}^k \mapsto \boldsymbol{C}_p$, where

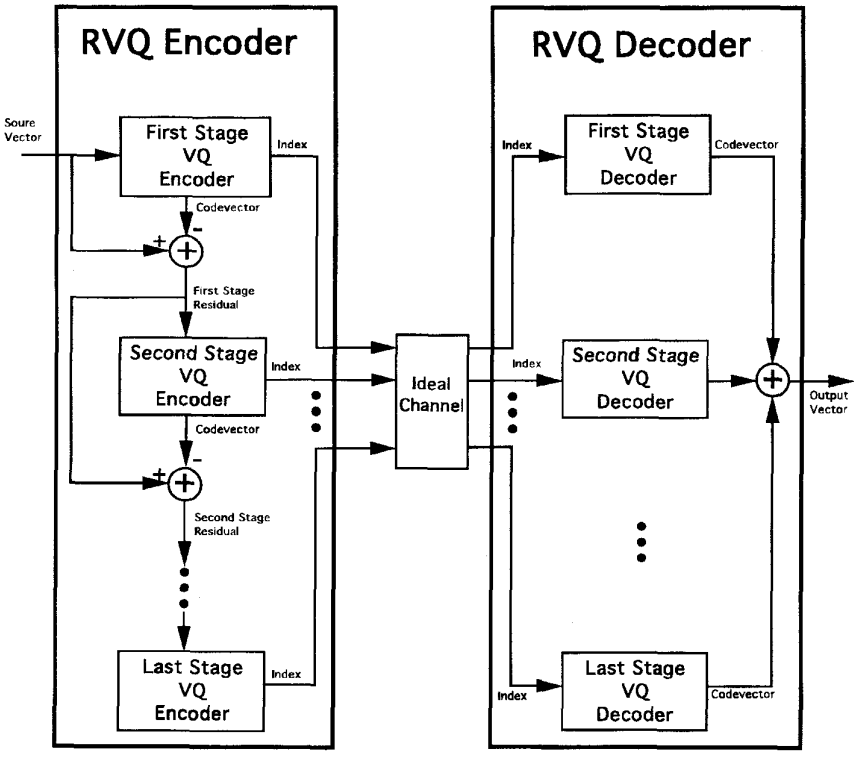


Fig. 1. RVQ block diagram.

$\mathbb{R}^k$ is the k-dimensional input space of each RVQ encoder stage, $p \in \{1, 2, \cdots, P\}$ is a *stage index*, $\boldsymbol{C}_p = \{\boldsymbol{y}_p(0), \boldsymbol{y}_p(1), \cdots, \boldsymbol{y}_p(N_p - 1)\}$ is the pth-stage *codebook*, $\boldsymbol{y}_p(i_p) \in \mathbb{R}^k$ is a pth-stage *codevector*, and i_p is an *index* in the pth-stage *index set* $\boldsymbol{I}_p = \{0, 1, \cdots, N_p - 1\}$. Residual quantizer stage mappings $\boldsymbol{Q}_p$ are collectively equivalent to a single mapping $\mathbf{Q}: \mathbb{R}^k \mapsto \mathbf{C}$, where $\mathbf{C}$ is the *direct sum codebook* $\{\mathbf{y}(\mathbf{i}): \mathbf{i} \in \mathbf{I}\}$, $\mathbf{y}(\mathbf{i})$ is a *direct sum codevector* $\mathbf{y}(\mathbf{i}) = \boldsymbol{y}_1(i_1) + \boldsymbol{y}_2(i_2) + \cdots + \boldsymbol{y}_P(i_P)$, and $\mathbf{i} = (i_1 i_2 \cdots i_P)$ is a *P-tuple index* contained in $\mathbf{I} = \boldsymbol{I}_1 \times \boldsymbol{I}_2 \times \cdots \times \boldsymbol{I}_P$.

The direct sum codebook is the direct sum of the stage codebooks $\mathbf{C} = \boldsymbol{C}_1 + \boldsymbol{C}_2 + \cdots + \boldsymbol{C}_P$ (if the operands of "+" are sets, then "+" denotes a direct sum operation, otherwise the usual summation operator is indicated). In practice, the stage mappings $\boldsymbol{Q}_p(\cdot)$ are realized as a composition of a stage encoder mapping $\boldsymbol{E}_p: \mathbb{R}^k \mapsto \boldsymbol{I}_p$ and a stage decoder mapping $\boldsymbol{D}_p: \boldsymbol{I}_p \mapsto \boldsymbol{C}_p$, that is, $\boldsymbol{Q}_p(\boldsymbol{x}_1) = \boldsymbol{D}_p[\boldsymbol{E}_p(\boldsymbol{x}_1)]$, where $\boldsymbol{x}_1$ is a realization of $\boldsymbol{X}_1$, which is the random source output. The direct sum quantizer also has a compositional realization $\boldsymbol{Q}(\boldsymbol{x}_1) = \mathbf{D}[\mathbf{E}(\boldsymbol{x}_1)]$. Although it is an abuse of notation, we find it useful in subsequent discussions to represent the direct sum encoder by $\mathbf{E} = \boldsymbol{E}_1 + \boldsymbol{E}_2 + \cdots + \boldsymbol{E}_P$ and to represent the direct sum decoder by a similar expression.

C. What is "Joint Optimality?"

A residual vector quantizer is said to be jointly optimal if a local or global minimum value of the average distortion $\mathcal{D}(\mathbf{E}, \mathbf{D}) = E\{d[\boldsymbol{X}_1, \mathbf{D}(\mathbf{E}(\boldsymbol{X}_1))]\}$ is achieved, where $d(\cdot, \cdot)$ is a distortion metric, and $E\{\cdot\}$ is the expectation operator. A distortion minimum $\mathcal{D}^*$ has been obtained when slight adjustments in either the encoder $\mathbf{E}^*$ or decoder $\mathbf{D}^*$ only increase distortion. An RVQ decoder $\mathbf{D}^*$ is said to be jointly optimal if for an arbitrarily fixed encoder $\mathbf{E}$, slight adjustments of $\mathbf{D}^*$ only increase distortion. An RVQ encoder $\mathbf{E}^*$ is said to be jointly optimal if for an arbitrarily fixed decoder $\mathbf{D}$, slight adjustments of $\mathbf{E}^*$ only increase the distortion.

The joint optimization problem is to find an encoder $\mathbf{E}^* = \boldsymbol{E}_1^* + \cdots + \boldsymbol{E}_P^*$ and a decoder $\mathbf{D}^* = \boldsymbol{D}_1^* + \cdots + \boldsymbol{D}_P^*$ that achieve a local distortion minimum. Ideally, it is desirable to have a design process that allows determination of the jointly optimal direct sum encoder $\mathbf{E}^*$ and the jointly optimal direct sum decoder $\mathbf{D}^*$ by first finding a sequence of stage encoders $\boldsymbol{E}_p^*$ and a set of stage decoders $\boldsymbol{D}_p^*$ that achieve $\mathcal{D}^*$. This is currently possible only for the decoder. Research has shown that an optimal direct sum decoder codebook $\mathbf{C}^*$ can be constructed by suitable design of the stage decoder codebooks $\boldsymbol{C}_p^*$ [28], [33]. The stage codebook design rules follow directly from a minimization of the overall error with respect to the stage codevectors (see Section II-D). These minimizations lead to conditions the decoder stage codevectors must satisfy for joint optimality. Joint optimization of the encoder is obtained in a converse manner: an optimal direct sum encoder $\mathbf{E}^*$ is identified first, and then, a sequence of "jointly optimal" stage encoders $\boldsymbol{E}_p^*$ are sought that are collectively equivalent to $\mathbf{E}^*$. Unlike the decoder case, there is currently no known method for minimizing the overall error of the RVQ with respect to the stage parameters of a sequential search encoder. What is known is that if a direct sum encoder is defined as optimal when it implements a Voronoi partition with respect

to a given direct sum codebook $\mathbf{C}$, then stage partitioning rules that define a sequence of $\boldsymbol{E}_p^*$ that are collectively equivalent to an exhaustive search Voronoi encoder $\mathbf{E}_V^*$ are known [8], [28] (the "V" subscript indicates that the direct sum encoder implements a Voronoi partition with respect to the direct sum codebook $\mathbf{C}$). Although this tack results in a tractable analysis, the stage encoders that follow from this joint design approach are often computationally expensive, and a performance-complexity compromise is necessary to control implementation cost. This compromise usually entails finding a sequence of encoder stages that sufficiently approximates $\mathbf{E}_V^*$ at a reasonable cost, i.e., a sequence of $\boldsymbol{E}_p$ are sought after such that $\boldsymbol{E}_1 + \cdots + \boldsymbol{E}_P \approx \mathbf{E}_V^*$.

D. Necessary Conditions for Joint Decoder Optimality

As explained in Section II-C, joint decoder optimality is obtainable by designing each stage codebook to achieve the desired overall result. The design rules follow from a minimization of the overall error with respect to the stage codevectors. For example, consider the case of a residual scalar quantizer (RSQ). Necessary conditions that the stage output values $\boldsymbol{y}_p(i_p)$ must satisfy for (alphabet-constrained) joint optimality follow from the requirement

$$\frac{\partial \mathcal{D}}{\partial \boldsymbol{y}_p(i_p)} = 0. \tag{1}$$

This condition is satisfied when the stage quanta $\boldsymbol{y}_p(i_p)$ are centroids of residuals formed from the encoding decisions of stages both prior and subsequent to the pth stage, i.e., the *causal–anticausal residual* [8], [28], [33]. Extending this result to higher dimensional spaces leads to a generalized centroid rule, and just like single stage VQ design methods [30], decoder optimality can be achieved by replacing old stage codevectors with updated (residual) centroids of repartitioned training set cells.

E. Necessary Conditions for Joint Encoder Optimality

Unlike the approach taken with the RVQ decoder, where stage decoders $\boldsymbol{D}_p^*$ are individually determined that realize a jointly optimal direct sum decoder $\mathbf{D}^*$, the RVQ encoder design problem starts by finding an optimal (unconstrained) direct sum encoder $\mathbf{E}_V^*$ for a given direct sum codebook $\mathbf{C}$ and then seeks a sequence of $\boldsymbol{E}_p^*$ that are collectively equivalent to $\mathbf{E}_V^*$. The $\boldsymbol{E}_p^*$ are then examined to see if they have an efficient implementation; .if not, then it is necessary to seek a sequence of suboptimal (but efficient) $\boldsymbol{E}_p$ that sufficiently approximate $\boldsymbol{E}_p^*$ and that, in turn, collectively provide a satisfactory approximation of $\mathbf{E}_V^*$.

If the direct sum codebook is fixed, then the optimal direct sum mapping and corresponding direct sum partition are defined by a nearest-neighbor rule

$$\mathbf{Q}(\boldsymbol{x}_1) = \mathbf{y}(\mathbf{i}) \Leftrightarrow d[\boldsymbol{x}_1, \mathbf{y}(\mathbf{i})] \le d(\boldsymbol{x}_1, \mathbf{C}) \tag{2}$$

where the distance $d(\cdot, \cdot)$ between the source output $\boldsymbol{x}_1$ and the set $\mathbf{C}$ is defined in the usual way. A sequential search encoder that induces a direct sum partition that satisfies (2) is said to be jointly optimal and can be described by the following sequence of stage encoding rules:

$$\begin{aligned} \boldsymbol{Q}_p(\boldsymbol{x}_p) = \boldsymbol{y}_p(i_p) \Leftrightarrow d[\boldsymbol{x}_p, \boldsymbol{y}_p(i_p) + \boldsymbol{C}_{p+1} + \cdots + \boldsymbol{C}_P] \\ \le d(\boldsymbol{x}_p, \boldsymbol{C}_p + \boldsymbol{C}_{p+1} + \cdots + \boldsymbol{C}_P) \end{aligned} \tag{3}$$

for $p = 1, 2, \cdots, P$, and where

$$\boldsymbol{x}_p = \begin{cases} \boldsymbol{x}_1, & \text{if} \quad p = 1, \\ \boldsymbol{x}_1 - \sum_{j=1}^{p-1} \boldsymbol{Q}_j(\boldsymbol{x}_j), & \text{otherwise} \end{cases} \tag{4}$$

is the pth-stage causal residual. Using the language of group theory [34], the pth-stage partitioning rule (3), which defines $\boldsymbol{E}_p^*$, requires that all $\boldsymbol{x}_p$ that are mapped to $\boldsymbol{y}_p(i_p)$ be at least as close to a point in the coset $\boldsymbol{y}_p(i_p) + \boldsymbol{C}_{p+1} + \cdots + \boldsymbol{C}_P$ as to any point in the cosets $\boldsymbol{y}_p(k_p) + \boldsymbol{C}_{p+1} + \cdots + \boldsymbol{C}_P$ with $k_p = 0, 1, \cdots, N_p - 1$ and $k_p \neq i_p$. Using the language of trees [8], the optimal (sequential search) pth-stage decision region corresponding to $\boldsymbol{y}_p(i_p)$ is a Voronoi region with respect to the leaf nodes of the subtree $\boldsymbol{y}_p(i_p) + \boldsymbol{C}_{p+1} + \cdots + \boldsymbol{C}_P$ contained within the tree $\mathbf{0} + \boldsymbol{C}_p + \boldsymbol{C}_{p+1} + \cdots + \boldsymbol{C}_P$, where $\mathbf{0}$ is the origin of $\mathbb{R}^k$.

Although (3) provides a sequence of stage encoding rules where the induced direct sum partition is a Voronoi partition with respect to the direct sum codebook, it does not necessarily follow that the resulting optimal sequential search encoder is computationally efficient. Application of these stage encoding rules generally results in a very complex sequential search encoder because optimal stage partition cells, or more appropriately, optimal stage equivalence classes are not necessarily convex nor connected. The pth-stage equivalence classes defined by (3) are unions of certain translated direct sum Voronoi partition cells (these cells are translated by the stage codevectors prior to the pth stage used to form the causal residual input vectors to the pth stage). These unions often form disjoint sets that make optimal sequential search encoding with standard nearest-neighbor search procedures impossible. The structural complexity of these unions can be related to characteristics of the tree structure of the RVQ encoder. Unlike most quantizer tree structures, RVQ tree structures may possess a property called *entanglement.* If an RVQ encoder tree is *unentangled,* then an optimal sequential search RVQ encoder has an efficient implementation. If an RVQ encoder tree is *entangled,* then implementation of an optimal sequential search encoder is more complex. More is said about entanglement in Section III-C.

F. Joint Design Methods

Joint design methods were developed by Barnes and Frost [8], [28], [33] and Chan and Gersho [7], [33], [36]. The two design approaches are similar in that both approaches are built upon the fundamental source code design paradigm [37] that is employed by the GLA: Given training data and initial codebooks, the codebooks are alternately updated to iteratively improve the source code's performance until a convergence criterion is satisfied. The differences in the design processes of Chan and Gersho and those of Barnes and Frost relate

to the number of fixed-point procedures and to the sequence of steps used to alternately update the encoder and decoder stage codebooks. Chan and Gersho also developed a "joint-update" method that can be applied to all decoder stage codebooks simultaneously. The following sections give a high level description of these joint design techniques and the basic principles of iterative source code design procedures on which these methods are based.

1) Iterative Source Code Design Principles: A principle commonly used to design encoder/decoder pairs is to iteratively find an optimal encoder for a fixed decoder, and vice versa, find an optimal decoder for a fixed encoder [37]–[39]. Use of this method requires necessary conditions for the optimality of the encoder and decoder when the other is held fixed. Let $\mathcal{O}_E$ represent a rule that is used to optimize the encoder when the decoder is held fixed, and let $\mathcal{O}_D$ represent a rule that is used to optimize the decoder when the encoder is held fixed. A general design procedure used for generating source encoder/decoder pairs that satisfy conditions necessary for optimality is as follows:

STEP 1: Select and hold fixed an initial decoder.

STEP 2: For the fixed decoder, use $\mathcal{O}_E$ to select an optimal encoder.

STEP 3: For the fixed encoder, use $\mathcal{O}_D$ to select an optimal decoder.

STEP 4: Compute the average distortion of the resulting code. If the average distortion falls below some predetermined value or if the relative change in average distortion falls below some predetermined threshold, then STOP; otherwise, continue with STEP 2.

Steps 2 and 3 of the iterative design procedure only reduce or leave unchanged the average distortion. Since the distortion is bounded below by zero, the monotonically nonincreasing sequence of distortion measures converges to a fixed point [40]. The next sections review the various techniques that have been used to implement these four steps of the iterative design process and describes ways that this iterative process has been adapted to accommodate the multiple stage structure of RVQ's.

2) Sequential-Update Method: The sequential-update, joint optimization method of Chan and Gersho [7], [33], [36], [41] is a generalization of the Sabin and Gray approach used to design shape/gain product code VQ's [5]. As shown in Fig. 2, the algorithm iteratively alternates between updating the encoder and the decoder, where only one stage of the decoder is updated between encoder updates. The encoder update step uses the improved decoder stage as a new encoder stage in a sequential search of the encoder stage codebooks. Once this training set repartitioning step is completed, the codevectors in a different decoder stage are updated by replacing old codevectors with new causal–anticausal residual centroids.

A shortcoming of the sequential-update method is that simultaneous satisfaction of the decoder optimality conditions by all decoder stages is only achieved asymptotically as the number of joint encoder-decoder design iterations become large. The decoder update step returns a decoder where only one stage satisfies conditions necessary for joint optimality, i.e., the RVQ decoder is such that $\mathbf{D} = \boldsymbol{D}_1 + \cdots + \boldsymbol{D}_{p-1} + \boldsymbol{D}_p^* + \boldsymbol{D}_{p+1} + \cdots + \boldsymbol{D}_P$. Thus, when the design process proceeds to the encoder update step, it cannot be said that the encoder uses a set of jointly optimal direct sum codebooks to repartition the training data (since only one stage has been improved). Only in the limit that a fixed point of the overall design process is approached is "joint optimality" obtained (assuming the encoder is sufficiently accurate to allow convergence to occur).

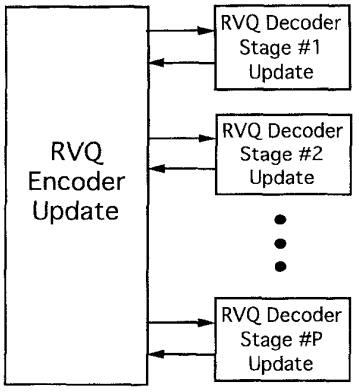


Fig. 2. Chan and Gersho sequential-update method for joint RVQ design.

3) Joint-Update Method: The joint-update, joint optimization method of Chan and Gersho [36], [42] is also based on the GLA paradigm (see Fig. 3), but unlike the sequential-update method, the joint-update method permits all RVQ decoder stages to be updated simultaneously before returning to the encoder improvement step. The objective of the joint-update method is to seek a set of stage codebooks whose equivalent product codebook best approximates a "reference product codebook." The reference codebook utilized is not really a product code but is the set of codevectors that is obtained if a structurally unconstrained codevector is used to represent the training vectors in each partition cell induced by the sequential search encoder. This "reference codebook" is equivalent to the codebook that would be used by the unconstrained decoder of a TSVQ (i.e., by a multiple-stage VQ with complete codebook fanout). This joint-update design method is based on a formulation of the "excess distortion" between the distortion of the RVQ direct sum codebook and this unstructured reference codebook [3]. The best RVQ codebook (i.e., one that minimizes this excess distortion) is found by solving a weighted, underdetermined, linear least-squares system of unknowns. To guarantee a unique solution, it is necessary to hold at least one stage codevector fixed during the decoder update step. More than one stage codevector may be held fixed to allow a complexity tradeoff during codebook design. If the stage codevectors of all but one stage are held fixed, Chan and Gersho maintain that this joint-update method embodies their sequential-update method [3]. A shortcoming of the joint-update method is that it becomes overly complex for large direct sum alphabet sizes since a system of equations that has as many unknowns as there are possible RVQ outputs must be solved.

4) Interlaced Fixed Point Method: Barnes and Frost also used an iterated process to design jointly optimal RVQ's [8]. The difference, however, between the Barnes and Frost

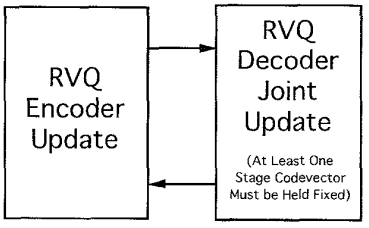


Fig. 3. Chan and Gersho joint-update method for joint RVQ design.

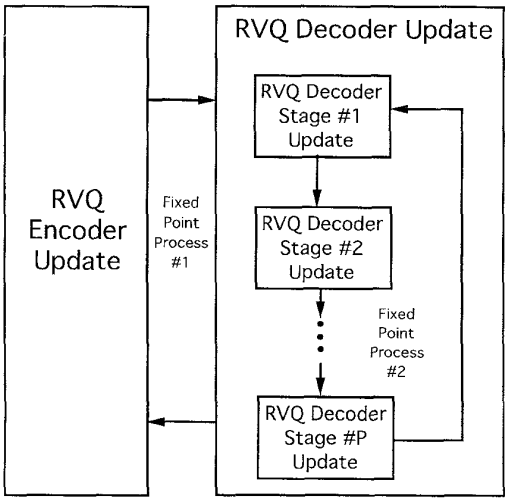


Fig. 4. Barnes and Frost interlaced fixed-point method for joint RVQ design.

algorithms and the Chan and Gersho algorithms is that the former uses two interlaced fixed point procedures to achieve joint optimality. One fixed point process (called the outer process) is used for optimization of the encoder/decoder pair. Another fixed point process (called the inner process) is used for joint optimization of all decoder stage codebooks. In the inner fixed point process, each stage codebook is iteratively updated while holding the codebooks of all other stages fixed. The new codevectors of the updated stage satisfy the necessary causal–anticausal residual centroid condition with respect to the fixed partition and the fixed codebooks of the other stages. The codevector update procedure is then repeated for a different stage (without performing a new encoder update), but since the process of optimizing the codevectors of a different stage causes the first stage that was optimized to no longer satisfy the residual centroid condition (this is also true for the sequential-update method), it is necessary to eventually return to each decoder stage and repeat the process in round-robin fashion to asymptotically reach the point where all stages simultaneously satisfy the optimality conditions (see Fig. 4) [8], [43], [44]. After this decoder-only fixed point has been reached (or approached sufficiently close), a new encoder/decoder iteration of the outer process is performed (a new partition is selected), and then, the entire process is repeated.

A problem of the interlaced fixed-point design method is its precarious stability. Since RVQ encoders derived from updated decoder codebooks often only approximate optimal

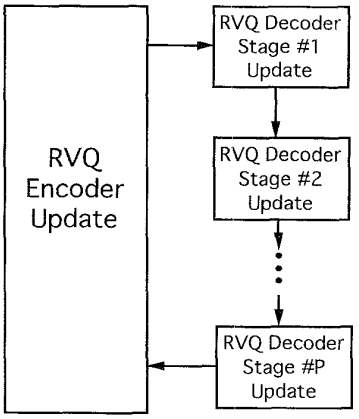


Fig. 5. Alternative joint design process for joint codebook design.

encoding rules, there is no guarantee that the encoder update step will improve performance. This is particularly true if the stage codebooks change substantially between encoder update steps. Since only one stage is updated between encoder updates in the sequential-update method, it is less likely that the restricted set of codebook updates of that method (restricted to one stage) will cause drastic decreases in encoder performance; hence, the sequential-update method tends to be fairly stable. In the interlaced fixed point method, since the codebooks are repeatedly adjusted during the decoder update step, it is more likely that the codebook changes will be substantial, and hence, the resulting decoder codebooks often have different (not necessarily better) sequential search performance characteristics. On the other hand, the interlaced fixed-point method does achieve a greater degree of joint optimization at all decoder stages between encoder update steps.

5) Alternative Sequential-Update Method: An alternative process flow that can be used for joint design of direct sum codebooks is shown in Fig. 5. Instead of forcing the decoder optimization process to a fixed point before returning to the encoder update step, each decoder stage can be updated just one time before returning to the encoder optimization step. This approach results in a simpler, less aggressive, and more stable design process than the interlaced fixed-point method but is more aggressive in seeking joint optimality than the sequential-update method. However, as defined in this paper, joint optimality requires that all stages simultaneously satisfy the necessary conditions, and if the decoder stage updates are not repeated over all stages until a fixed point is reached or sufficiently approached, then joint optimality is not achieved between encoder updates.

Another tempting simplification of a joint design process is to update all stages simultaneously where the old codevectors in each stage are replaced with new residual centroids without accounting for previous codevector changes made in other decoder stages. This method requires less design operations than the approach where all prior codevector updates are accounted for when forming the causal–anticausal residual space of each stage. Unfortunately, this decoder update approach is

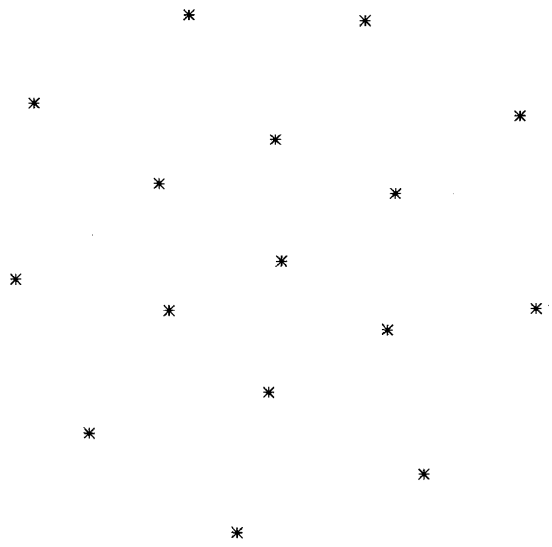


Fig. 6. Codevector constellation of single-stage ESVQ with 16 codevectors. SNR = 9.67 dB.

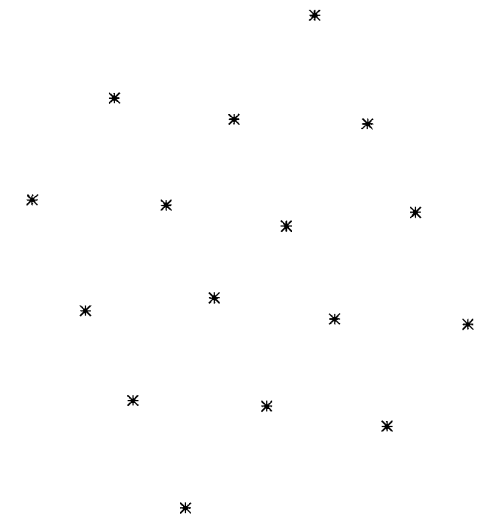


Fig. 7. Codevector constellation of an exhaustive search two-stage RVQ with four codevectors per stage. SNR = 9.42 dB.

not convergent under a mean squared error criterion and is not recommended [44].

III. Improved RVQ Design Methods

Principles used to develop improved design methods for RVQ's with suboptimal but computationally efficient sequential search encoders are given, but before the improved design methods are presented, motivational material on direct sum codebook performance potential and RVQ tree structure entanglement is presented.

A. Jointly Optimal Product Codes: An Upper Bound

The performance limitations of the direct sum codebook constraint can be assessed by comparing jointly optimal, direct sum product codes with optimal, unconstrained, single-stage VQ's. Consider the unconstrained ESVQ codebook shown in Fig. 6 designed with the GLA for the memoryless Gaussian source. The codebook consists of 16 codevectors of dimensionality 2 to give an output rate of 2 bits per sample (b/sample). The average signal-to-noise ratio (SNR) of the codebook is 9.67 dB (using mean squared error). This codebook constellation has the desirable characteristics of a denser codevector packing in the more probable center region of the 2-D probability density function (pdf) and a sparser packing in the skirt or tail region of the pdf. In addition, note that the constellation is approximately hexagonal in the center region. A hexagonal constellation has optimal covering properties in 2-D spaces [45]. The performance of this unconstrained, single-stage VQ is next compared with that of two different jointly optimal, direct sum product code structures that employ exhaustive search encoders.

Consider two direct sum product codes with the first having two stages and four codevectors per stage, and the second having four stages and two codevectors per stage. Both product codes have 16 possible output codevector combinations and hence can be compared to the ESVQ of Fig. 6. The jointly optimal direct sum codebook sets of these two product codes were generated with the use of an exhaustive search encoder during the interlaced fixed point design method of Section II-F-4 and are shown, respectively, in Figs. 7 and 8. Although conventional wisdom would say the two-stage quantizer is structurally less constrained than the four-stage quantizer, in this instance, the four-stage quantizer provides better performance (SNR = 9.55 dB) than the two-stage quantizer (SNR = 9.42 dB). Compared with the SNR = 9.67 dB of the ESVQ, the jointly optimal two-stage product code suffers a SNR loss of 0.25 dB, whereas the four-stage code looses only 0.12 dB. Examining the constellations of Figs. 7 and 8, we see that although the two-stage constellation of Fig. 7 is basically hexagonal, it has a codevector density that is rather uniform. This lack of a varying codevector density contributes to the 0.25-dB performance loss. The four-stage constellation of Fig. 8 has both a varying codevector density and a hexagonal characteristic that yields a performance level closer to that of the single stage VQ. Prior results [8] have shown that for some information sources, two-codevector per stage direct sum codebooks can be just as effective as two-stage direct sum codebooks for implementing the same direct sum codebook size. In these cases, jointly optimal (exhaustive search) direct sum product code performance is nearly independent of the number of stages. Since RVQ costs go down as the number of stages goes up (for a fixed rate and dimension), these empirical results suggest that effective design methods for sequential search RVQ's that have small stage codebook sizes may result in profitable returns. In general, the performances of jointly optimal direct sum product codes provide empirical upper bounds to the obtainable performance of RVQ's that have well designed, but suboptimal, sequential search encoders.

B. Greedy Sequential Search Product Codes: A Lower Bound

Experiments with exhaustive search direct sum product codes, as demonstrated in Section III-A, give encouraging

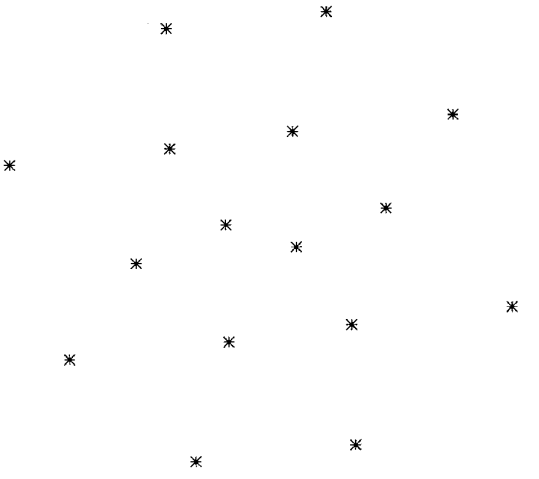


Fig. 8. Codevector constellation of an exhaustive search four-stage RVQ with two codevectors per stage. SNR = 9.55 dB.

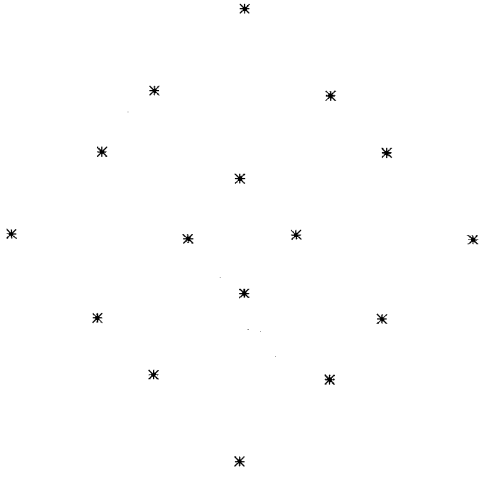


Fig. 9. Codevector constellation of a sequential search four-stage RVQ with two codevectors per stage designed with sequential use of the GLA. SNR = 8.84 dB.

results. Experiments with sequential search direct sum product codes, as we show in this section, give discouraging results when designed with sequential use of the GLA. For example, the codevector constellation of a four-stage, two codevectors per stage RVQ designed with sequential use of the GLA is shown in Fig. 9. Unlike the corresponding jointly optimal constellation of Fig. 8, this constellation has neither a variable codevector density nor a hexagonal characteristic. In fact, this constellation, with an SNR of 8.84 dB, does not even perform as well as the pdf-optimized uniform scalar quantizer, which yields a SNR of 9.25 dB [46].

Furthermore, consider the situation where a high-rate RVQ is desired. If a high-rate RVQ is obtained by adding stages instead of increasing the stage codebook sizes, and if sequential application of the GLA is used to design the stage codebooks, then experience has shown that such RVQ's are extremely suboptimal. Consider the example shown in Fig. 10. This graph shows the SNR versus number-of-stages performance of an

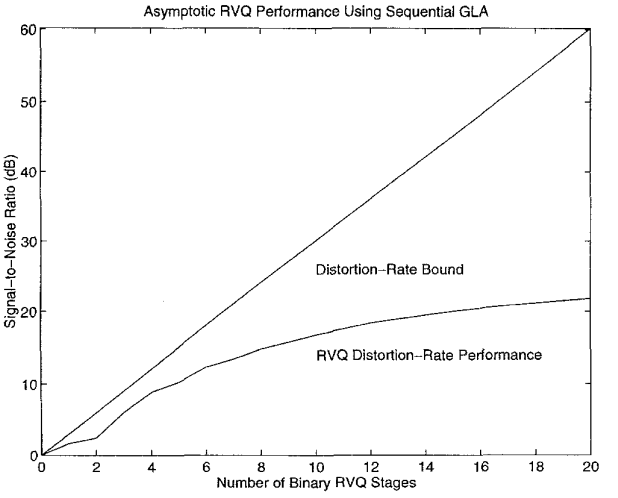


Fig. 10. SNR performance of RVQ's with binary stages as a function of the number of stages designed with sequential use of the GLA.

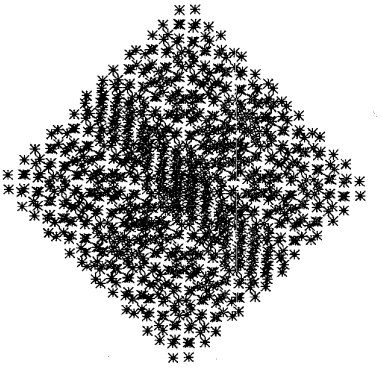


Fig. 11. Codevector constellation of a 10-stage RVQ with two codevectors per stage designed with greedy sequential use of the GLA. SNR = 16.74 dB.

RVQ when binary (two codevector) stages are generated and added to the RVQ with sequential use of the GLA. In this case, the 20-stage binary RVQ returns an SNR performance that is more than 38 dB below the distortion-rate bound of the memoryless Gaussian source. If we examine the equivalent direct sum codebook of the 10-stage RVQ shown in Fig. 11, for example, it is clear why this type of RVQ gives such poor performance. This direct sum codebook has codevectors that are nearly coincident, has rather compacted (poorly dispersed) codevector placements, has little variation in the codevector density, and has no hexagonal properties that exploit the geometric coding gain available in two dimensions. All of these undesirable attributes suggests that the performances of these RVQ's can be improved with better design procedures that mitigate these undesirable characteristics. Indeed, the performances of these RVQ's should provide an empirical lower bound to the performance of any so called "jointly optimal RVQ" designed with improved design methods.

C. Entanglement: The Nemesis of Sequential Search Product Codes

Although we have gained insight into reasons for the poor performance of RVQ's designed with sequential use of the

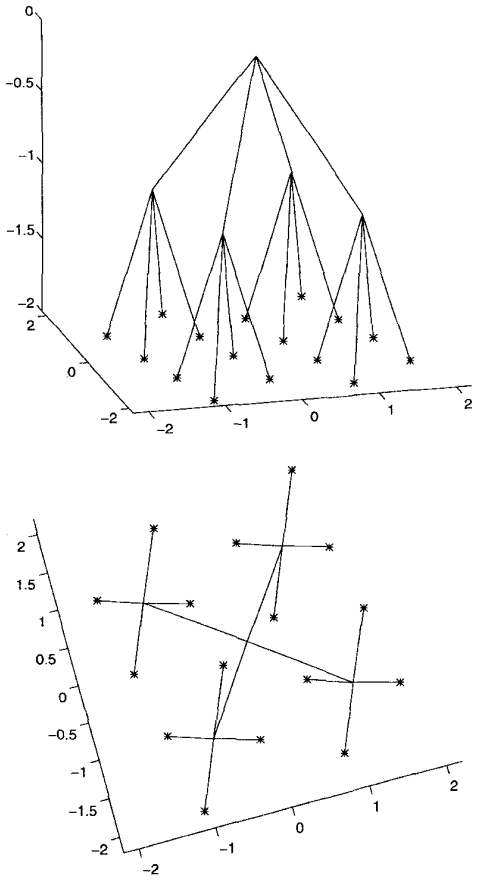


Fig. 12. Tree structure of a two-stage RVQ with four codevectors per stage.

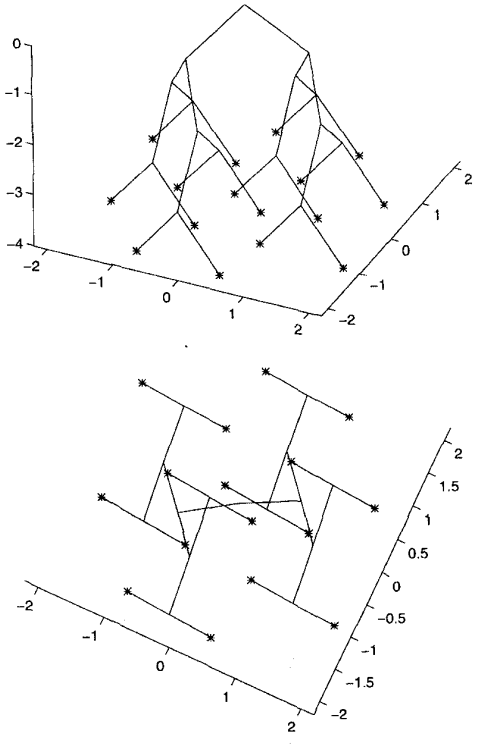


Fig. 13. Tree structure of a four-stage RVQ with two codevectors per stage.

GLA, we have yet no insight into the relative severity of a sequential search structural constraint when it is applied to jointly optimal product codes. This section addresses this issue.

Direct sum codebook sets have specific tree structures that are associated with each of the permutations of the stage codebooks [34]. A tree structure of the two-stage direct sum codebook shown previously in Fig. 7 is shown in the top half of Fig. 12. (The bottom half of Fig. 12 contains a top view of the tree.) The root node corresponds to the origin of the 2-D space in which the codevectors lie, each additional level of the tree represents a particular stage of the RVQ. The nodes at the first level below the root node are the values of the first stage codevectors. The second level node values are the direct sum codevectors of the first and second stage codebooks. Although the tree structure of Fig. 12 seems fairly conventional, consider the tree structure shown in Fig. 13 that corresponds to the direct sum codebook shown in Fig. 8. The branches of this tree structure are intertwined: This is called *entanglement*. The potential for an RVQ tree structure to be entangled is viewed with ambivalence. On the one hand, entanglement hinders the effectiveness of sequential search encoding, but on the other hand, entanglement provides greater freedom in direct sum codevectors placement, which, in turn, allows RVQ's to achieve lower distortion.

The negative aspects of entanglement relate to the inability of a sequential search encoder to access all possible stage codevector combinations. For example, the top view of the entangled tree structure of Fig. 13 is shown in Fig. 14, and overlaid on this tree structure diagram are the decision boundaries of the first stage of a nearest-neighbor sequential search encoder (top figure of Fig. 14) and the decision boundaries of the second stage of a nearest-neighbor sequential search encoder (bottom figure of Fig. 14). When RVQ tree structures violate sequential search decision boundaries in this way, oftentimes, the corresponding renegade direct sum codevectors are never used by the RVQ, and hence, the size of the effective RVQ alphabet is reduced below the size suggested by fixed rate coding [8]. Entanglement can also render other negative effects. For example, exceedingly entangled RVQ's designed with sequential GLA techniques can have nearly coincident quanta (for an example, see Fig. 11). In addition, wayward direct sum codevectors sometimes lie outside the support region of the source pdf (for examples, see [34]). These phenomena can lead to direct sum codevectors or certain path segments through the RVQ tree structure that have zero probability of occurrence.

Fortunately, these potentially detrimental attributes of entangled RVQ tree structures can be controlled or compensated by using techniques during the RVQ design process that are described latter in this paper or by using entropy coding of the RVQ output as in entropy constrained RVQ's. In the latter case, entropy coding allows zero-rate codeword assignments for the inactive portions of the RVQ tree and, hence, in effect provides a type of pruning of the RVQ tree structure. However, one

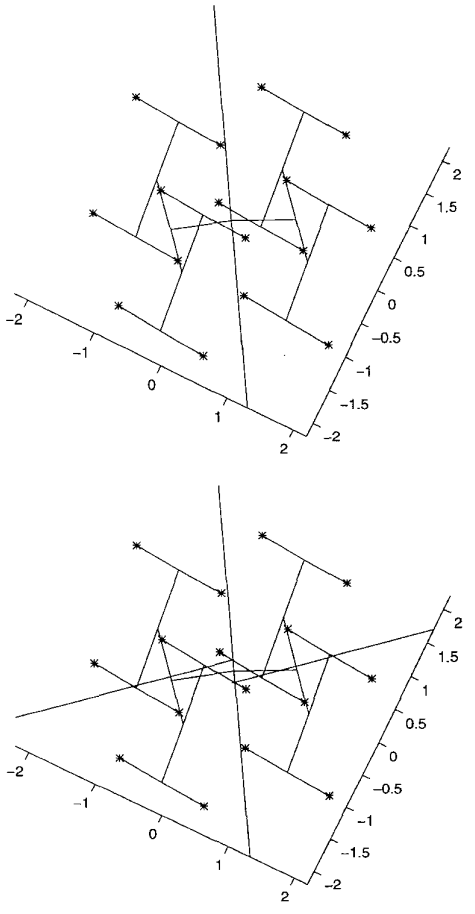


Fig. 14. Entanglement of the tree structure of a four-stage RVQ with two codevectors per stage.

should not embrace the design objective of entirely eliminating entanglement. Although RVQ tree structure entanglement is detrimental to sequential search encoder efficiency, if entangled tree structures can be effectively searched at reasonable computation costs by entanglement-permissive encoders, then increased rate-distortion performance results because entanglement permits a form of variable codevector density, or nonuniform quantization, with direct sum codebook structures.

1) Entanglement-Permissive Encoders: The use of suboptimal, but effective, entanglement-permissive search techniques allows a degree of constructive entanglement to be tolerated and exploited in RVQ design and implementation. Achieving a degree of managed entanglement with the use of entanglement-permissive encoders provides increased performance for a specified but restricted level of encoder search complexity. Various types of entanglement-permissive search techniques are briefly reviewed.

The conventional multiple path search algorithm, or so called M-search technique [47], [48], has been shown by many to be an effective means of increasing the performance of RVQ [8], [36], [49]–[53]. In M-search encoding, the first-stage encoder finds the M closest first-stage codevectors. The second-stage codebook is then searched to find the best second-stage matches for each of the M residual vectors output by the first stage. The M best two-stage path segments selected from the restricted set of examined first- and second-stage codevector combinations are retained as candidate survivor paths through the RVQ tree structure. This process is repeated on the next stage, and so on, until the last RVQ stage has been searched, where the best of the M complete candidate paths is retained. The benefit of M-search is increased performance—the price of M-search is increased cost, but since the number of paths that are retained as survivors is a controllable parameter, careful selection of M permits complexity/performance tradeoffs to be easily managed. Furthermore, techniques for reducing the cost of M-search by using dynamic control of the value of M have been reported in [54].

Multiple path searching, in effect, provides varying degrees of joint search optimization of the RVQ encoder stages. A shortcoming of the conventional M-search technique, however, is that joint search optimization of encoder stages is limited to consecutive stages, that is, the sequential search ordering constraint of the RVQ encoder limits the effect of joint optimization to only neighboring stages. An iterated multiple path search algorithm that is not subject to any particular ordering of the residual quantizer stages has been proposed to overcome this shortcoming [22]. Given some initial path through the RVQ tree (possibly obtained from an M-search), the iterated multiple path search technique is obtained by removing the codebook ordering imposed by the sequential search encoder constraint and then applying a nearest-neighbor encoding rule within an arbitrary sequence of causal–anticausal residual spaces of one or more RVQ encoder stages. This type of coding procedure for generalized product codes and for joint entropy coding systems has been reported in [55].

Another approach for improving RVQ encoder performance is to use locally exhaustive, joint searches. For generalized product codes, this oxymoron means that two or more features are jointly encoded [3]. For the RVQ case, this method can be characterized as a sequence of exhaustive searches over the direct sum codebooks formed from subsets of two or more RVQ stages. As suggested in [3], these searches are performed in the causal residual space of the RVQ. As suggested in [22], these searches are performed in causal–anticausal residual spaces.

2) Entanglement and the RVQ Design Problem: Even with the use of entanglement-permissive encoding procedures, most joint design processes encounter monotonicity difficulties when the complexity of exhaustive encoding is not tolerated, especially if the RVQ has many stages or if training data is limited. Since joint design techniques with the use of suboptimal sequential search encoding are not guaranteed to converge, various ad hoc methods are sometimes employed in the design process to encourage convergence and prevent catastrophic divergence.

The easiest approach for dealing with nonmonotonic design events is to simply capitulate whenever an increase in distortion occurs and terminate the design process. However, experience has shown that nonmonotonic events are often slight and temporary in nature and performance usually im-

proves if the design process is forced to continue. One solution is to run the design process until the iteration count reaches a predetermined limit or until the relative change in distortion is below some threshold for several consecutive iterations [33]. An improved strategy is to run the joint design algorithm for a sufficiently large number of iterations while tolerating nonmonotonic events and recording the average distortion after each iteration. The design process is then repeated using the number of iterations that gave the smallest average distortion [33].

Another method for stabilizing the design process that is suitable for use on RVQ's with many stages is to use a block approach to joint optimization. In this method, only a subset of the RVQ stages are jointly optimized before proceeding to the next block of stages. This optimization is local in the sense that the stages are partitioned into overlapping blocks, and the joint optimization process is restricted to each block of stages. This method was used in [51], [56]–[58].

These previous methods increase only the complexity of the design process with no additional expense required for RVQ implementation. Other methods have also been suggested but require increased RVQ implementation complexity. It has been noted that nonmonotonic design events are less likely, or can be rectified, by increasing encoder complexity (e.g., increasing the number of paths used in M-searching) and/or by increasing codebook storage complexity (e.g., by permitting codebook fanout) [59]. Another approach that improves convergence properties but modifies the structure of RVQ is to impose additional structural constraints on RVQ (e.g., symmetry constraints [16] and algebraic group closure constraints (i.e., lattice constraints) [14], [15]).

Another seemingly ad hoc but in some ways more fitting approach for solving design monotonicity problems is to use separate, and in general, different stage codebooks at the RVQ encoder and decoder [33], [60], [61]. When this approach is used in the sequential-update method, for example, if the decoder and second-stage encoder are fixed in the two-stage case, then a reasonable but not necessarily optimal choice for the first-stage encoder codebook is the current first-stage decoder codebook. Often, this choice results in improved performance, but occasionally, due to entanglement, this approach leads to a nonmonotonic event. Chan and Gersho investigated a method that guarantees monotonicity by disabling the update of the first-stage encoder codebook while still allowing the first-stage decoder codebook to be updated. They also suggested skipping the encoder stage updates (for any stage) whenever the updates would lead to decreased performance [33]. This results in encoder and decoder codebooks that are different from each other.

Miller and Rose suggested that RVQ encoder and decoder mappings should be permitted to be different to improve design descent characteristics and to allow greater freedom in choosing decoder and encoder codebooks. By using a deterministic annealing approach [62] to avoid poor local minima, they optimized the encoder to "best agree" with an exhaustive search partition, and similar to the design objective of the joint-update method, designed the decoder to approximate an "exhaustive search decoder" [60]. They noted that M-search techniques require increased computations and proposed instead that memory be increased to achieve improved performance by keeping a copy of the different decoder codebooks at the encoder to use in forming causal residuals.

The use of separate, and in general, different stage codebooks at the RVQ encoder and decoder is also suggested in [61], but in this case, a copy of the decoder codebooks is not required at the encoder. A new design procedure for generating these types of RVQ's, and example design results are given in the next section.

D. Improved Design Procedure Using Separate Encoder and Decoder Codebooks

RVQ design processes should do the following:

1) achieve joint decoder optimality
2) handle a variety of sequential and/or entanglement-permissive, nongreedy, encoder structures
3) be stable.

Basic principles of a design method that achieves these objectives are given in the next section, and then, example design results of the proposed design process are demonstrated.

1) Basic Principles of the Improved Design Method: Since RVQ encoders have a sequential search structural constraint not required at the decoder and since methods for optimizing sequential search encoders are either unknown or difficult to apply, direct application of the GLA to sequential search RVQ presents some problems. Fortunately, RVQ's with good performance, and encoder and decoder codebook sets well matched to their respective structural constraints can be generated by modifying the GLA design procedure in the following manner. Instead of seeking a new encoder that is *optimal for the new decoder* (as required by the GLA), when designing RVQ's with sequential search encoders, it is more appropriate (since the encoder is subject to constraints not imposed on the decoder) to seek a new encoder that is *improved relative to the old encoder.* This less ambitious design goal is more easily obtained if the encoder and decoder codebooks are permitted to be different.

To describe a design method based on this modified objective, assume that we have a set of fixed direct sum encoder codebooks. Jointly optimal direct sum decoder codebooks are then generated for the partition induced by a sequential search of the fixed encoder codebooks using the interlaced fixed point method of Section II-F-4. Given the new direct sum decoder codebooks and the old set of encoder codebooks, a combination of some subset of the old encoder stage codebooks and some subset of the new decoder stages codebooks is selected to form a new sequential search encoder with improved performance (relative to the performance of the previous sequential search encoder).

The artifice required by this design method is the rule used for choosing different stages from the different prior systems to find an updated encoder with improved sequential search performance. An understanding of the effect of RVQ tree structure entanglement on sequential search effectiveness can provide reasonable heuristic guidelines for either accepting or rejecting optimized decoder stage codebooks as new encoder stage codebooks. One such rule is described next.

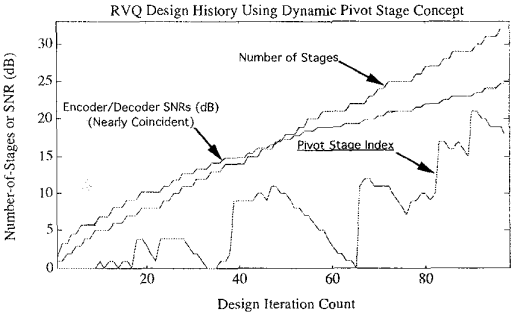


Fig. 15. Example RVQ design history using pivot stage concept.

Suppose we have a given fixed encoder $\mathbf{E}^i$ (where i is an index for the outer iterative design loop), and the inner loop decoder optimization process has just been completed to yield the decoder $\mathbf{D}^i$. This decoder optimization step is guaranteed to give $\mathcal{D}(\mathbf{D}^i) \leq \mathcal{D}(\mathbf{E}^i)$, that is, the new decoder distortion has not been increased relative to the previous encoder distortion. Next, as a first step toward finding an improved encoder, set $\mathbf{E}^{i+1} = \mathbf{D}^i = \boldsymbol{D}_1^i + \boldsymbol{D}_2^i + \cdots + \boldsymbol{D}_P^i$, and evaluate $\mathcal{D}(\mathbf{E}^{i+1})$. Since the use of the updated decoder $\mathbf{D}^i$ in the sequential search encoder $\mathbf{E}^{i+1}$ is not guaranteed to be effective, it is not guaranteed that $\mathcal{D}(\mathbf{E}^{i+1})$ will be less than $\mathcal{D}(\mathbf{D}^i)$. However, considering the fact that the sequential search structural constraint is imposed only at the encoder and not at the decoder, a more appropriate design goal, as stated earlier, is to seek an encoder $\mathbf{E}^{i+1}$ such that $\mathcal{D}(\mathbf{E}^{i+1}) \leq \mathcal{D}(\mathbf{E}^i)$. That is, a sequential search encoder is sought after that performs no worse than the last sequential search encoder (not the updated direct sum decoder). This modification to the design process seems reasonable in that, in this case, knowledge of an optimality condition $\mathcal{O}_E$ does not exist for generating an optimal and efficient sequential encoder. Now, returning to our example, what if the new encoder $\mathbf{E}^{i+1}$ as defined above is such that $\mathcal{D}(\mathbf{E}^{i+1}) > \mathcal{D}(\mathbf{E}^i)$? In this case, we suggest the following approach. Based on the conjecture that it is the inner loop design changes at the first few RVQ stages that causes entanglement at the latter stages [8], it is reasonable to try the sequential search encoder based on using the initial stages of the old encoder $\mathbf{E}^i$ and the latter stages of the new decoder $\mathbf{D}^i$. That is, the new encoder is parametrically selected as $\mathbf{E}^{i+1}(s) = \boldsymbol{E}_1^i + \boldsymbol{E}_2^i + \cdots + \boldsymbol{E}_{s-1}^i + \boldsymbol{D}_s^i + \cdots + \boldsymbol{D}_P^i$, where s is the index of what we choose to call the pivot stage. As described above, the pivot stage index is initially set to 1, and the tentative encoder design $\mathbf{E}^{i+1}(s)$ is tested for improved performance. If $\mathcal{D}[\mathbf{E}^{i+1}(s)] > \mathcal{D}(\mathbf{E}^i)$, then s is incremented by some value, and the redesigned $\mathbf{E}^{i+1}(s)$ is tested to see if $\mathcal{D}[\mathbf{E}^{i+1}(s)] \leq \mathcal{D}(\mathbf{E}^i)$. The process of incrementing the pivot stage index and of testing the resulting direct sum structure as a sequential search encoder is continued until either $\mathcal{D}[\mathbf{E}^{i+1}(s)] \leq \mathcal{D}(\mathbf{E}^i)$ or until $s = P$. If $s = P$, then $\mathbf{E}^{i+1}(s) = \mathbf{E}^i$ and $\mathcal{D}[\mathbf{E}^{i+1}(s)] = \mathcal{D}(\boldsymbol{E}^i)$, and the entire design process can either be stopped or a new stage added to the RVQ.

A summary of the design method is as follows:

STEP 1: Choose an initial $\mathbf{E}^i$.

STEP 2: Use $\mathcal{O}_D$ to find an updated $\mathbf{D}^i$. ($\mathcal{O}_D$ guarantees that $\mathcal{D}(\mathbf{D}^i) \leq \mathcal{D}(\mathbf{E}^i)$.)

STEP 3: Given $\mathbf{E}^i$ and $\mathbf{D}^i$, search for a s such that $\mathcal{D}[\mathbf{E}^{i+1}(s)] \leq \mathcal{D}(\mathbf{E}_i)$.

STEP 4: If the stopping criteria is satisfied, stop. Else, increment i, and continue from STEP 2.

Experiments have shown that there usually exist combinations of old encoder and new decoder stage codebooks that give improved sequential search performance. Thus, a value of s that is less than P is usually discovered that achieves monotonicity. Furthermore, as illustrated by the experimental results given in the next section, the pivot-stage value required to achieve monotonicity can be manipulated to either increase or decrease between RVQ encoder update steps.

2) Fixed-Rate RVQ Design Examples: Fig. 15 illustrates an example that demonstrates that the pivot stage concept can be used to achieve RVQ design monotonicity. The training set contained 8192 vectors of dimensionality 16 sampled from the AR(2) correlated Gaussian source [63]. The initial RVQ was seeded with only one stage codebook, and the final RVQ has a total of 32 stages with two codevectors in each stage. Fig. 15 contains four plotted curves. One plot shows the number of stages as a function of the iteration count of the outer loop of the interlaced fixed point design process. A new stage codebook was added whenever the relative change in distortion between outer loop iterations fell below a threshold value of 0.005 or if the pivot stage index equaled the number of existing stages (which in this example never occurred). The SNR measurements of both the RVQ encoder and decoder, which are nearly coincident, are also plotted in this figure. The fourth plot shows the pivot-stage index value as a function of the iteration count. Note that at certain points during the design process the pivot stage index value increased sharply to find an encoder/decoder stage codebook combination set that provided improved encoder performance. All nonzero pivot stage index values were decremented by one after each outer loop iteration (if an increase in the pivot stage value was not required to maintain monotonicity). The pivot-stage index was not reset to zero to reduce design complexity. Although large pivot-stage values were sometimes required to achieve monotonicity, the pivot-stage values tended to gradually drift back toward zero. In a certain sense, the pivot-stage value determines the degree of "greediness" of the encoder design process. When the pivot-stage index is large, the encoder refuses most of the joint design results of the inner loop and prefers to retain most of the existing sequential search properties of the previous encoder. When the pivot-stage index is small, the encoder is less greedy and readily accepts the joint design results of the inner loop. In this sense, this design process in nongreedy.

Since the improved design method uses encoder and decoder stage codebooks that are generally different, the SNR performances of the encoder and decoder sometimes differ. Fig. 16 illustrates an example, that is also based on the synthetic AR(2) source, where the performances vary significantly. The figure shows the SNR performances of the encoder and decoder for several cases where the training set size varies from only 64 vectors, up to 131 072 vectors of dimensionality 16 (all SNR measurements were obtained from the training set data). The

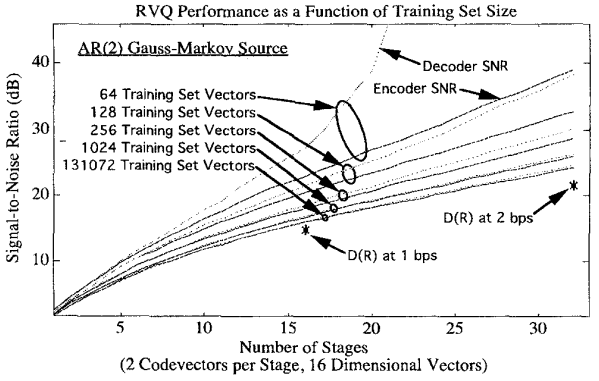


Fig. 16. RVQ performance as a function of training set size using different encoder and decoder codebooks.

number of stages ranges from 1–32, and all stage codebooks contain two codevectors. Note that the SNR performances are monotonically increasing, which suggests that the proposed design method is stable. In addition, note that for small training set sizes and large number of stages, the decoder performance is vastly superior to the encoder performance. In these cases, the decoder has adapted to the local statistics of the finite-sampled training set. Since the decoder is not as structurally constrained as the encoder, it is reasonable to expect that the decoder is capable of superior performance. In fact, the new design process is sufficiently stable that it is possible to design high rate RVQ's that provide essentially perfect decoder codebook representations of sufficiently small training sets. For example, consider the 32-stage RVQ designed with only 64 training vectors shown in Fig. 16. An encoder has been generated that has satisfactory sequential search properties, and a compatible, jointly optimal decoder has been generated that provides a nearly perfect reconstruction—in this instance, an SNR of 133 dB (a level that is out-of-range in Fig. 16) direct sum decomposition of the 64-element training set. In such cases, the RVQ design process is more akin to an analysis/synthesis decomposition process that generates a (nearly) perfect reconstruction direct sum decomposition of the processed data.

The distortion-rate bounds of the AR(2) Gauss–Markov source at 1 and 2 b/sample are also shown in Fig. 16. Note that in all cases, the training-data-SNR performances exceed these bounds. This apparent violation of information-theoretic principles is partly a result of adaptation to local statistics of the finite training set sizes (and illustrates the general danger of testing VQ codes on training data). The use of such small training set sizes is ill advised if the intent is long-term use of the resulting codebooks for future sampled outputs of the source, but if, instead, a means for efficiently communicating the decoder codebooks as overhead information can be found, then the performance advantages of direct sum codebook adaptation to local statistics may return a profitable rate-distortion tradeoff. These possibilities are addressed further in the next section, where the use of RVQ in image-adapted coding systems is examined.

3) Image-Adapted RVQ Design Examples: One of the concerns associated with applied VQ is out-of-training set performance. If a small or unrepresentative training set is used, the resulting codebook performs poorly. This problem is normally corrected by increasing the training set size, but even so, if a VQ is expected to encounter a wide variety of image types, then instead of providing superior performance for each type of imagery, the training set representation of the composite-source probability density function often results in codebooks with mediocre performance on each of the different image types. This problem is solved with the use of image-adaptive RVQ. Instead of a liability, the sensitivity of codebook design to various image types becomes an asset (ignoring for the moment the complexity of on-line codebook generation). Experiments that illustrate the differences in performances between a generic RVQ designed for a wide class of imagery, and image-adapted RVQ's designed for each particular image type where codebook overhead transmission is included in rate calculations, are provided in [18]. Experiments where the side information associated with the transmission of adapted codevectors is considered in a rate-distortion context is contained in [64]. Another type of on-line, adaptive RVQ is described in [65].

Sample results that illustrate image-adaptive RVQ performance and show the effectiveness of the new design method for image waveform coding are shown next. Fig. 17 is a 8-b gray-scale image called "Moffet" that has 1024×1024 pixels. The inset shows a zoomed-in subregion of the image that facilitates subjective evaluations. When a 32-stage RVQ with four codevectors per stage is generated using the sequential GLA design method, the decoded image of Fig. 18 is obtained. Please keep in mind the caveat that in these experiments, the image-adapted codebooks were both trained and tested on this single image. Furthermore, the mean of each vector is removed prior to RVQ coding, and the information needed to communicate the codebook and mean information is regarded as side information. If a single-precision 32-bit format is used to represent each codevector element, then the overhead rate associated with the 32–stage codebooks is 0.25 bits per pixel (b/pixel) when amortized over the entire 1024×1024 image. The vector means are assumed to be reconstructed without loss at the RVQ decoder. This sequential GLA RVQ obtains a 255 peak signal-to-noise ratio (PSNR) of 27.65 dB when 1.0 b/sample are expended on RVQ index information (fixed rate coding). Using the interlaced fixed-point method together with the pivot-stage concept of Section III-D, the jointly optimized RVQ with the same number of stages and same stage codebook size obtains a PSNR of 28.45 dB (see Fig. 19).

The plots of Fig. 20 illustrate the performance improvements obtained when M-search is used. Increased values of M yield increased SNR performance; however, this performance gain comes at the price of increased encoder complexity. One may be tempted to say that joint optimization produces a certain performance gain and that M-search provides an additional coding gain, but this presupposes that the two gains are separable. This supposition is wrong if M-search is employed during the design process. The use of M-search facilitates the natural tendencies of joint optimization by permitting the growth of entangled RVQ tree structures and results in a nonseparable coding gain [43].

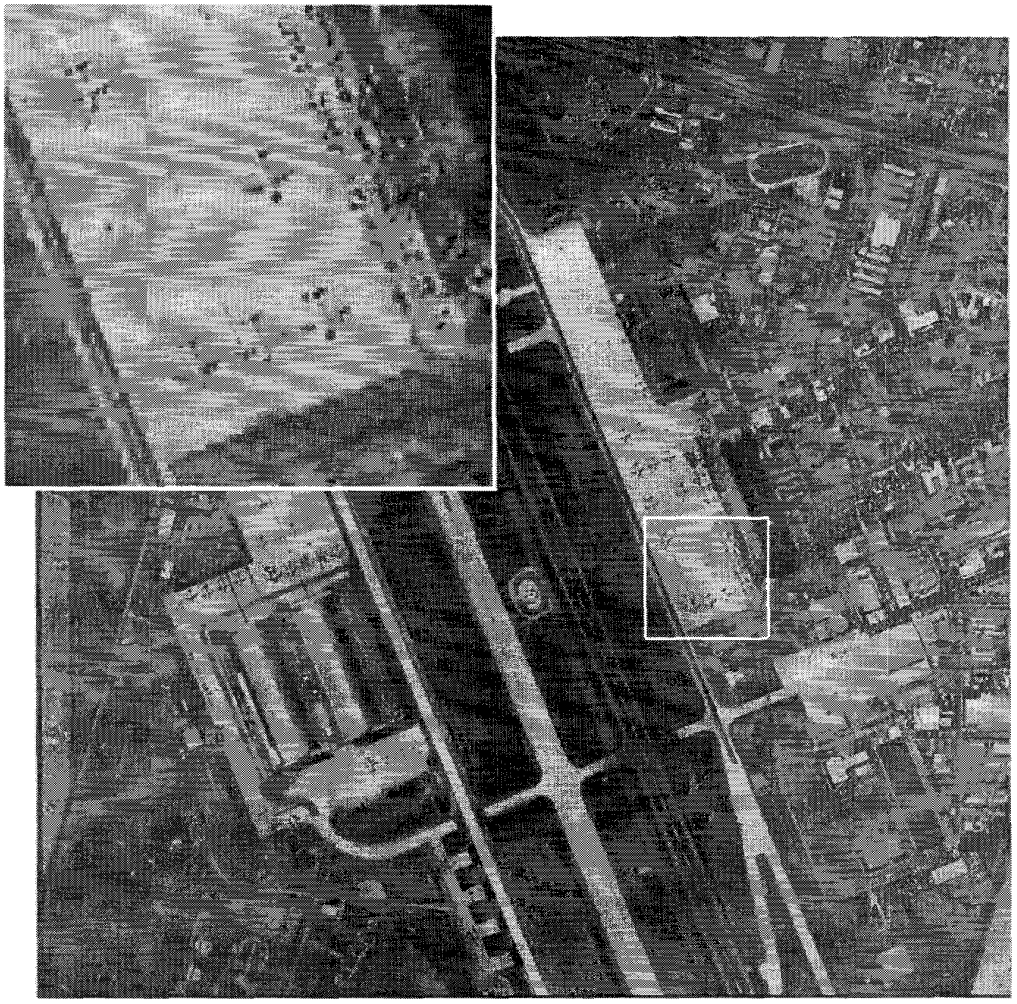


Fig. 17. Original 1024 × 1024 pixel Moffet image.

Fig. 21 shows the jointly-optimal, image-adapted RVQ result when a M-search value of 16 is used. The performance improvement over the sequential GLA is 1.86 dB. Whether or not the corresponding subjective coding gain is significant depends on the intended image exploitation task. At best, the subjective coding gain of this example can only be characterized as slight. The typically modest performance gains of jointly optimized, fixed-rate RVQ's has also been documented in [33]. The real advantages of the RVQ structure, which will be illustrated in subsequent sections, lie in the application of RVQ in variable-rate, predictive, and successive approximation coding systems, but before we examine such systems, we note ways that the complexity of the joint design process can be reduced for image-adapted applications and review an additional technique for the joint design of RVQ codebooks.

One of the concerns of image-adapted RVQ applications is design/encoding complexity. RVQ design complexity is roughly proportional to the square of the number of stages. There are ways to reduce the complexity of both the design and implementation of RVQ. One way is to use the technique proposed in [66], where comparisons are made with partial distortion sums to prevent unnecessary multiply-add operations. This technique is particularly useful when vector sizes are large. A way of reducing the complexity of the decoder optimization process is to form the causal–anticausal residual not by taking the input vectors and subtracting the codevectors of all stages but one, but by computing the total residual at the output of the last stage and then forming the causal–anticausal residual by adding the codevectors of one stage to the total residual. The price paid for this reduction in the computational burden is the additional memory required to store the total residuals.

E. Constrained Joint Optimization Techniques

In the previous sections, we discussed several techniques for designing a set of jointly optimized RVQ codebooks subject to a sequential search constraint. Recently, a new design based on neural networks was developed by Rizvi and Nasrabadi [25], [26]. In this technique, a multilayer competitive neural network

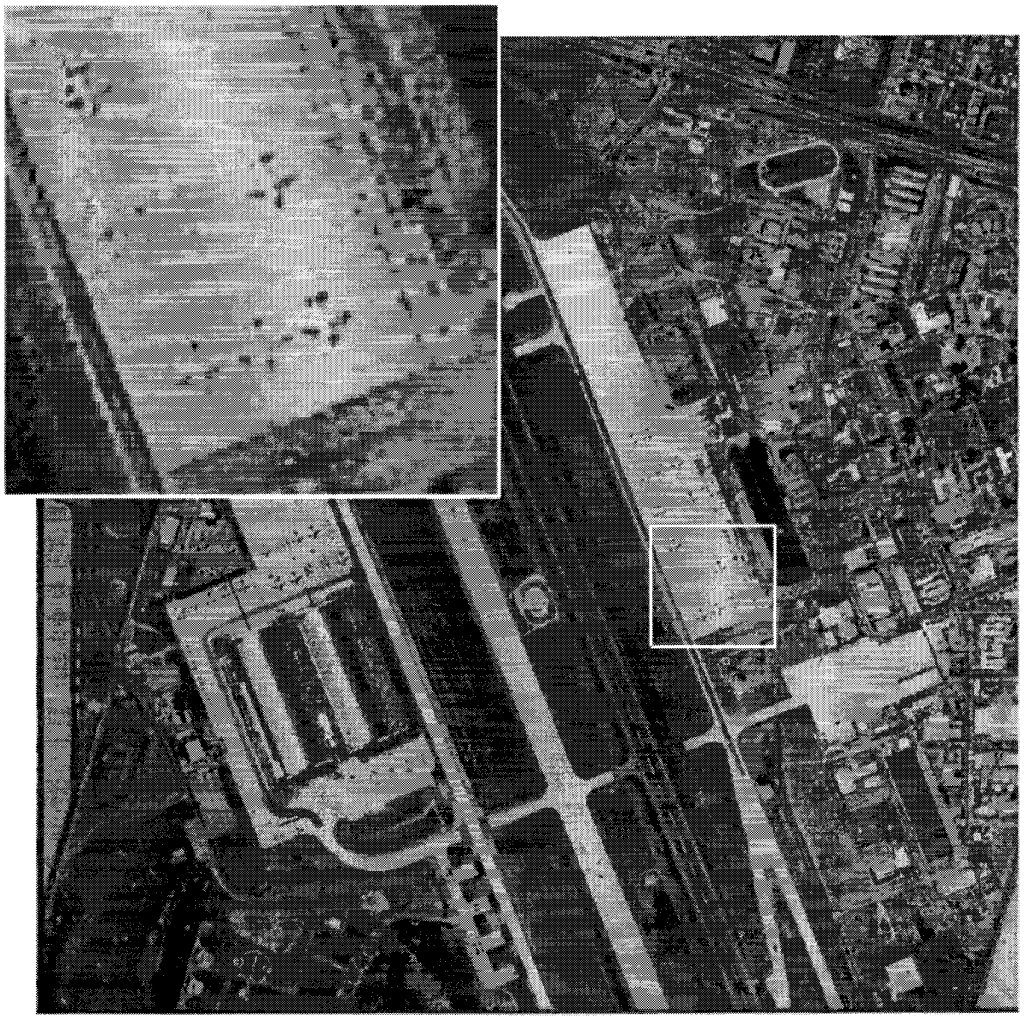


Fig. 18. Moffet image coded with a mean-removed (sequential GLA) 32-stage RVQ using four codevectors per stage and 8 × 8 blocks. RVQ index rate = 1.0 b/pixel, RVQ codebook overhead rate = 0.25 b/pixel, SNR = 27.65 dB.

is used to jointly optimize the RVQ-stage codebooks. The RVQ design problem is considered as a nonlinearly constrained optimization that is formulated as minimizing a Lagrangian error function. This algorithm implements a nonlinear gradient descent minimization of the causal error with a constraint on the anticausal error. We now briefly describe the basic principle of this technique.

Consider an m-stage RVQ. The pth-stage squared error

$$\mathbf{E}_p[\mathbf{y}_p(i_p)] = \tfrac{1}{2}\left\|\mathbf{x}_1 - \sum_{j=1}^{p} \mathbf{y}_j(i_j)\right\|^2 \tag{5}$$

must be minimized to obtain optimal stage codevectors. (The "$\frac{1}{2}$" factor used to scale the distortion measure simplifies the final updating equations.) Let $\mathbf{G}_q[\mathbf{y}_p(i_p)]$ be the squared error caused by the subsequent stages indexed by $q = p+1, p+2, \cdots, m$, which can be expressed as

$$\mathbf{G}_q[\mathbf{y}_p(i_p)] = \tfrac{1}{2}\left\|\mathbf{x}_1 - \sum_{k=1}^{q} \mathbf{y}_k(i_k)\right\|^2. \tag{6}$$

In the constrained optimization approach the objective is to minimize $\mathbf{E}_p[\mathbf{y}_p(i_p)]$ while subject to the constraint that $\mathbf{G}_q[\mathbf{y}_p(i_p)] = 0$ for all subsequent stages. This task is formulated as minimizing the Lagrangian error function $\mathbf{D}_p$ of the pth stage

$$\mathbf{D}_p[\mathbf{y}_p(i_p), \lambda] = \mathbf{E}_p[\mathbf{y}_p(i_p)] + \sum_{q=p+1}^{m} \lambda_{pq}\mathbf{G}_q[\mathbf{y}_p(i_p)] \tag{7}$$

where λ_{pq} is the Lagrangian multiplier. Minimization is achieved by using a gradient-descent-like method based on updating equations developed for training a multi-layer competitive neural network. The updating equations can be found in [25] and [26].

An advantage of this technique is that final design results are independent of the choice of initial stage codebooks, and with carefully chosen learning rate and Lagrangian multipliers, convergence to a fixed point is guaranteed. A drawback is the excessive training time that is required by the neural network. A batch-mode version has been developed that requires significantly less training time but at the expense of

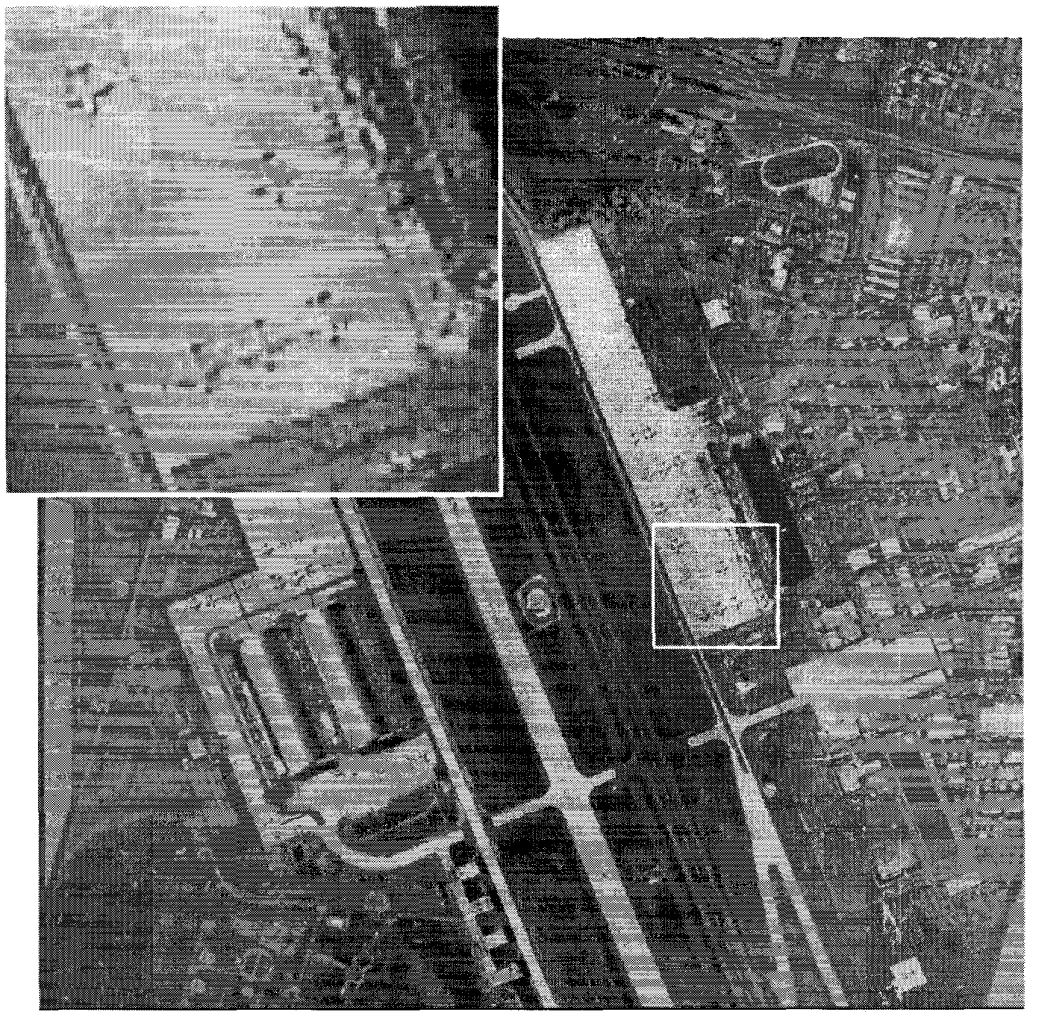


Fig. 19. Moffet image coded with a mean-removed (jointly optimal) 32-stage RVQ using four codevectors per stage and 8×8 blocks. RVQ index rate = 1.0 b/pixel, RVQ codebook overhead rate = 0.25 b/pixel, SNR = 28.45 dB.

slightly deteriorated performance [67]. A similar constrained optimization technique has also been reported for jointly optimal predictive vector quantizers [68].

IV. Variable-Rate RVQ

Variable rate RVQ's that rely on entropy coding for their variable rate structure are reviewed. Reductions in entropy that often result from the imposition of the direct sum codebook structure and the sequential search constraint are shown to partially compensate for the concomitant increases in distortion. The use of partial conditioning is shown to permit the memory requirements of entropy coded RVQ's to be controlled and managed.

A. Direct Sum Product Codes and Entropy

The prior sections have shown that the imposition of a direct sum product code constraint leads to an unavoidable increase in distortion, but since structured systems are inherently "less random" or "more ordered," it is reasonable to expect that the imposition of structure also reduces output entropy. Experimental results have shown that indeed this is the case [69]. Consider, for example, the distortion-entropy points shown in Fig. 22 that correspond to several VQ's: some with and some without the direct sum product code constraint. The distortion-entropy pair of a single-stage ESVQ designed and tested on the commonly used 512×512 Lena image is shown in the top right corner of Fig. 22. This ESVQ has 256 codevectors of dimension 4×4 and, hence, a fixed rate of 0.50 b/pixel. The entropy of this codebook (relative to the Lena image) is 0.4538 b/pixel, and the PSNR is 31.53 dB. A corresponding two-stage exhaustive search direct sum codebook with 16 codevectors in each stage yields 30.37 dB at an entropy of 0.4287 b/pixel. An eight-stage exhaustive search direct sum codebook with two codevectors in each stage yields 29.86 dB at an entropy of 0.3997 b/pixel. In each case, a decrease in entropy (relative to that of the ESVQ) compensates in part for the increase in distortion due to the imposed direct sum structural constraint.

Previously, RVQ tree structure entanglement was shown to lead to a reduction in the effective direct sum alphabet size

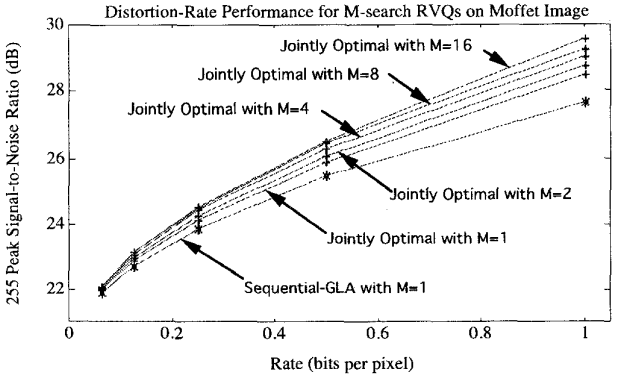


Fig. 20. Rate-distortion performances of jointly optimized, image-adapted RVQ's on the Moffet image as a function of the design method and the number of paths used in M search.

by causing some of the RVQ tree segments to suffer a zero probability of occurrence; this "dead wood" also lowers the output entropy of the RVQ. For example, the distortion-entropy points of the two- and eight-stage RVQ's, when sequential search encoders are employed, are also shown in Fig. 22. The two-stage RVQ yields 30.07 dB at an entropy of 0.4202 b/pixel, and the eight-stage RVQ yields 29.26 dB at 0.3830 b/pixel. This decreased entropy (relative to the corresponding exhaustive search direct sum product codes) compensates in part for the increased distortion due to the sequential search structural constraint. The next sections review how entropy codes have been integrated into the RVQ framework to improved performance and partly compensate for the increased distortion consequences of both the direct sum codebook constraint and the sequential search constraint.

B. Entropy-Coded RVQ

The lower entropy of direct sum codebooks can be exploited by entropy coding the RVQ output. Entropy coding partially compensates for the increased distortion due to the constraints of the direct sum codebook structure and makes it possible to in effect "cut away" the dead wood of severely entangled tree structures. Thus, cascaded RVQ/entropy codes achieve lower average rates while incurring no additional increase in distortion. However, the increase in performance comes at the expense of dealing with variable rate buffering problems and the cost of memory required to store entropy codetables.

A P-stage encoder generates P-tuple indices $\boldsymbol{i} = (i_1 i_2 \cdots i_P)$ that can be considered as symbols in an extended source alphabet $\mathbf{I}$, and a set of variable length codes (such as Huffman codes) can be designed for this extended alphabet. A lower bound to the length of such codewords, when each index is encoded without consideration of the values of neighboring indices, is the self-information measure of the direct sum index $l(\boldsymbol{i}) = -\log[P(\boldsymbol{i})]$, where $P(\boldsymbol{i})$ is the probability of the composite index. (If $P(\boldsymbol{i}) = 0$, then $l(\boldsymbol{i})$ is defined to be 0.) Thus, an entropy codeword with length at least as large as $l(\mathbf{i})$ must be stored for each $\boldsymbol{i} \in \mathbf{I}$. If the direct sum codebook size becomes too large, the memory associated with the codetable becomes excessive. Fortunately, the multiple stage structure of RVQ makes it possible to control memory cost by using conditional entropy coding, but before we show this, let us first consider the optimization of cascaded RVQ/entropy coding systems by jointly designing both the quantizer and entropy coder [70].

C. Entropy-Constrained RVQ

Similar to unstructured VQ's [71], the distortion of structured RVQ's can be minimized while subject to a constraint on the output entropy of the quantizer [72]. Necessary conditions that the stage output values $\boldsymbol{y}_p(i_p)$ of a RSQ must satisfy for entropy constrained joint optimality follow from the requirement

$$\frac{\partial}{\partial \boldsymbol{y}_p(i_p)} [\mathcal{D} - \lambda(H - H_c)] = 0 \tag{8}$$

for some constant λ and while subject to the constraint that the output entropy H of the RSQ does not exceed H_c. The necessary conditions are satisfied when the stage quanta are centroids of the now familiar causal–anticausal residual. Encoder optimization leads to a biased nearest-neighbor decision rule, where the bias terms are λ times the log probability terms $l(\boldsymbol{i})$, that lower bound the length of the variable-length codewords assigned to the composite RSQ indexes. Vector generalizations of these concepts lead to the development of entropy-constrained RVQ's. An iterative descent algorithm based on a Lagrangian formulation can be used for designing jointly optimal, entropy-constrained RVQ's [44].

Just like jointly optimal fixed-rate RVQ systems, for a given cost, entropy-constrained RVQ's can provide performances superior to that of unstructured, exhaustive search, entropy-constrained VQ's [72]. These performance gains come not only from the use of larger block sizes but also from the larger practical peak rates that are obtainable with RVQ. As background information, it has been shown that a significant source of the performance gain of entropy constrained VQ over simply entropy coding the output of a fixed rate VQ comes from the larger entropy constrained codebook size used to achieve the same rate as that of the entropy coded codebook, and that this gain increases as the initial alphabet size increases (ideally, the initial alphabet should be infinitely large) [73], [74]. However, the computation and memory requirements associated with unstructured systems impose a rather modest size constraint on the initial codebook. Thus, realizable entropy constrained (single stage) VQ systems are actually both alphabet and entropy constrained [75], [76]. The implementation efficiencies of RVQ can be used to push the practical peak rates of entropy constrained systems to larger values and, in return, achieve additional performance advantages over unstructured, single-stage systems [55]. A detailed treatment of the optimality conditions, the design process, and the performances of entropy-constrained RVQ's on synthetic sources is given in [44]. The performance of entropy-constrained RVQ's on image sources is described in [77].

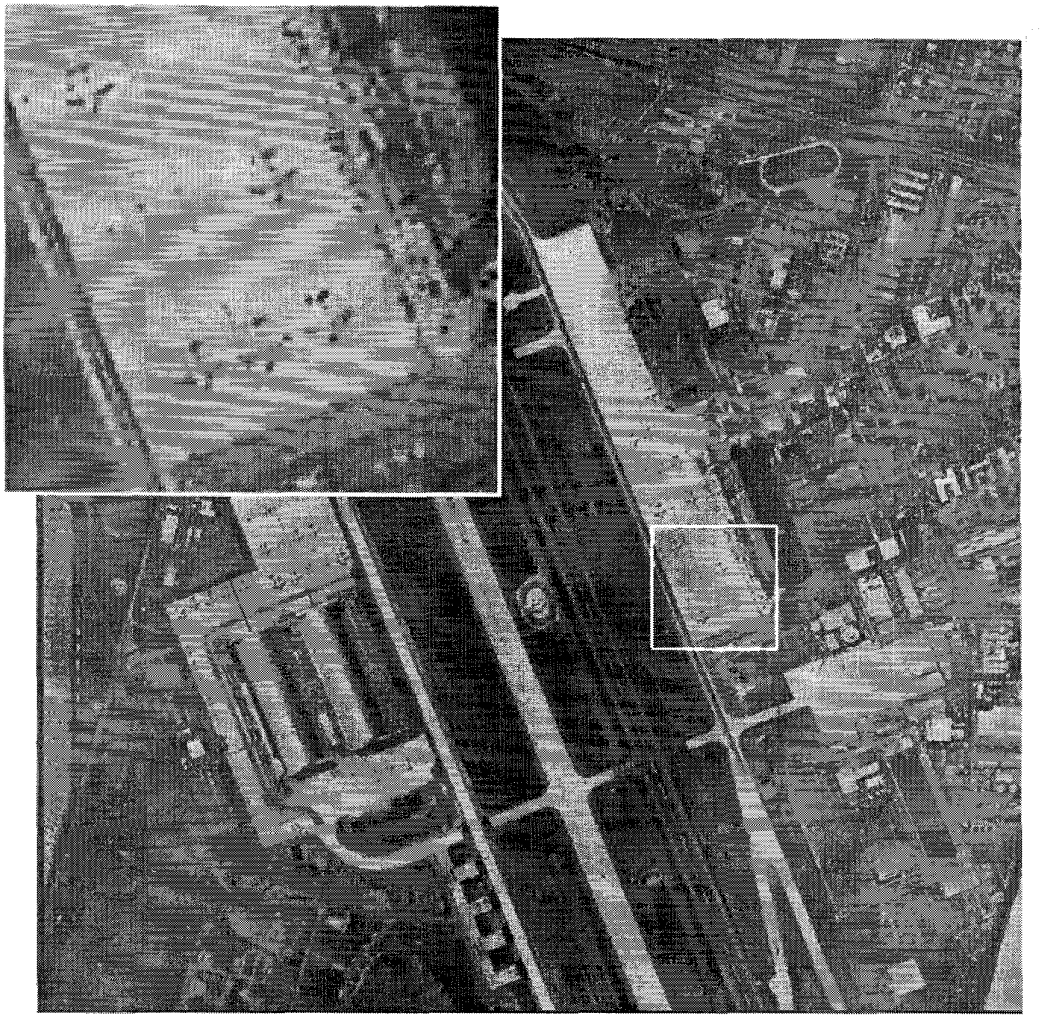


Fig. 21. Moffet image coded with a mean-removed (jointly optimal) 32-stage RVQ using $M = 16$-search, four codevectors per stage, and 8×8 blocks. RVQ index rate = 1.0 b/pixel, RVQ codebook overhead rate = 0.25, PSNR = 29.51 dB.

D. Conditional Entropy-Constrained RVQ

Since a variable-length codeword must be stored for each composite direct sum index, the size of the entropy tables of entropy-constrained RVQ's become unwieldy at high rates. However, the following conditional chain rule formulation of direct sum index probability can be exploited to reduce entropy code memory requirements.

Consider the probability $P(\boldsymbol{i})$ of the composite direct sum index $\boldsymbol{i} = (i_1 i_2 \cdots i_P)$. This joint probability can be expressed as a product of conditional probabilities

$$P(i_1, i_2, \cdots, i_P) = P(i_P|i_{P-1}, i_{P-2}, \cdots, i_1) \cdot P(i_{P-1}|i_{P-2}, i_{P-3}, \cdots, i_1) \cdots P(i_2|i_1)P(i_1) \quad (9)$$

and consequently, the self-information of the direct sum codeword can be expressed as

$$l(\boldsymbol{i}) = l(i_P|i_{P-1}, i_{P-2}, \cdots, i_1) + l(i_{P-1}|i_{P-2}, i_{P-3}, \cdots, i_1) + \cdots + l(i_2|i_1) + l(i_1) \quad (10)$$

where $l(i_p|i_{p-1}, i_{p-2}, \cdots, i_1)$ is equal to $-\log[P(i_p|i_{p-1}, i_{p-2}, \cdots, i_1)]$ for $p = 2, \cdots, P$. Equation (10) shows that the self-information of a direct sum index, which is a lower bound to the length of any realizable variable-length codeword, is equal to a sum of stage-dependent terms that are functions of probabilities conditioned on the encoding decisions of prior stages.

Ideally, one should be able to construct an optimum variable-length codeword for a direct sum index in an incremental fashion by obtaining a first-stage codeword with an optimum length $l(i_1)$, then appending the second-stage increment with ideal length $l(i_2|i_1)$, and so on for each of the appended increments with lengths $l(i_p|i_{p-1}, i_{p-2}, \cdots, i_1)$ for $p = 3, \cdots, P$. However, the memory required to store the codewords corresponding to just the last-stage increment $l(i_P|i_{P-1}, i_{P-2}, \cdots, i_1)$, for example, is proportional to $|\mathbf{I}_P| \times |\mathbf{I}_{P-1}| \times \cdots \times |\mathbf{I}_1|$, where $|\mathbf{I}_p|$ denotes the cardinality of $\mathbf{I}_p$. This memory requirement of just the Pth-stage increment is equal to the entire amount of memory required for direct entropy coding of the RVQ output without conditioning. The

cost of the memory needed for the other stage's conditional codewords must also be incurred. Thus, instead of reducing memory costs, it appears as though memory requirements have been increased by the introduction of conditioning. However, a complexity reducing feature of conditional entropy-constrained RVQ is the potential to restrict the degree of conditioning to only a certain number of prior stages and thus limit the amount of memory needed for conditional entropy tables [72], [78], [79]. For example, if conditioning on only m prior stages is permitted, then the memory requirements of the pth stage are reduced to $|\mathbf{I}_p| \times |\mathbf{I}_{p-1}| \times \cdots \times |\mathbf{I}_{p-m}|$. For the extreme case where no conditioning is permitted, zeroth-order entropy coding produces composite direct sum codewords with lengths lower bounded by

$$l(i_1, \cdots, i_P) = \sum_{p=1}^{P} l(i_p). \tag{11}$$

In this case, only one variable-length codeword need be stored for each stage codevector. If first-order conditioning is permitted, then the lower bound becomes

$$l(i_1, \cdots, i_P) = l(i_1) + \sum_{p=2}^{P} l(i_p|i_{p-1}) \tag{12}$$

which, when averaged over the composite indices, is smaller than (11), and only the relatively small, first-order conditional entropy codetables need to be stored. Similar expressions hold for each possible degree of conditioning.

The price paid for the reduction in memory requirements through restricted conditioning is increased average codeword length since decreased conditioning can only increase the conditional entropy. Thus, one of the design objectives of conditional entropy-coded RVQ's should be to minimize the increase in the average lengths of the composite codeword lengths as the degree of conditioning is restricted.

Although the previous examples show restrictions where conditioning is limited to a specific number of prior stages, a better performance-cost tradeoff can be achieved if the number N of conditioning states is restricted, and the value of m used for each conditioning state is allowed to vary [80]–[82]. Furthermore, in addition to conditioning on the output of previous stages, conditional entropy-constrained RVQ can also employ conditioning on previously coded vectors [83]–[86] and, in subband coding systems, on surrounding subbands [81], [82]. In general, one can define a collection of conditioning states $\{\mathcal{S}_n(\cdot): n = 1, \cdots, N\}$ that depend on a set of general state parameters $\theta_1, \theta_2, \cdots, \theta_K$. The variable-length codes that result from such finite-state conditional codes can be described [55] with ideal lengths given by

$$l(i_1, \cdots, i_P) = \sum_{p=1}^{P} l[i_p|\mathcal{S}_n(\theta_1, \theta_2, \cdots, \theta_K)]. \tag{13}$$

The performance of this approach is strongly dependent on how states are defined and on how many states N are allowed to exist. The design problem is to select the number of states and the particular states that result in the lowest possible entropy for a given cost.

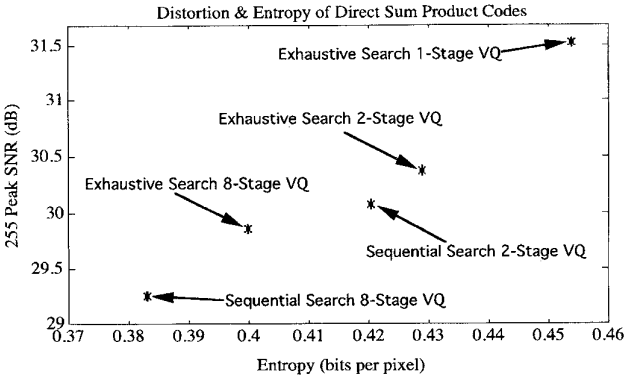


Fig. 22. Distortion-entropy points realized by several VQ's and RVQ's.

Kossentini *et al.* [81], [82], [87]–[93] have used conditional entropy coding based on intraband and interband coding decisions and on prior RVQ stages and have developed algorithms to locate good conditioning state parameters by jointly optimizing the overall quantizer structure while constraining both entropy and complexity. This approach has the advantage that explicit bit allocation is avoided and has led to image coding performance for RSQ versions that is competitive with that of embedded wavelet transform coding systems.

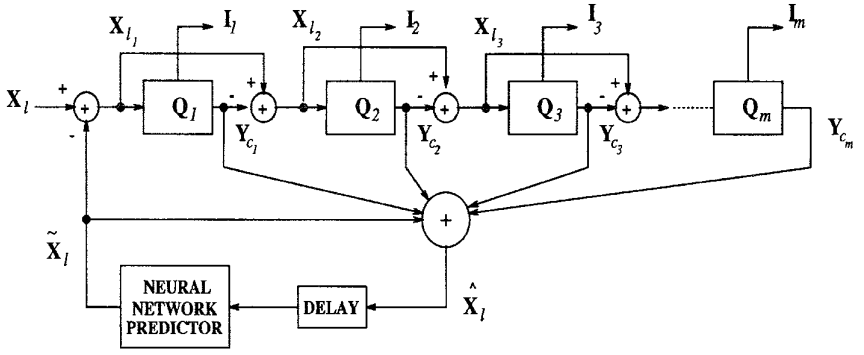
V. RVQ's with Memory

A restriction on vector dimensionality can be imposed on RVQ to further control computation and memory requirements, especially at high rates. A small vector dimension, however, leaves neighboring blocks highly correlated. This interblock correlation can be exploited by incorporating memory into RVQ. Two RVQ schemes with memory, predictive residual vector quantization (PRVQ) [94], [95], and finite-state residual vector quantization (FSRVQ) [96]–[98] are reviewed.

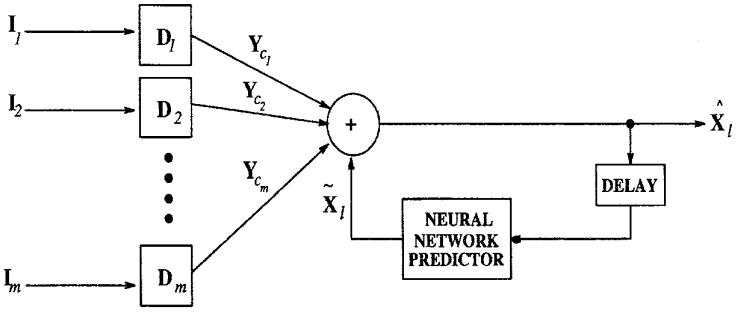
A. Predictive RVQ

PRVQ predicts the current block from previously encoded blocks and constructs a prediction residual vector that is encoded by the RVQ [94], [95]. Three PRVQ design methods are discussed. The first method is based on a constrained optimization technique in which each constituent component of the PRVQ is optimized by minimizing an appropriate stage error function with a constraint on the overall error. This technique makes use of a Lagrangian formulation and iteratively solves the error function to obtain a locally optimal solution. The second method is based on the Chang and Gray scheme [99], [100], where the predictor and quantizers are jointly-optimized by minimizing the overall error. The third method is an extension of the closed-loop design technique suggested in for designing predictive vector quantizers (PVQ's).

Fig. 23 shows a PRVQ encoder and decoder consisting of a neural network predictor and a m-stage RVQ. The neural network predictor is based on a multilayer perceptron [101], [102]. Before reviewing these three methods, let us first establish notation and to be consistent with notation used in



A PRVQ with an m-stage quantizer



Decoder for a PRVQ

Fig. 23. Predictive residual vector quantizer with an m-stage RVQ.

previous, more detailed papers [94], [95]; there is a slight divergence between the notation used in this section for RVQ's with memory and the other sections that deal with memoryless RVQ's.

Let $\mathbf{X}_l$ be the lth vector in the training set, let $\tilde{\mathbf{X}}_l$ be the prediction of the vector $\mathbf{X}_l$, and let $\hat{\mathbf{X}}_l$ be the reconstruction of the vector $\mathbf{X}_l$. Let $\mathbf{Y}_{c_p}$ be the cth codevector in the pth-stage codebook, and let $\mathbf{X}_{l_p}$ be the pth-stage residual input vector given by

$$\mathbf{X}_{l_p} = \begin{cases} \mathbf{X}_l - \tilde{\mathbf{X}}_l, & \text{if} \quad p = 1; \\ \mathbf{X}_l - \tilde{\mathbf{X}}_l - \sum_{q=1}^{p-1} \mathbf{Y}_{c_q^*} & \text{otherwise} \end{cases} \tag{14}$$

where $\mathbf{Y}_{c_q^*}$ represents an optimal qth-stage codevector. The distortion is the squared error measure $d(\mathbf{X}_l, \mathbf{Y}_l) = ||\mathbf{X}_l - \mathbf{Y}_l||^2$. The pth-stage quantizer $\mathbf{Q}_p(.)$ maps the residual input vector $\mathbf{X}_{l_p}$ to $\mathbf{Y}_{c_p^*}$ if $d(\mathbf{X}_{l_p}, \mathbf{Y}_{c_p^*}) \leq d(\mathbf{X}_{l_p}, \mathbf{Y}_{c_p})$ for all $c_p \neq c_p^*$, and the reconstructed vector $\hat{\mathbf{X}}_l$ is given by $\hat{\mathbf{X}}_l = \tilde{\mathbf{X}}_l + \mathbf{Y}_{c_1^*} + \mathbf{Y}_{c_2^*} + \mathbf{Y}_{c_3^*} + \cdots + \mathbf{Y}_{c_p^*} + \cdots + \mathbf{Y}_{c_m^*}$.

A multilayer competitive neural network [25], [26] is used in the constrained optimization and the jointly optimized design methods. The GLA is used in the closed-loop design method. Cost functions used in these joint predictor-quantizer design methods are as follows. $\mathbf{E}_\Psi(\tilde{\mathbf{X}}_l) = \frac{1}{2}||\mathbf{X}_l - \tilde{\mathbf{X}}_l||^2$ is the squared error minimized by the predictor, and

$$\mathbf{G}_m(\tilde{\mathbf{X}}_l) = \frac{1}{2}\left\|\mathbf{X}_l - \tilde{\mathbf{X}}_l - \sum_{q=1}^{m} \mathbf{Y}_{c_q^*}\right\|^2 \tag{15}$$

is the overall error when the predictor is updated. Similarly

$$\mathbf{E}_p(\mathbf{Y}_{c_p}) = \frac{1}{2}\left\|\mathbf{X}_l - \tilde{\mathbf{X}}_l - \sum_{k=1}^{p-1} \mathbf{Y}_{c_k^*} - \mathbf{Y}_{c_p}\right\|^2 \tag{16}$$

for $1 \leq p < m$ is the squared error minimized to obtain the best reproduction pth-stage codevector, and

$$\mathbf{G}_m(\mathbf{Y}_{c_p}) = \frac{1}{2}\left\|\mathbf{X}_l - \tilde{\mathbf{X}}_l - \sum_{k=1}^{p-1} \mathbf{Y}_{c_k^*} - \mathbf{Y}_{c_p} - \sum_{q=p+1}^{m} \mathbf{Y}_{c_q^*}\right\|^2. \tag{17}$$

is the overall error when the pth stage is updated.

1) Constrained Optimization Technique: In the constrained optimization technique, each component of the PRVQ is designed by minimizing the error due to that particular component with a constraint on the overall error. For the predictor, the task is to minimize $\mathbf{E}_\Psi(\tilde{\mathbf{X}}_l)$ with the constraint that $\mathbf{G}_m(\tilde{\mathbf{X}}_l) = 0$. For the pth-stage quantizer, the task is to minimize $\mathbf{E}_p(\mathbf{Y}_{c_p})$ with the constraint that $\mathbf{G}_m(\mathbf{Y}_{c_p}) = 0$. Using a Lagrangian formulation, the predictor error function is $\mathbf{D}_\Psi(\tilde{\mathbf{X}}_l, \lambda) = \mathbf{E}_\Psi(\tilde{\mathbf{X}}_l) + \lambda \mathbf{G}_m(\tilde{\mathbf{X}}_l)$, and the pth-stage quantizer error function is $\mathbf{D}_p(\mathbf{Y}_{c_p}, \beta) = \mathbf{E}_p(\mathbf{Y}_{c_p}) + \beta \mathbf{G}_m(\mathbf{Y}_{c_p})$, where λ and β are Lagrangian multipliers. The above Lagrangian error functions are minimized with a gradient-descent approach that finds the weights of the neural network predictor and the stage codebooks of the RVQ.

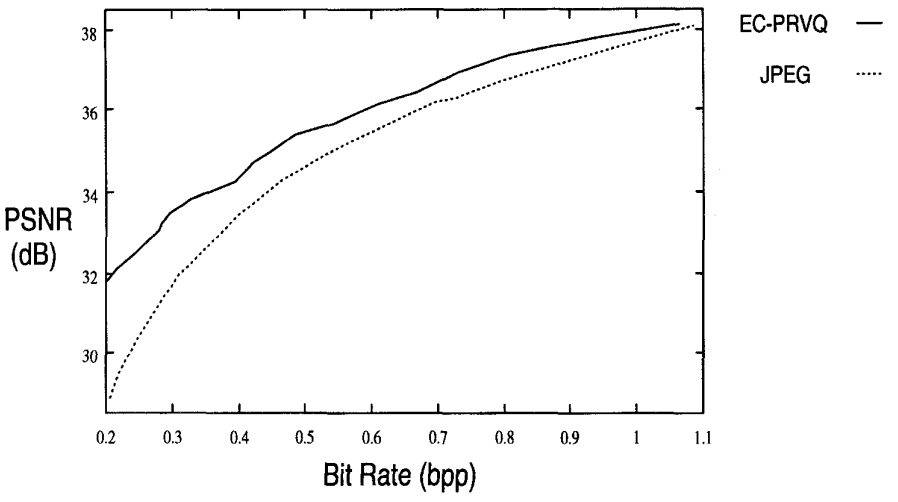


Fig. 24. Rate-distortion performance comparison of EC-PRVQ and JPEG for the test image Lena.

2) Joint Optimization Technique: The joint optimization technique is an extension of the Chang and Gray method [99] used for designing jointly optimized predictive VQ's. In this technique, the predictor and each of the quantizers seek to jointly minimize the overall error $\mathbf{G}_m(\mathbf{Y}_{c_p})$.

3) Closed-Loop Design Technique: In the closed-loop design technique, the predictor and the quantizers are initially designed separately, and then, the quantizers are further optimized by closing the loop [1]. This design technique is based on the following iterative procedure:

STEP 1: Obtain the residual signal by subtracting predicted blocks from original blocks (while the predictor inputs are original blocks). Set $p \rightarrow 1$.

STEP 2: Design the pth-stage codebook based on the causal residual using the GLA.

STEP 3: Close the loop, and update the pth-stage codebook for the fixed predictor by rerunning the GLA (note that after closing the loop, the prediction is based on previously encoded blocks).

STEP 4: Form residual blocks by subtracting encoded blocks from original blocks when the predictor inputs are previously encoded blocks.

STEP 5: Set $p \rightarrow p + 1$. If $p \leq m$, then goto STEP 2; otherwise, STOP.

4) Predictive Tree-Structured RVQ: We now consider a variation of the PRVQ called the predictive tree-structured residual vector quantizer (PTSRVQ) in which each stage of the residual quantizer is replaced by a tree-structured VQ [95]. PTSRVQ is implemented with a predictor with a modified multilayer competitive neural network structure. The memory requirements of PTSRVQ are almost the same as that of an PVQ. The increase in memory costs for PTSRVQ, however, pays off in terms of a PSNR performance gain. The computational complexity of PTSRVQ is the same as that of PRVQ. Therefore, PTSRVQ is the preferable system at high rates, if memory restrictions are not stringent.

5) Entropy-Constrained Predictive RVQ: In Section IV, we discussed the design of entropy-constrained RVQ, which gives significant performance improvements relative to simple RVQ. Similarly, the performance of a PRVQ can be improved if a constraint on the output entropy is imposed. A relatively simple design approach for entropy constrained, predictive residual vector quantization (EC-PRVQ) is based on a closed-loop approach. An initial PRVQ is designed by using one of the previously mentioned techniques. The EC-PRVQ design algorithm is then applied to the system with a predetermined sequence of Lagrangian multipliers to obtain a trace of operational rate-distortion points. Details are contained in [103]–[105].

TABLE I
PERFORMANCE OF A PRVQ DESIGNED USING DIFFERENT TECHNIQUES. THE BLOCK SIZE USED IS 4×4 AND THE PEAK RATE IS 0.75 b/pixel

Image	Constrained Optimization	Joint Optimization	Closed-loop Design
"Lena"	34.0078 *dB*	32.2031 *dB*	33.9167 *dB*
"Pepper"	33.6989 *dB*	31.8153 *dB*	33.7116 *dB*
"Boat"	31.0345 *dB*	29.5283 *dB*	30.9030 *dB*
"Merie"	37.7990 *dB*	35.1960 *dB*	37.7434 *dB*

6) Predictive RVQ Performance: We now present experimental results to evaluate the relative performances of the various PRVQ design methods. A two-stage PRVQ was designed using blocks of size 4×4 with stage codebook sizes of 64, corresponding to a rate of 0.75 b/pixel. A PTSRVQ was also designed to compare its performance with that of the PRVQ. Furthermore, the equivalent (in terms of rate) PVQ, RVQ and unconstrained VQ were also designed.

Table I shows the performances of PRVQ's at a fixed rate of 0.75 b/pixel for the various design techniques. It can be seen that the PRVQ designed by the constrained optimization technique performs the best. The performance of the closed-loop technique is very close to that of the constrained optimization technique. The predictor-quantizer joint optimization technique performed poorly since the RVQ stage were not jointly optimized by that method.

Table II shows the performance of PRVQ and PTSRVQ along with the performances of PVQ, RVQ, and VQ using

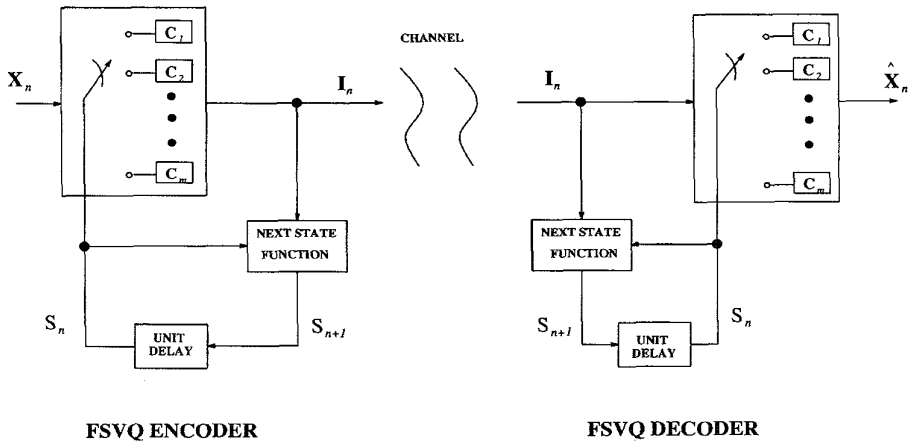


Fig. 25. FSVQ encoder and decoder.

TABLE II
PERFORMANCE COMPARISON OF PTSRVQ AND PRVQ WITH THAT OF PVQ, RVQ, AND THE UNCONSTRAINED VQ FOR THE TEST IMAGE LENA. THE PEAK RATE IS 0.75 b/pixel.

VQ Schemes	Test Image Lena		Search Complexity	Storage Requirement
	Average Bit-rate	PSNR dB		
PTSRVQ	0.5833	34.2714	160	67584
PRVQ	0.5475	34.0078	128	2048
PVQ	0.6546	34.4858	4096	65536
RVQ	0.6051	32.3110	128	2048
VQ	0.6376	32.6015	4096	65536

the test image Lena (which is outside the training set). Table II shows the average rate (entropy coded), PSNR, search complexity, and storage requirements. The search complexity is given in terms of the number of multiplications required to encode one pixel, and the storage requirement is in terms of the number of memory words required to store all the codebooks. It can be seen that both the PRVQ and PTSRVQ schemes outperform RVQ and the unconstrained VQ. PTSRVQ performs close to PVQ but with a lower search complexity (32 times lower). Furthermore, the entropies of PTSRVQ and PRVQ are lower than that of PVQ, which gives an extra saving of 0.07–0.1 b/pixel when indices are entropy coded.

The EC-PRVQ was found to give good rate-distortion performance. A training set consisting of 480 060 vectors was used to design a six-stage EC-PRVQ with a peak rate of 2.25 b/pixel. The 512 × 512 test image "Lena" was used to compare the rate-distortion performance of EC-PRVQ with that of JPEG. Huffman coding was used to encode the EC-PRVQ indices. Fig. 24 shows the performances of EC-PRVQ and JPEG in terms of PSNR versus average rate. It can be seen that at rates near 1.00 b/pixel, EC-PRVQ and JPEG give almost identical performance. As the rate decreases, EC-PRVQ performs better than JPEG. The performance gain of EC-PRVQ over JPEG is 0.65 and 3.0 dB at about 0.80 and 0.20 b/pixel, respectively. This corresponds to a 15–50% saving in rate. The Huffman coding overhead rate is about 10% at low rates. This suggests that a better rate-distortion performance can be achieved at low rates if a more sophisticated entropy coding scheme is used.

B. Finite-State RVQ

A finite-state VQ (FSVQ) has a finite number of states where a relatively small sized codebook is associated with each state [1]. The state of the encoder (and decoder) is determined by the previous state and previously encoded vectors. The input is quantized using the current state codebook. An FSVQ encoder and decoder is shown in Fig. 25, where state codebooks are denoted by $\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_m$. The basic function of an FSVQ is as follows: Given an initial state s_0, the encoder ε vector quantizes the current input vector $\mathbf{X}_n \in \Re^k$ using the current state codebook $\mathbf{C}_n$ and produces an index $\mathbf{I}_n$, that is, $\mathbf{I}_n = \varepsilon(\mathbf{X}_n, S_n)$. The next state is determined from the current encoding decision $\mathbf{I}_n$ and the current state, that is, $S_{n+1} = f(\mathbf{I}_n, S_n)$. The decoder β generates the reconstructed vector $\hat{\mathbf{X}}_n$ using the current state codebook and the index $\mathbf{I}_n$, that is, $\hat{\mathbf{X}}_n = \beta(\mathbf{I}_n, S_n)$.

In the FSVQ scheme proposed by Kim [106], the next state function is determined by classifying each of the previously encoded neighboring vectors into one of a finite number of classes. Specifically, the upper and the left blocks to the current block are chosen to determine the current state as shown in Fig. 26. In order to encode the current block $\mathbf{X}_{i,j}$ (i, j represent the location of the current block in a 2-D plane) the current state $S_{i,j}$ is given by $S_{i,j} = f[class(\hat{\mathbf{X}}_{i-1,j}), class(\hat{\mathbf{X}}_{i,j-1})]$, where $\hat{\mathbf{X}}_{i-1,j}$ and $\hat{\mathbf{X}}_{i,j-1}$ represent the previously reconstructed vectors at locations $(i-1, j)$ and $(i, j-1)$, respectively. Note that the states of the top block row and the left-most vector block are assumed to be initially known. We now discuss two schemes that combine FSVQ and RVQ to form FSRVQ.

1) Finite-State BRVQ: This scheme was developed by Kossentini *et al.* [96]. In this scheme, each state codebook of an ordinary FSVQ is replaced by a binary residual vector quantizer (BRVQ). This scheme is called finite state BRVQ (FSBRVQ). In FSBRVQ, the criterion used to determine the next state is the same as that of Kim's technique [106]. The attractive feature of this scheme is that the memory

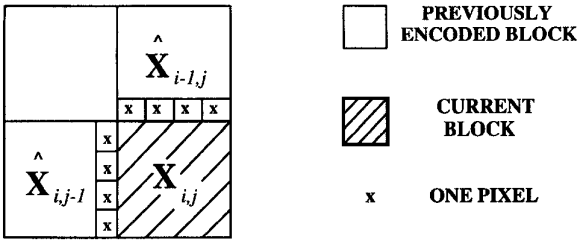


Fig. 26. Geometry of the state prediction.

requirements are reduced tremendously. Another advantage of an FSBRVQ is that a high-dimensional vector can be used due to the low computational complexity of BRVQ. However, the problem with large block size is that large blocks often contain complicated details and a simple classifier based on a single edge detection is no longer effective. The details of FSBRVQ can be found in [96]. Experimental results show that the FSBRVQ gives better performance than BRVQ; however, the performance deteriorates when compared to FSVQ at the same rate [96]. This performance degradation is due to the inherent structural constraints of the BRVQ scheme. On the other hand, memory requirements are significantly reduced.

2) Finite-State RVQ: As we discussed earlier, in PRVQ, we restrict the search complexity and memory requirements by constraining the codebook size as well as the vector dimension. Therefore, PRVQ provides a good trade-off between performance, search complexity, and memory requirement. In terms of cost of implementation, search complexity is generally more expensive than memory requirements. The PRVQ structure, however, treats both constraints equally. This implies that the performance of a PRVQ can be improved if we ease the constraint on the memory requirement. This can be achieved, for example, by incorporating a finite number of states in the PRVQ scheme. The resultant scheme is referred to as finite-state RVQ (FSRVQ) [97], [98]. FSRVQ can be viewed as an adaptive PRVQ.

The basic structure of a FSRVQ scheme is shown in Fig. 27. The scheme consists of a neural network predictor and a next-state codebook containing N_s codevectors corresponding to each of the N_s states. Each state quantizer consists of an m-stage RVQ, where the pth-stage codebook consists of N_p codevectors. This scheme differs from the conventional FSVQ in that the state-RVQ codebooks encode residual vectors instead of original vectors. A neural network predictor predicts the current block based on the four previously encoded blocks as shown in Fig. 27. The predicted vector $\tilde{\mathbf{X}}_l$ is then classified by performing a nearest neighbor search of the next-state codebook. The index of the codevector closest to $\tilde{\mathbf{X}}_l$ determines the current state. The residual vector $\mathbf{E}_l = \mathbf{X}_l - \tilde{\mathbf{X}}_l$ is then encoded using the current state-RVQ codebooks.

3) Finite-State RVQ Performance: Table III shows a performance comparison of FSRVQ and PRVQ as well as conventional PVQ. It can be seen from Table III that the rate-distortion performance of the FSRVQ scheme is superior. FSRVQ gives a 55% saving in rate with only 0.23 dB loss in the quality of the reconstructed image compared to PVQ, and the search complexity is also substantially reduced. The memory requirement of FSRVQ and PVQ are almost the same for this particular case. However, at high rates, FSRVQ requires significantly lower memory than that of PVQ. For example, at a peak rate of 2.0 b/pixel and a block size of 4×4, a PVQ would require 2^{36} bytes (approximately 68.72 GBytes) of memory, whereas the FSRVQ, with stage codebooks sizes of 16, requires 2^{15} bytes (32 KBytes), a savings of about seven orders in magnitude.

TABLE III
PERFORMANCE COMPARISON OF FSRVQ WITH THAT OF PRVQ AND PVQ, FOR THE TEST IMAGE LENA. THE PEAK RATE IS 0.625 b/pixel.

VQ Schemes	Output Entropy	PSNR dB	Search Complexity	Storage Requirement
FSRVQ	0.3716 *bpp*	33.4355	80	16640
PRVQ	0.4305 *bpp*	32.9513	64	1024
PVQ	0.5694 *bpp*	33.6570	1024	16384

The FSRVQ scheme outperforms PRVQ both in terms of rate and the quality of the reconstructed image. As mentioned earlier, in PRVQ, we constrain both the codebook search complexity and the memory requirement. However, the implementation cost of codebook search complexity is substantially higher than that of the storage requirement. Therefore, the constraint on storage requirement can be traded for an additional improvement in the quality of the reconstructed image. This is the basic philosophy behind FSRVQ: we strictly impose the constraint on the codebook search complexity and relax the constraint on the storage requirement to improve performance.

VI. SUCCESSIVE APPROXIMATION RVQ

Embedded source coding enables a decoder to extract subsets of the encoded data stream and decode useful but partial information. Refinable source coding enables an encoder to incrementally describe source data [107]. The key to achieving both embedded and refinable source coding is successive approximation data descriptions. Successive approximation is naturally obtained with multiple stage RVQ's. Previously proposed successive approximation RVQ systems that use different numbers of stages on different input vectors is reviewed, and a new type of successive approximation, variable block rate RVQ is introduced. The use of this new RVQ in image waveform coding systems is demonstrated. The use of a scalar version of this residual quantizer in image subband coding systems based on embedded wavelet transforms is also demonstrated.

A. Embedded and Refinable Source Coding

Shannon addressed the problem of *how* information can be reliably communicated [108]–[110]. However, in a lossy source coding context, there often remains unanswered the question of *what* information should be retained and communicated and *what* information should be discarded and lost. Obtaining an answer to this question is sometimes made even more difficult when the information user, and not the information source, is the best judge of what information is required. Examples of such applications include remote image database browse and access by image exploitation

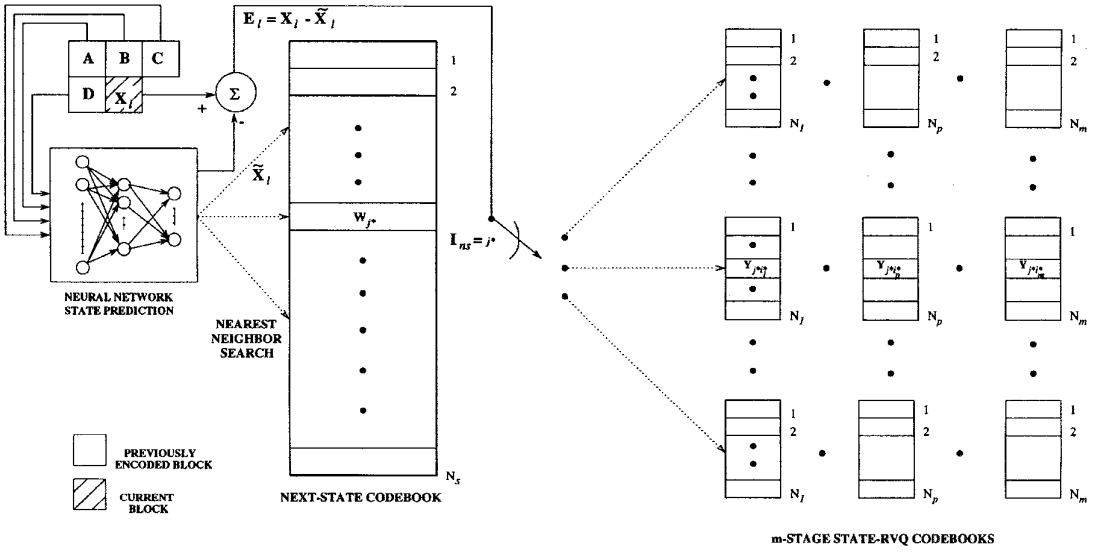


Fig. 27. FSRVQ scheme.

experts, where the channel capacity is too low to permit timely transmission of all available data. In such applications, the use of interactive progressive image transmission is an attractive solution [111]–[113]. A two-way link allows an expert user to influence early in the communication process the decision of what information is transmitted and what information is discarded, or at best, only coarsely approximated. The progressive transmission capability permits small initial amounts of coarse information to be communicated to the expert user for preliminary image evaluations. If the expert user requires higher fidelity in some image subregions, the expert makes this known via the feedback channel and the encoder provides successive refinements to the information previously transmitted.

Other applications that utilize successive approximation data representations are multiresolution broadcasting [114], embedded video coding [115], and digital television where channel degradations limit the reliability of the most refined information in the coded data stream [78], [79]. In each of these applications, the decoders must be able to extract and decode only the embedded subset of data that either has been reliably communicated through the noisy channel or that data that is compatible with receiver characteristics and terminal display formats. In a similar manner, successive approximation also helps mitigate the buffer overflow problems of variable rate systems since performance degrades gracefully as rate is reduced [116].

B. Direct Sum Successive Approximations

RVQ's have a natural successive approximation property. Whereas a single-stage vector quantizer is capable of providing a single operational rate R and associated distortion $\mathcal{D}$, a residual quantizer with P stages is capable of providing a set of operational rates $\{R_i: i = 1, 2, \cdots, P\}$ and associated distortions $\{\mathcal{D}_i\}$. Since $\mathcal{D}_i > \mathcal{D}_{i+1}$ whenever $R_i < R_{i+1}$, RVQ's provide successive approximation. However, the successive approximations provided by RVQ are not in the conventional point-wise sense [117], [118]; RVQ's provide refinements with respect to expected distortion. Although the average distortion decreases with each RVQ stage, there may exist a subset of the inputs of a RVQ stage where the decoded representation for each point in the subset degrades with the additional stage. This shortcoming does not seem so severe when one considers that Equitz and Cover proved that not all sources have (uniform-in-rate) successively refinable representations that are optimal in a rate-distortion sense [118], [119]. For a source to be theoretically successively refinable along points of the source's rate-distortion curve, each solution of the rate distortion problem must be describable as a Markov chain. Unfortunately, for finite dimensional signals, the class of successively refinable sources is rather restricted. Even the finite-dimensional Gaussian source under squared error distortion is not successively refinable. However, if asymptotically long blocks of identically distributed memoryless Gaussian random variables are considered, then Equitz and Cover have shown that successive refinement is theoretically possible. In particular, they have shown that the optimal successive approximation code achieving the rate distortion bound at each point of refinement has what Equitz and Cover call "an especially nice tree structure." Inspection of this code reveals that this "nice" tree structure is a direct sum tree structure.

One of the advantages of realizing successive approximation with RVQ is the ability to implement low-, moderate-, and high-rate codes with a single system architecture. The implementation costs of almost all other VQ structures limit their use to low or moderate rates. Hence, if one of these other VQ structures is used as the front end of an interactive progressive transmission system, then the code used to provide high rate refinements is often different in structure from the initial low rate code. The use of two different but complementary coding schemes increases the overall complexity of such a system and makes the joint design process of the low- and high-rate

TABLE IV
RVQ PERFORMANCES WHEN EMPLOYED AS EMBEDDED CODES

Total Number of Stages	Partial Number of Stages Used to Encode Moffet Image					
	2	4	8	16	32	64
2	21.13 dB 0.022673 bpp					
4	21.06 dB 0.022608 bpp	21.75 dB 0.050905 bpp				
8	20.96 dB 0.022154 bpp	21.61 dB 0.049374 bpp	22.71 dB 0.099789 bpp			
16	20.99 dB 0.022266 bpp	21.63 dB 0.049757 bpp	22.67 dB 0.099802 bpp	24.37 dB 0.213474 bpp		
32	20.99 dB 0.022266 bpp	21.63 dB 0.049757 bpp	22.66 dB 0.099771 bpp	24.23 dB 0.212315 bpp	26.91 dB 0.491703 bpp	
64	20.98 dB 0.022266 bpp	21.63 dB 0.049757 bpp	22.66 dB 0.099771 bpp	24.23 dB 0.212315 bpp	26.80 dB 0.490891 bpp	30.88 dB 1.203270

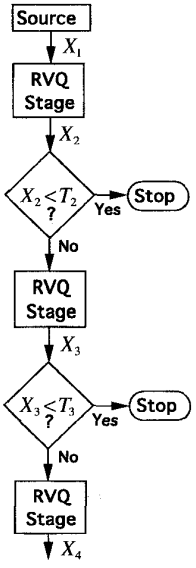


Fig. 28. Variable number of stages RVQ: Type I encoder.

components more difficult. However, the slow rate of growth of memory and computation cost of RVQ makes it possible to achieve low-, moderate-, and high-rate coding with a single RVQ structure, resulting in a consistent and simpler system architecture.

C. Variable Block Rate RVQ's

The topic addressed in this section is the successive RVQ refinement problem, where not all blocks are refined to the same extent, i.e., the variable rate successive refinement problem (as apposed to the uniform rate successive refinement problem [118]). As previously discussed, human interaction can decide the relevancy of different blocks and allocate selective refinements accordingly, but it is not desirable to always require human intervention in a coding process. Methods for achieving autonomous selective refinements are described below.

A successive approximation RVQ that uses different numbers of stages on different blocks results in a variable rate coding scheme that generally requires side information to communicate how many stages are used for each block [56], [120]–[124]. Such RVQ's are useful in coding systems that deliver a specified level of compressed image fidelity by varying the instantaneous block rate. For image coding, the

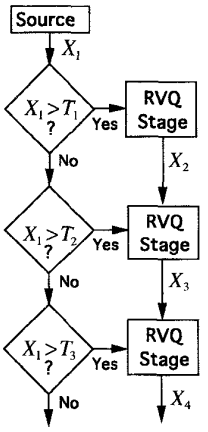


Fig. 29. Variable number of stages RVQ: Type II encoder.

idea is to use more stages on edge blocks and less stages on shade blocks. The encoder of such a variable rate system is shown in Fig. 28. This type of encoder is referred to in this paper as a Type I encoder. In a Type I encoder, the random source outputs $\boldsymbol{X}_1$ are input into the first RVQ stage. The magnitudes of the first stage residual vectors $\boldsymbol{X}_2$ are compared with a first stage threshold T_2. If a residual magnitude falls below the threshold value the distortion is deemed to be acceptably small, and the encoding process stops for that vector, but if the error magnitude exceeds the acceptance threshold T_2, then the coding process proceeds to the next stage, and so on, for a sequence of comparisons with a sequence of monotonically decreasing thresholds $T_2 > T_3 > \cdots > T_{P-1}$.

One of the design problems associated with this type of variable block rate RVQ is the problem of choosing the thresholds. The thresholds were obtained from training data in [122] and [125] and employed in such a way that side information was not required. The thresholds used in [56] and [126] were derived not from training data but from the image to be encoded. In that technique, the total energy of the input image E_{image} was determined and divided by the number of vectors $N \times N$ contained in the image. This result and a user-specified desired SNR (SNR $_{\text{desired}}$) were used to compute the threshold value $E_{\text{threshold}} = (E_{\text{image}}/N \times N) \cdot 10^{-(\text{SNR}_{\text{desired}}/10)}$. The threshold selection technique used in [123] was to first classify the input vectors according to their energy level and then have a specific SNR-derived threshold for each class. The SNR thresholds where manually manipulated to provide acceptable subjective coding performance.

Another type of refinable RVQ coding system that uses different numbers of stages on different blocks is shown in Fig. 29. This type of encoder is referred to in this paper as a Type II encoder. Instead of using a "stop encoding?" decision to effect a variable rate coding structure, the Type II encoder uses a "start encoding?" decision. For instance, as shown in Fig. 29, not all source outputs are encoded by the first RVQ stage. Only those with magnitudes that exceed the first stage threshold T_1 are encoded by the first and all subsequent stages. If a vector is not encoded by the first stage, then its magnitude is compared

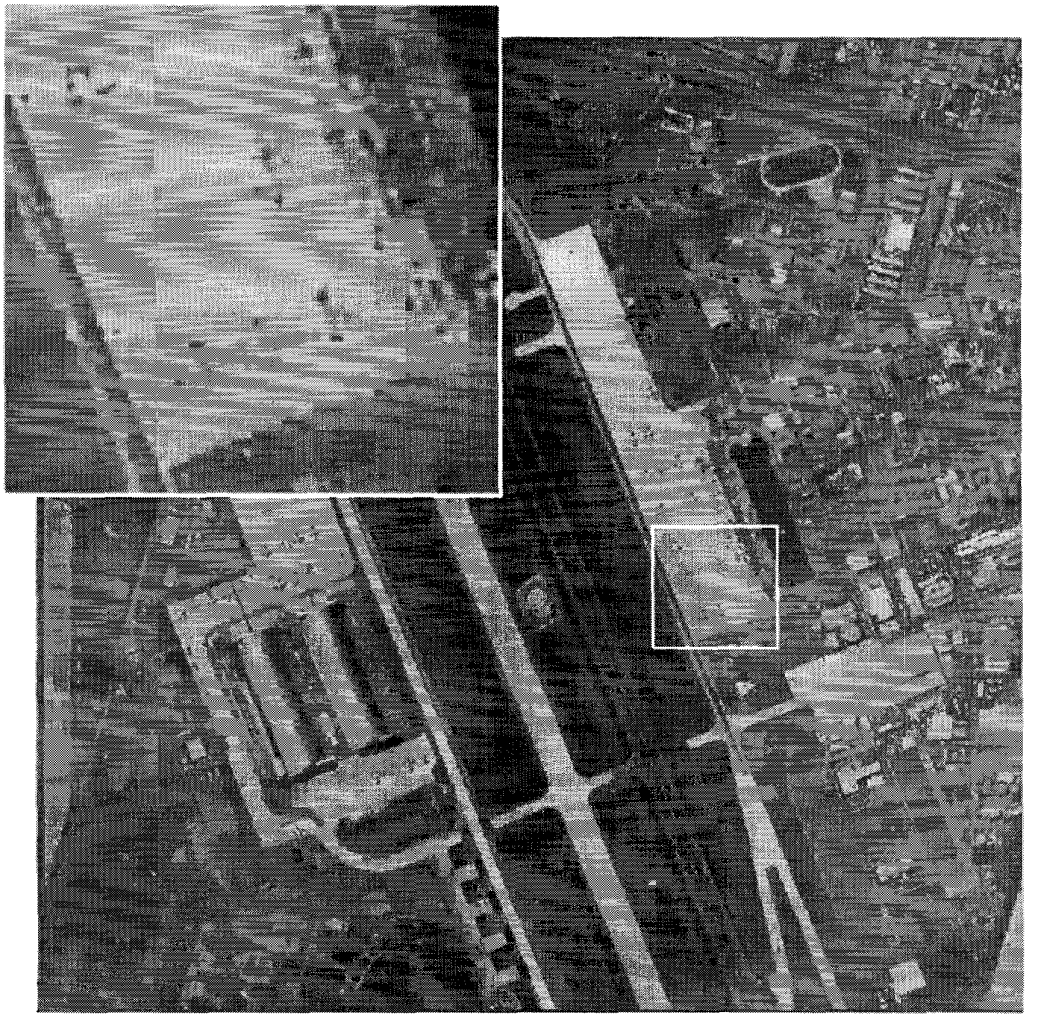


Fig. 30. Moffet image coded with a 16–stage RVQ with the variable block Type II Encoder using four codevectors per stage and 8×8 blocks. RVQ index rate = 0.151 b/pixel, number-of-stages overhead rate = 0.0625 b/pixel, codebook overhead rate = 0.125 b/pixel, PSNR = 24.34 dB.

with the second stage threshold T_2, and so on, until a threshold is found with a value less than the vector's magnitude. If the vector's magnitude is so small that it does not exceed the last stage threshold T_P, then the vector remains uncoded and is represented by the null vector at the RVQ decoder. The Type II encoder implements a vector generalization of the so-called quantizer dead zone concept [127], [128].

A rule for choosing a set of thresholds for the Type II encoder can be based on the following observation: it is reasonable to choose a pth-stage threshold T_p equal to the expected distortion level at the output of the pth stage. Only if the input energy of the vector exceeds the expected error energy of a given stage should the encoding process for the vector be commenced at that stage. Thus, the average errors at the outputs of a given set of RVQ stages can be used as set of thresholds for a Type II encoder. Moreover, empirical investigations have shown that the threshold process can be embedded into the codebook design process to permit simultaneous generation of a set of thresholds that are compatible with the associated set of codebooks. These successive approximation thresholds can be selected on the basis of either average distortion or peak distortion [119]. However, embedding the thresholding process into the design process does make design stability more precarious, but if one sets a fixed number of mandatory outer loop iterations of the interlaced fixed point design process (regardless of the lack of monotonicity that may occur during the process), a reasonably good set of initial "seed" thresholds and codebooks can be found for subsequent design iterations.

Justification for the acceptance of a Type II encoder structure can be found in the tenets of rate-distortion theory of sources with memory—this justification is the so called water filling paradigm [129]. By having a sequence of monotonically decreasing thresholds and a successive approximation quantization structure matched to the thresholds, successive approximation that mimics this paradigm can be achieved. The next section demonstrates the use of this encoding system on mean-removed blocks of image waveforms, but first, as a side note, we observe that these variable block rate encoders can also employ variable rate entropy coding techniques, and

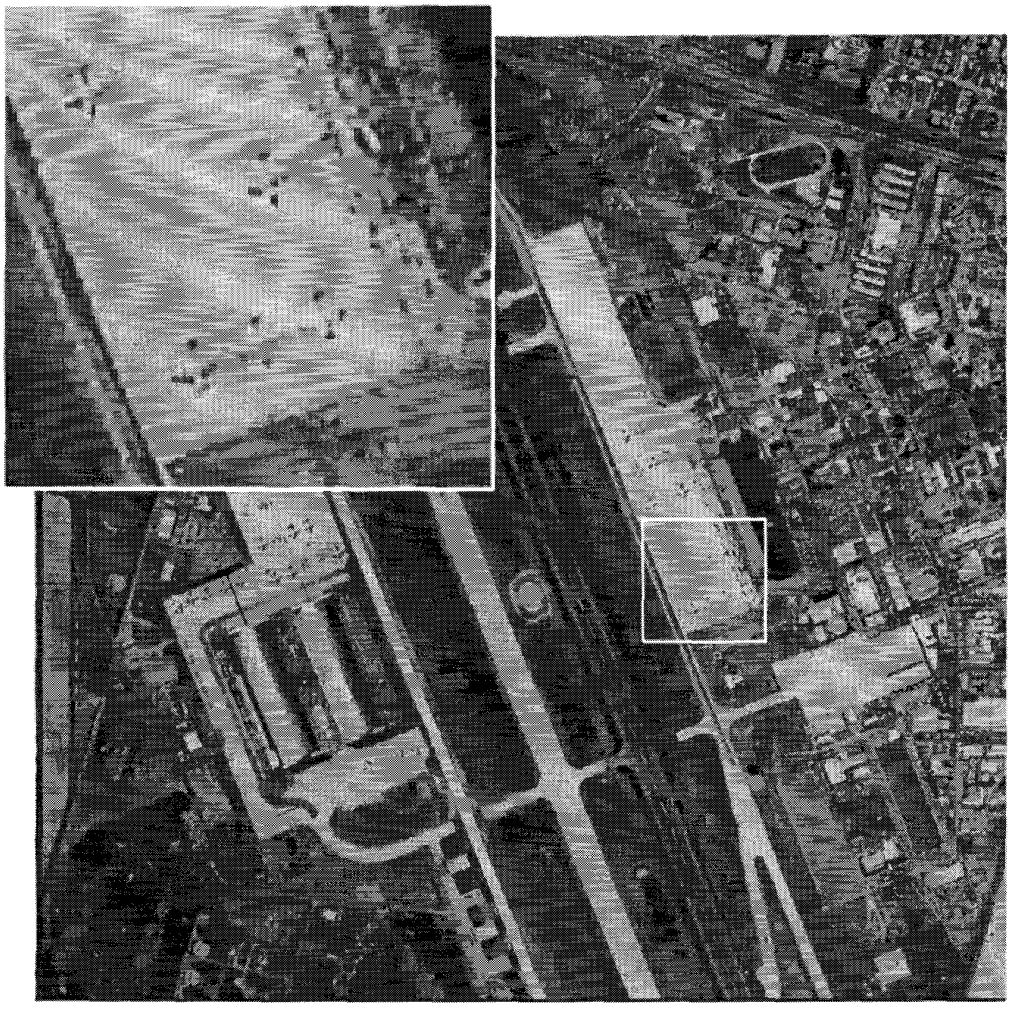


Fig. 31. Moffet image coded with a 32–stage RVQ with the variable block Type II Encoder using four codevectors per stage and 8×8 blocks. RVQ index rate = 0.414 b/pixel, number-of-stages overhead rate = 0.0781 b/pixel, codebook overhead rate = 0.25 b/pixel, PSNR = 26.91 dB.

moreover, it is conceivable that the Type I and II encoders can be combined into a single coding system.

D. Successive Approximation RVQ Image Waveform Coding

Figs. 30–32 demonstrate the performance of the variable block rate Type II encoder on mean-removed, image waveform data. Note that the shade blocks have been allocated low rates, and the detail blocks allocated higher rates—all through the use of the thresholds generated by the design process. The variable block rate RVQ used in Fig. 30 has 16 stages with four codevectors in each stage. The RVQ index rate is 0.15 b/pixel, the number-of-stages side information rate is 0.0625 b/pixel, the codebook overhead rate for image-adaptation is 0.125 b/pixel (using 32-b single precision formats for transmitted codevector elements), and the PSNR is 24.37 dB. The variable block rate RVQ used in Fig. 31 has 32 stages with four codevectors in each stage. The RVQ index rate is 0.41 b/pixel, the number-of-stages side information rate is 0.0781 b/pixel, the codebook overhead rate is 0.25 b/pixel, and the PSNR is 26.91 dB. The variable block rate RVQ used in Fig. 32 has 64 stages with four codevectors in each stage. The RVQ index rate is 1.1 b/pixel, the number-of-stages side information rate is 0.0938 b/pixel, the codebook overhead rate is 0.50 b/pixel, and the PSNR is 30.88 dB.

It has been suggested that because joint design methods inevitably increase the distortion of the intermediate stages of RVQ, even though the overall error is reduced, RVQ's are not well suited to applications where the stage codebooks are employed as embedded or multiresolution quantizers [33]. However, experiments have shown that the "embedded code losses" of jointly optimal direct sum codebooks are usually slight. For example, Table IV contains the PSNR and rate performances of each of the RVQ's designed for the Figs. 30–32 when subsets of the encoded data stream are extracted for lower rate decoding. As can be seen, the PSNR degradations are typically small, and just like the entropy coded systems, a small reduction in rate typically compensates for the distortion increase. However, it is true that to guarantee the best possible successive refinements

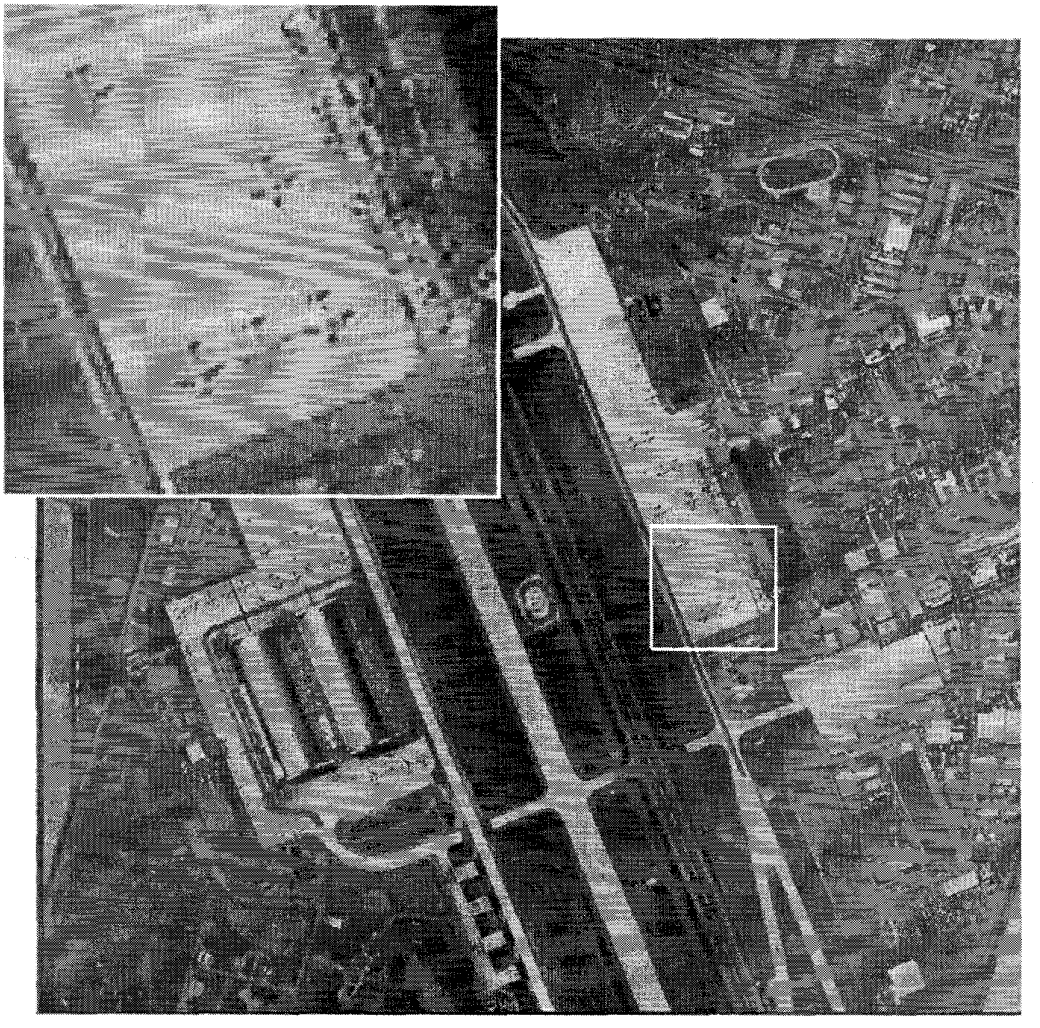


Fig. 32. · Moffet image coded with a 64–stage RVQ with the variable block Type II Encoder using four codevectors per stage and 8 × 8 blocks. RVQ index rate = 1.110 b/pixel, number-of-stages overhead rate = 0.0938 b/pixel, codebook overhead rate = 0.50 b/pixel, PSNR = 30.88 dB.

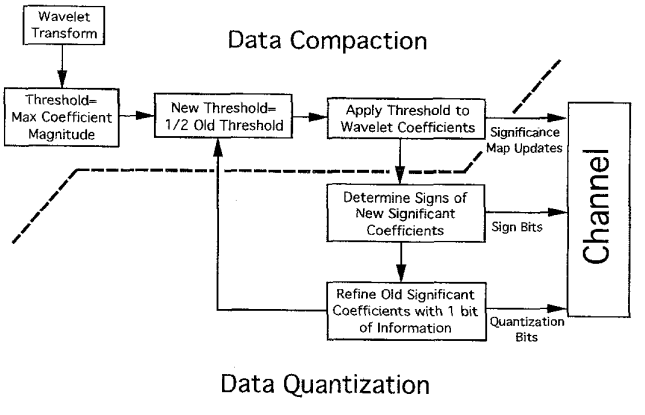


Fig. 33. Block diagram of the embedded wavelet encoder.

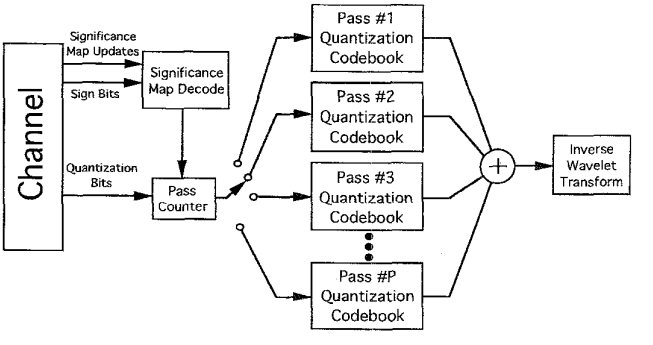


Fig. 34. Block diagram of the embedded wavelet decoder.

with direct sum decoder representations, multiple sets of jointly optimized direct sum decoder codebooks with different numbers of stages are required at the receiver (since the joint optimization process depends on the number of stages used at each point of successive refinement).

E. Successive Approximation RSQ Image Subband Coding

Multistage VQ subband coding first appeared in [21]; since then, many have combined RVQ and subband coding, partic-

Fig. 35. Lena image coded with seven passes of an embedded wavelet code using a jointly optimal RSQ. Rate = 0.0856 b/pixel (including RSQ codebook overhead), PSNR = 29.67 dB.

ularly with the use of wavelet transforms [88], [130]–[134]. There seems to be a natural merger of the successive approximation properties of RVQ's and the multiresolution characteristics of wavelet subband decompositions.

One of the more effective subband coding systems is the relatively simple embedded wavelet image coder developed by Shapiro [135], [136]. This is the subband coding system discussed in the remainder of this section. Two key components of Shapiro's system are the use of a significance map and progressive transmission of successive bit planes of significant coefficients. Shapiro's quantization mechanism is the use of a scaled successive approximation uniform scalar quantizer. This section demonstrates that the performance of embedded wavelet coders can be improved with the use of jointly optimized, successive approximation residual scalar quantizers. Future research will investigate the use of residual vector quantizers in these subband coding systems.

1) Direct Sum Formulation of Embedded Wavelet Coding: A block diagram of the embedded wavelet encoder is shown in Fig. 33. One of the primary functions of the wavelet transform in an embedded wavelet coder is data compaction. Data compaction is the process of reducing the number of samples necessary to represent a sampled waveform (in this

TABLE V
PERFORMANCE COMPARISONS OF THE EMBEDDED WAVELET CODE WITH AND WITHOUT JOINTLY OPTIMAL DIRECT SUM SCALAR QUANTIZERS ON THE LENA IMAGE (POSTED TOTAL RATE INCLUDES RSQ CODEBOOK OVERHEAD RATE)

Pass Count	Rate (bpp)	Overhead Rate	Total Rate	PSNR w/o RSQ	PSNR w/ RSQ	PSNR Gain
2	0.000977	0.000977	0.001953	16.86	17.05	0.19
3	0.002289	0.001465	0.003754	18.72	18.89	0.17
4	0.005615	0.001953	0.007568	21.30	21.52	0.22
5	0.013733	0.002441	0.016174	23.61	23.76	0.15
6	0.035614	0.002930	0.038544	26.46	26.65	0.19
7	0.082153	0.003418	0.085571	29.46	29.67	0.21
8	0.179321	0.003906	0.183228	32.70	32.94	0.24
9	0.365814	0.004395	0.370209	35.88	36.14	0.26
10	0.750854	0.004883	0.755737	39.10	39.37	0.27
11	1.603577	0.005371	1.608948	43.40	43.86	0.46
12	2.803802	0.005860	2.809662	49.25	50.17	0.92
13	4.003906	0.006348	4.010254	55.79	58.22	2.43
14	5.124878	0.006684	5.131714	73.22	79.48	6.26
15	6.188293	0.007324	6.195618	∞	∞	–

Fig. 36. Lena image coded with eight passes of an embedded wavelet code using a jointly optimal RSQ. Rate = 0.1832 b/pixel (including RSQ codebook overhead), PSNR = 32.94 dB.

case, a 2-D image). The embedded wavelet coder uses the wavelet transform and an iterated thresholding process for data compaction. Initially, the magnitude of the largest wavelet coefficient is determined and the first threshold is set to one half of this value. All wavelet coefficients are compared with this threshold. If a coefficient's magnitude exceeds the threshold it is deemed significant, otherwise, the coefficient is viewed as insignificant and is treated (at least temporarily) as a zero value. The locations of all significant coefficients in the wavelet space-frequency plane for a given threshold constitute the significance map. A significance map can be effectively represented by one of several methods. Shapiro uses a combination of symbols that represent zerotree roots, isolated zeros, and the signs of significant coefficients. Said and Pearlman have developed a more efficient means of communicating the significance map [137].

After communicating significance map information, data that represents the magnitudes of the significant coefficients are communicated in successive iterations of the embedded wavelet coder. Although one may transmit these magnitudes with perfect amplitude information in a single pass [138], it makes little sense to simultaneously have a coarse spatial representation generated by the compaction process, and a fine amplitude representation generated by a high rate quantizer. Rather, experiments have shown that it is more appropriate to have initially coarse spatial information and coarse amplitude information, and then iteratively refine both the spatial and spectral descriptions. Refinements are made in the embedded wavelet coder when the threshold is reduced by a factor of two, new significant coefficients are identified to refine the spatial information, and additional bit planes are transmitted to refine the spectral information. This combined process of successive approximation with respect to both spatial and spectral characteristics is likely the main reason for the effectiveness of the embedded wavelet coder.

A block diagram of the embedded wavelet decoder is shown in Fig. 34. The significance map information is decoded and used to control the positioning of successive approximation quantization bits on each refinement pass of the embedded wavelet coder. This block diagram makes the direct sum struc-

Fig. 37. Lena image coded with nine passes of an embedded wavelet code using a jointly optimal RSQ. Rate = 0.3702 b/pixel (including RSQ codebook overhead), PSNR = 36.14 dB.

ture of the quantization mechanism of the embedded wavelet coder readily apparent. With the direct sum structure made explicit, it is possible to use the joint optimization techniques of residual quantizers to improve the performance of the embedded wavelet coder. The following sections describe the various possibilities for exploiting a direct sum formulation of embedded wavelet coding to enhance performance.

2) Generalized Embedded Wavelet Coding: In Shapiro's system, the thresholding operation used for data compaction is intimately related to the partition thresholds used for data quantization. The separation of these two thresholding functions permits greater design freedom in controlling spatial and spectral successive approximation operations. With independent thresholding operations, the compaction threshold can be adjusted to control the rate at which additional coefficients become significant, and various direct sum encoder structures can be used for the successive approximation quantization operation.

There are at least two ways of generalizing the direct sum structures of uniform successive approximation quantizers. The first retains the conventional successive approximation uniform quantizer encoder structure and employs a direct sum decoder structure that is jointly optimal with respect to the source probability density function and the partition of the quantizer's input space induced by the encoder. This approach has the advantage that the generalized decoder is compatible with the bit stream generated by the uniform successive approximation quantizer encoder [61]. The second approach seeks jointly optimal encoder and decoder direct sum structures, where compatibility with the standard encoder bit stream is sacrificed for the performance improvement obtained by simultaneously generalizing the encoder and decoder structures. The following describe these generalizations in more detail.

One formulation of a uniform scalar quantizer is in terms of a successive approximation tree structure [1]. A uniform scalar quantizers with 2^P output values can be described in terms of a P-stage direct sum structure. For example, the P-b binary encoder of a uniform scalar quantizer maps a value x into a P-tuple binary codeword $\boldsymbol{i}$. Standard binary coding is

Fig. 38. Original 512 × 512 Lena image.

such that a representation $\hat{x}$ of the source sample x is formed by the expression

$$\hat{x} = \sum_{p=1}^{P} i_p \cdot 2^{P-p} \tag{18}$$

where $i_p \in \{0, 1\}$ is the pth element of $\boldsymbol{i}$. By changing notation, the direct sum structure of this standard binary code is readily apparent. Let $y_p(i_p) = i_p 2^{P-p}$, and hence, (18) can be rewritten

$$\hat{x} = \sum_{p=1}^{P} y_p(i_p). \tag{19}$$

If $C_p = \{y_p(0), y_p(1)\}$ comprises a pth-stage bit plane codebook, then the set of all possible values of $\hat{x}$ is the direct sum set $C = C_1 + \cdots + C_P$.

An important question is whether or not the standard set of bit plane direct sum constituent sets $\{0, 2^{P-p}\}$ generates the best representation for a particular source distribution. If the source is uniformly distributed, the answer is yes, but, in general, for nonuniform sources, better performance can be obtained by using joint design techniques for the direct sum codebooks. Furthermore, the variable block rate RVQ encoder Type II can be employed with the associated set of thresholds used for significance map determination. The thresholds of the Type II encoder provide an answer to the rate allocation problem of embedded wavelet encoder that has been addressed in various ways [140]–[143]. Figs. 35–38 show the performances of jointly optimal RSQ's in embedded wavelet transform coding systems that use the significance map encoding procedures of [137]. Table V shows the RSQ performance gains (including RSQ codebook overhead rates) as a function of the number of complete bit planes used in the successive approximation process. The PSNR gain values range from a couple of tenths of a decibel at low rates to several decibels at high rates. These gains are achieved by image-dependent adaptation of the thresholding and successive approximation quantization processes.

6) RVQ and LVQ: The standard binary codes typified by (18) can be considered as a subset of the 1-D integer lattice, and as shown in the previous section, the standard binary code has a formulation in term of a direct sum structure. In general,

all lattices of any dimension have a formulation in terms of direct sum structures, and certain finite subsets of lattices also have direct sum decompositions [34], [144]. Future research on the relationship between lattice-based quantizers and direct sum quantizers may focus on the following points. Lattice vector quantizers (LVQ's) have efficient encoding structures [45], [145] but cannot be optimized with respect to the source pdf (except for scaling operations) without violating their group closure properties. RVQ's, on the other hand, may be pdf optimized but do not necessarily have the fast and optimal encoding procedures of LVQ's. Future research may show that a merger of RVQ/LVQ concepts may lead to pdf-optimized code structures that also have fast, optimal encoders. Furthermore, it has been conjectured by Gersho that optimal, high-resolution, entropy-constrained VQ's have the form of a lattice [146]. If so, then these optimal high-resolution entropy constrained VQ's have formulations in terms of entropy-constrained RVQ's, which in turn, may lead to practical implementations.

VII. Summary

This paper has provided a tutorial review of residual vector quantization and has surveyed recent advances in RVQ image coding techniques. Entropy coding the RVQ output was shown to partially compensate for distortion increases due to both the imposed direct sum structural constraint and to sequential search inefficiencies. A previously proposed variable block rate, successive approximation RVQ was reviewed, and a new type of variable block rate encoder structure was introduced. The latter encoder structure has a rate control mechanism that follows closely the encoding paradigm suggested by rate-distortion bound descriptions of Gaussian sources with memory. The performance of this new quantizer was demonstrated on image waveform data, and a scalar version of the new system was shown to provide good performance in generalized embedded wavelet transform coding systems. The notions of joint encoder and joint decoder optimality were reexamined, and the approximation process often employed in RVQ encoder design was illuminated. The development of a truly jointly optimal design method for computationally efficient, sequential search encoders was shown to be an outstanding problem; nearly all so called "jointly optimal RVQ's" are only approximately so. A design method for RVQ was described that is capable of providing separate encoder and decoder codebooks but where the two sets of codebooks work together to provide good overall performance. The design method generates encoders with acceptable sequential search performance and generates decoders where all stage codebooks simultaneously satisfy necessary conditions for joint optimality. Particular attention was paid to the design of RVQ's with many stages. Although most of the implementation efficiency of an RVQ over an exhaustive search VQ is obtained in going from one stage to two stages, there are applications where the use of RVQ's with more than two stages is advantageous. RVQ's designed for combined channel/source coding are more robust [147], [148], adaptive RVQ's more efficiently communicate overhead codebook information [18], variable block rate RVQ's have more precise rate control [123], and conditional entropy-constrained RVQ's more easily manage memory costs [89]. In spite of these observations, the research community has, in general, only side stepped the problem of designing RVQ's with many stages, preferring instead to accept the performance levels obtainable with conventional design methods when only two or three stages are used. Thus, the community has not yet encountered many of the problems associated with the design of RVQ's with many stages and may not yet fully appreciate the value of solutions which mitigate these design difficulties. Nevertheless, the authors feel that as time goes along, interest in RVQ's with many stages will increase, and researchers will find the techniques of this paper of value.

References

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston: Kluwer, 1992.

[2] N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.*, vol. 36, no. 8, pp. 957–971, Aug. 1988.

[3] W.-Y. Chan and A. Gersho, "Generalized product code vector quantization: A family of efficient techniques for signal compression," *Digital Signal Processing.* New York: Academic, vol. 4, pp. 95–126, 1994.

[4] R. L. Baker and R. M. Gray, "Differential vector quantization of achromatic imagery," in *Proc. Int. Picture Coding Symp.*, Mar. 1983, pp. 105–106.

[5] M. J. Sabin and R. M. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 3, pp. 474–488, June 1984.

[6] K. L. Oehler and R. M. Gray, "Mean-gain-shape vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Minneapolis, MN, USA, vol. V, Apr. 27–30, 1993, pp. 241–244.

[7] W.-Y. Chan and A. Gersho, "Enhanced multistage vector quantization with constrained storage," in *Proc. Twenty-Fourth Asilomar Conf. Signals, Syst., Comput.*, Nov. 1990, pp. 659–663.

[8] C. F. Barnes and R. L. Frost, "Vector quantizers with direct sum codebooks," *IEEE Trans. Inform. Theory*, vol. 39, no. 2, pp. 565–580, Mar. 1993.

[9] B. H. Juang and A. H. Gray, "Multiple stage vector quantization for speech coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, Apr. 1982, pp. 597–600.

[10] R. M. Gray, "Vector quantization," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 1, pp. 4–29, Apr. 1984.

[11] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985.

[12] L. Wang and M. Goldberg, "Progressive image transmission by transform coefficient residual error quantization," *IEEE Trans. Commun.*, vol. 36, no. 1, pp. 75–87, Jan. 1988.

[13] Y. Yamada and S. Tazaki, "Recursive vector quantization for monochrome video signals," *IEICE Trans.*, vol. E–74, no. 2, pp. 399–405, Feb. 1991.

[14] J. Pan and T. R. Fischer, "Two-stage vector quantization-lattice vector quantization," *IEEE Trans. Inform. Theory*, vol. 41, no. 1, pp. 155–163, Jan. 1995.

[15] P. F. Swaszek, "Unrestricted multistage vector quantizers," *IEEE Trans. Inform. Theory*, vol. 38, no. 3, pp. 1169–1174, May 1992.

[16] C. F. Barnes and R. L. Frost, "Residual vector quantizers with jointly optimized code books," in *Image Mathematics and Image Processing*, vol. 84 of *Advances in Electronics and Electron Physics*, P. W. Hawkes, Ed. New York: Academic, 1992, pp. 1–59.

[17] J. Salillas and N. Farvardin, "Adaptive multi-stage vector quantization of images," in *Signal Processing VI-Theories Applications, Proc. EUSIPCO-92, Sixth Euro. Signal Processing Conf.*, Brussels, Belgium, vol. 3, Aug. 24–27, 1992, pp. 1239–1242.

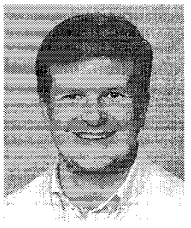
[18] C. F. Barnes, "Adaptive successive approximation quantization of image waveforms with efficient code-book updates," in *Proc. IEEE Int. Conf. Image Processing*, vol. III, Nov. 1994, pp. 876–880.

[19] ——, "An initial evaluation of a new approach to high resolution sonar imagery," Georgia Tech Res. Inst., Atlanta, GA, GTRI Final Tech. Rep. Project no. A-9523, Nov. 1994.

[20] D. J. Vaisey and A. Gersho, "Variable block-size image coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,*, Apr. 1987, pp. 1051–1054
[21] X. Ran and N. Farvardin, "Combined VQ-DCT coding of images using interblock noiseless coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Apr. 1990, pp. 2281–2284.
[22] C. F. Barnes, "A new multiple path search technique of residual vector quantizers," in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 29–31, 1994, pp. 42–51.
[23] H. O. Adeyemi and C. F. Barnes, "Adaptive RVQ image coding of FLIR video," in *Proc. Twenty-Eighth Asilomar Conf. Signals, Syst. Comput.,* vol. 1, Nov. 1–3, 1994.
[24] C. F. Barnes and J. P. Watkins, "Embedded wavelet zerotree coding with direct sun quantization structures," in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 28–30, 1995, pp. 252–261.
[25] S. A. Rizvi and N. M. Nasrabadi, "Multistage vector quantization using competitive neural networks," in *Proc. SPIE Visual Commun. Image Processing,* Chicago, IL, Sept. 25–28, 1994, pp. 150–164.
[26] ——, "Residual vector quantization using a multi-layer competitive neural network," *IEEE J. Selected Areas Commun.,* vol. 12, no. 9, pp. 1452–1459, Dec. 1994.
[27] R. L. Baker, "Vector quantization of digital images," Ph.D. thesis, Stanford Univ., Stanford, CA, USA, June 1984.
[28] C. F. Barnes and R. L. Frost, "Necessary conditions for the optimality of residual vector quantizers," in *Abstracts IEEE Int. Symp. Inform. Theory,* San Diego, CA, Jan. 14–19, 1990, p. 34.
[29] S. P. Lloyd, "Least squares quantization in PCM," Bell Labs. Tech. Note, 1957, published in the Special Issue on Quantization of the *IEEE Trans. Inform. Theory, Part 1,* vol. 28, pp. 129–137, Mar. 1982.
[30] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.,* vol. COM-28, no. 1, pp. 84–95, Jan. 1980.
[31] D. H. Lee, "Asymptotic quantization error and cell-conditioned two-stage vector quantization," Ph.D. thesis, Univ. Michigan, Ann Arbor, USA, 1990.
[32] D. H. Lee, D. L. Neuhoff, and K. K. Paliwal, "Cell-conditioned multistage vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* 1991, pp. 653–656.
[33] W.-Y. Chan, S. Gupta, and A. Gersho, "Enhanced multistage vector quantization by joint codebook design," *IEEE Trans. Commun.,* vol. 40, no. 11, pp. 1693–1697, Nov. 1992.
[34] C. F. Barnes, "Tree structured signal space codes," *Coding and Quantization,* R. Calderbank, G. D. Forney, Jr, and N. Moayeri, Eds. New York: Amer. Math. Soc., 1992, vol. 14, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pp. 33–53.
[35] US Patent 5 250 949: "System and method for data compression using multiple codewords and transmitted indices," filed Aug. 14, 1989, issued Oct. 5, 1993.
[36] W.-Y. Chan, "The design of generalized product-code vector quantizers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* San Francisco, CA, vol. 3, Mar. 23–26, 1992, pp. 389–392.
[37] T. Fine, "Properties of an optimum digital system and applications," *IEEE Trans. Inform. Theory,* pp. 287–296, 1964.
[38] J. D. Gibson and T. R. Fischer, "Alphabet-constrained data compression," *IEEE Trans. Inform. Theory,* pp. 443–457, May 1982.
[39] J. G. Dunham, "An iterative theory for code design," in *Proc. IEEE Int. Symp. Inform. Theory, Abstracts Papers,* Sept. 1983, pp. 89–90.
[40] R. M. Gray, J. C. Kieffer, and Y. Linde, "Locally optimal block quantizer design," *Information and Control,* vol. 45, no. 2, pp. 178–198, May 1980.
[41] W.-Y. Chan and A. Gersho, "Joint codebook design for summation product-code vector quantizer," in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 30–Apr. 2, 1993, pp. 42–51.
[42] W.-Y. Chan, "Product code vector quantization methods with application to high fidelity audio coding," Ph.D. thesis, Univ. of California, Santa Barbara, Dec. 1991.
[43] C. F. Barnes, "Residual quantizers," Ph.D. thesis, Brigham Young Univ., Provo, UT, Dec. 1989.
[44] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Necessary conditions for the optimality of variable rate residual vector quantizers," *IEEE Trans. Information Theory,* vol 41, no. 6, pp. 1903–1914, Nov. 1995.
[45] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups.* New York: Springer-Verlag, 1988.
[46] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory,* vol. 6, no. 2, pp. 7–12, Mar. 1960.
[47] F. Jelinek and J. B. Anderson, "Instrumental tree encoding of information sources," *IEEE Trans. Inform. Theory,* vol. 17, pp. 118–119, Jan. 1971.
[48] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.,* vol. COM–32, pp. 169–176, Feb. 1984.
[49] E. Paksoy, W.-Y. Chan, and A. Gersho, "Vector quantization of speech LSF parameters with generalized product codes," in *Proc. Int. Conf. Spoken Language Processing,* Oct. 1992, pp. 33–36.
[50] B. Bhattacharya, W. LeBlanc, S. Mahmound, and V. Cuperman, "Tree searched multi-stage vector quantization on LPC parameters for 4 kb/s speech coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* San Francisco, CA, vol. 1, Mar. 23–26, 1992, pp. 105–108.
[51] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Large block RVQ with multipath searching," in *Proc. IEEE Int. Symp. Circuits Syst.,* San Diego, CA, vol. 5, May 10–13, 1992, pp. 2276–2279.
[52] T. Miyano, M. Serizawa, J. Takizawa, S. Ikeda, and K. Ozawa, "Improved 4.8 kb/s CELP coding using two-stage vector quantization with multiple candidates," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* San Francisco, CA, vol. 1, Mar. 23–26, 1992, pp. 321–324.
[53] W. P. LeBlanc, B. Bhattacharya, S. A. Mahmound, and V. Cuperman, "Efficient search and design procedures for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech and Audio Processing,* vol. 1, no. 4, pp. 373–385, Oct. 1993.
[54] F. Kossentini and M. J. T. Smith, "A fast searching technique for multistage residual vector quantization," *IEEE Signal Processing Lett.,* vol. 1, no. 7, pp. 114–116, July 1994.
[55] M. Lightstone, D. Miller, and S. K. Mitra, "Entropy-constrained product code vector quantization with application to image coding," in *Proc. IEEE Int. Conf. Image Processing,* vol. I, Nov. 1994, pp. 623–627.
[56] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Image coding with variable rate RVQ," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* San Francisco, CA, vol. 3, Mar. 23–26, 1992, pp. 369–372.
[57] ——, "Residual VQ with state prediction: A new method for image coding," *SPIE/IST Symp. Electron. Imaging: Sci. Technol.,* Feb. 24–Mar. 1, 1991.
[58] ——, "A perspective view of finite state binary residual VQ," in *1991 Int. Symp. Circuits Syst.,* June 11–14, 1991, pp. 300–303.
[59] W.-Y. Chan and A. Gersho, "Constrained-storage quantization of multiple vector sources by codebook sharing," *IEEE Trans. Commun.,* vol. 39, no. 1, pp. 11–13, Jan. 1991.
[60] D. Miller and K. Rose, "An improved sequential search multistage vector quantizer," in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 30–Apr. 2, 1993, pp. 12–21.
[61] C. F. Barnes and E. J. Holder, "Successive approximation quantization with generalized decoding for wavelet transform image coding," in *Proc. Twenty-Seventh Asilomar Conf. Signals, Syst., Comput.,* vol. 1, Nov. 1993, pp. 533–537.
[62] K. Rose, E. Gurewitz, and G. C. Fox, "Vector quantization by deterministic annealing," *IEEE Trans. Inform. Theory,* vol. 38, no. 4, pp. 1249–1258, July 1992.
[63] M. B. Priestly, *Spectral Analysis and Time Series.* San Diego, CA: Academic, 1989.
[64] M. Lightstone and S. K. Mitra, "Adaptive vector quantization for image coding in an entropy-constrained framework," in *Proc. IEEE Int. Conf. Image Processing,* vol. I, Nov. 1994, pp. 618–622.
[65] J. A. Rodriguez-Fonollosa and E. Masgrau, "Adaptive multistage vector quantization," in *Proc. IEEE Processing MELECON,* Apr. 1990, pp. 225–228.
[66] C.-D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Trans. Commun.,* vol. 33, no. 10, pp. 1132–1133, Oct. 1985.
[67] M. Venkatraman and N. M. Nasrabadi, "Constrained gradient descent algorithm for residual vector quantization," in *Proc. IEEE Int. Conf. Image Processing,* Austin, TX, Nov. 13–16, 1994, pp. 124–128.
[68] S. A. Rizvi and N. M. Nasrabadi, "Predictive vector quantization using constrained optimization," *IEEE Signal Processing Lett.,* vol. 1, no. 1, pp. 15–18, Jan. 1994.
[69] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Finite-state residual vector quantization," *J. Visual Commun. Image Represent.,* vol. 5, no. 1, pp. 75–87, Mar. 1994.
[70] N. Farvardin and J. W. Modestino, "Adaptive buffer-instrumented entropy-coded quantizer performance for memoryless sources," *IEEE Trans. Inform. Theory,* vol. 32, no. 1, pp. 9–22, Jan. 1986.
[71] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, no. 1, pp. 31–42, Jan. 1989.
[72] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Entropy-constrained residual vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Minneapolis, MN, vol. V, Apr. 1993, pp. 598–601.
[73] P. A. Chou, "Application of entropy-constrained vector quantization to

waveform coding of images," in *Proc. SPIE Symp. Visual Commun. Image Processing,* vol. 1199, 1989, pp. 970–978.
[74] T. D. Lookabaugh, E. A. Riskin, P. A. Chou, and R. M. Gray, "Variable rate vector quantization for speech, image, and video compression," *IEEE Trans. Commun.,* vol. 41, no. 1, pp. 186–199, Jan. 1993.
[75] R. P. Rao and W. A. Pearlman, "Alphabet-constrained vector quantization," *IEEE Trans. Inform. Theory,* vol. 39, no. 4, pp. 1167–1179, Mar. 1993.
[76] F. Nesci, F. Kossentini, and M. J. T. Smith, "Lapped RVQ and alphabet and entropy constraints," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,* 1994, pp. 621–624.
[77] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Image coding using entropy-constrained RVQ," *IEEE Trans. Image Processing,* vol. 4, no. 5, pp. 1349–1357, Oct. 1995.
[78] J. W. Woods and T. Naveen, "Motion compensated multiresolution transmission of HD video using multistage quantizers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Minneapolis, MN, vol. V, Apr. 27–30, pp. 582–585.
[79] F. Bosveld, R. L. Lagendijk, and J. Biemond, "Compatible HDTV transmission using conditional entropy coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* vol. V, 1993, pp. 674–677.
[80] M. Lei, T.-C. Chen, and K.-H. Tzou, "Subband HDTV coding using high-order conditional statistics," *IEEE J. Selected Areas Commun.,* vol. 11, no. 1, pp. 65–76, Jan. 1993.
[81] F. Kossentini, W. C. Chung, and M. J. T. Smith, "Subband image coding with optimal intra- and inter-band subband quantization," in *Proc. 27th Asilomar Conf. Signals, Syst., Comput.*, pp. 876–879, 1993.
[82] ——, "Application of entropy-constrained RVQ to coding image subbands," in *Proc. Int. Symp. Circuits Syst.,* 1993, pp. 683–686.
[83] X. Yuan and V. K. Ingle, "A conditional entropy-coded multi-stage vector quantizer," in *Proc. Sixth Multidimensional Signal Processing Workshop,* Pacific Grove, CA, Sept. 6–8, 1989, pp. 232–233.
[84] R. A. Cohen and J. W. Woods, "Entropy-constrained SBPVQ for image coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* 1990, pp. 2269–2271.
[85] D. P. Garrido and W. A. Pearlman, "Conditional entropy-constrained vector quantization: High-rate theory and design algorithms," *IEEE Trans. Inform. Theory,* vol. 41, no. 4, pp. 901–916, 1995.
[86] P. A. Chou and T. Lookabough, "Conditional entropy-constrained vector quantization of linear predictive coefficients," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* vol. 4, 1990, pp. 197–200.
[87] F. Kossentini, W. C. Chung, and M. J. T. Smith, "Image coding using high-order conditional entropy-constrained residual VQ," in *Proc. IEEE Int. Conf. Image Processing,* 1994, pp. 613–617.
[88] ——, "Subband image coding using entropy-constrained residual vector quantization," *Inform. Processing Management,* vol. 30, no. 6, pp. 887–896, 1994.
[89] ——, "Subband image coding with jointly optimized quantizers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* May 1995, pp. 2221–2224.
[90] W. C. Chung, F. Kossentini, and M. J. T. Smith, "A new approach to scalable video coding," J. A. Storer and M. Cohn, Eds., in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 1995, pp. 381–390.
[91] A. Scales, W. Roark, F. Kossentini, and M. J. T. Smith, "Lossless compression using conditional entropy-constrained subband quantization," in *JPL Pub. 95-8: Proc. Space Earth Sci. Data Compression Workshop,* Mar. 27, 1995, pp. 35–43.
[92] A. Docef, F. Kossentini, W. Chung, and M. J. T. Smith, "Multiplication-free subband coding of color images," in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 28–30, 1995, pp. 352–359.
[93] F. Kossentini, M. J. T. Smith, and A. Scales, "High order entropy-constrained residual VQ for lossless compression of images," in *Proc. ISCAS,* Seattle, WA, 1995.
[94] S. A. Rizvi and N. M. Nasrabadi, "Predictive residual vector quantization," in *Proc. IEEE Int. Conference on Image Processing,* Nov. 13–14, 1994, pp. 608–612.
[95] ——, "Predictive residual vector quantization," *IEEE Trans. Image Processing,* vol. 4, no. 11, pp. 1482–1495, Nov. 1995.
[96] F. Kossentini, M. J. T. Smith, and C. F. Barnes, "Residual VQ with state prediction: A new method for image coding," in *Proc. Visual Commun. Image Processing,* Nov. 11–13, 1991, pp. 383–394.
[97] S. A. Rizvi and N. M. Nasrabadi, "Finite state residual vector quantization using tree-structured competitive neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Detroit, MI, vol. 4, May 9–12, 1995, pp. 2579–2582.
[98] S. A. Rizvi, L.-C. Wang, and N. M. Nasrabdi, "Finite-state residual vector quantization," in *Proc. SPIE Visual Commun. Image Processing,* Taipei, Taiwan, May 23–26, 1995, pp. 608–612.
[99] P.-C. Chang and R. M. Gray, "Gradient algorithms for designing predictive vector quanitzers," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 34, no. 4, pp. 679–690, Aug. 1986.
[100] K. Zeger, "Corrections to 'Gradient algorithms for designing predictive vector quantizers.'" *IEEE Trans. Signal Processing,* vol. 39, pp. 764–765, Mar. 1991.
[101] N. Mohsenian, S. A. Rizvi, and N. M. Nasrabadi, "Predictive vector quantization using a neural net approach," *Opt. Eng.,* vol. 3, no. 7, pp. 1503–1513, July 1993.
[102] S. A. Rizvi, L.-C. Wang, and N. M. Nasrabadi, "Neural network vector predictors with applications to image coding," in *Proc. IEEE Int. Conf. Image Processing,* Washington, DC, Oct. 23–26, 1995, pp. 296–299.
[103] S. A. Rizvi and N. M. Nasrabadi, "Variable-rate predictive residual vector quantizer," *IEEE Signal Processing Lett.,* vol. 2, no. 4, pp. 70–72, Apr. 1995.
[104] S. A. Rizvi, N. M. Nasrabadi, and L.-C. Wang, "Variable-rate predictive residual vector quantizer," in *Proc. SPIE Visual Commun. Image Processing,* Taipei, Taiwan, May 23–26, 1995.
[105] S. A. Rizvi, L.-C. Wang, and N. M. Nasrabadi, "Entropy-constrained predictive residual vector quantization of digital images," in *Proc. IEEE Int. Conf. Image Processing,* Washington, DC, Oct. 23–26, 1995.
[106] T. Kim, "Side match and overlap match vector quantizers for images," *IEEE Trans. Image Processing,* vol. 1, no. 2, pp. 170–185, Apr. 1992.
[107] B. Hammer, A. V. Brandt, and M. Schiein, "Hierarchical encoding of image sequences using multistage vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Apr. 1987, pp. 1055–1058.
[108] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.,* vol. 27, pp. 379–423, 623–656, 1948.
[109] ——, "Communication in the presence of noise," *Proc. IRE,* vol. 37, pp. 10–21, Jan. 1949.
[110] ——, "Coding theorems for a discrete source with a fidelity criterion," in *IRE Nat. Conv. Rec., Part 4,* Mar. 1959, pp. 142–163; also in *Information and Decision Processes,* R. E. Machol, Ed. New York: McGraw Hill, 1960, pp. 93–126.
[111] L. Wang and M. Goldberg, "Progressing image transmission using vector quantization on image in pyramid form," *IEEE Trans. Commun.,* vol. 37, pp. 1339–1349, Dec. 1989.
[112] ——, "Block transform image coding by multistage vector quantization with optimal bit allocation," *IEEE Trans. Commun.,* vol. 39, no. 9, 1379–1388, Sept. 1991.
[113] C.-K. Chan and Y.-H. Chan, "Progressive image transmission using variable block size vector quantization," in *Proc. Fourth Int. Conf. Signal Processing Applications Technol.,* Santa Clara, CA, vol. 1, Sept. 28–Oct. 1, 1993, pp. 761–765.
[114] K. Ramchandran, A. Ortega, K. M. Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding," *IEEE J. Selected Areas Commun.,* vol. 11, no. 1, pp. 6–23, Jan. 1993.
[115] A. Singh and V. M. Bove, Jr., "Multidimensional quantizers for scalable video compression," *IEEE J. Selected Areas Commun.,* vol. 11, no. 1, pp. 36–45, Jan. 1993.
[116] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding," *IEEE Trans. Inform. Theory,* vol. 35, no. 2, pp. 299–315, Mar. 1989.
[117] W. H. R. Equitz, "Successive refinement on information," Ph.D. thesis, Dept. of Elect. Eng., Stanford Univ., Stanford, CA, USA, 1989.
[118] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory,* vol. 37, no. 2, pp. 269–275, Mar. 1991.
[119] B. Rimoldi, "Successive refinement on information: Characterization of the achievable rates," *IEEE Trans. Inform. Theory,* vol. 40, no. 1, pp. 253–259, Jan. 1994.
[120] Y. Ho and A. Gersho, "Variable-rate multi-stage vector quantization for image coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* New York, vol. 3, Apr. 1988, pp. 1156–1159.
[121] R. J. Safranek, K. MacKay, N. S. Jayant, and T. Kim, "Image coding based on selected quantization of the reconstruction noise in the dominant sub-band," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* New York, vol. 2, Apr. 1988, pp. 756–768.
[122] W.-Y. Chan and A. Gersho, "High fidelity audio transform coding with vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Albuquerque, NM, 1990, pp. 1109–1112.
[123] C. F. Barnes, and E. J. Holder, "Classified variable rate residual vector quantization applied to image subband coding," J. A. Storer and M. Cohn Eds., in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 30–Apr. 2, 1993, pp. 272–281.
[124] F. Bosveld, R. L. Lagendijk, and J. Biemond, "Compatible HDTV distribution using fixed distortion subband coding," in *Proc. HDTB Workshop,* Kawasaki, Japan, 1992.

[125] W.-Y. Chan and A. Gersho, "Constrained storage vector quantization in high fidelity audio transform coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Toronto, Canada, vol. 5, May 14–17, 1991, pp. 3597–3600.
[126] C. F. Barnes and E. S. Sjoberg, "A comparative study of JPEG and residual VQ compression of radar imagery," in *Proc. Nat. Telesyst. Conf.: Commercial Applications Dual-Use Technol.,* Atlanta, GA, June 1993, pp. 203–207.
[127] H. Gharavi and A. Tabatabai, "Sub-band coding of monochrome and color images," *IEEE Trans. Circuits Syst.,* vol. 35, no. 2, pp. 207–214, Feb. 1988.
[128] N. Coppisetti, S. C. Kwatra, and A. K. Al-Asmari, "Low-complexity subband encoding for HDTV images," *IEEE J. Selected Areas Commun.,* vol. 11, no. 1, pp. 77–87, Jan. 1993.
[129] L. D. Davisson, "Rate-distortion theory and application," *Proc. IEEE,* vol. 60, no. 7, pp. 800–808, July 1972.
[130] I. Furukawa, M. Nomura, and S. Ono, "Hierarchical coding of super high definition images with subband + multistage VQ," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Toronto, Canada, May 14–17, 1991, pp. 2637–2640.
[131] ——, "Hierarchical sub-band coding of super high definition images with adaptive block-size multistage VQ," *Signal Processing: Image Commun.,* vol. 5, nos. 5–6, pp. 527–538, 1993.
[132] Q. Liu, A. K. Chan, C. K. Chui, E. Pettit, and D. Rhines, "A hybrid technique using spline-wavelet packets and vector quantization for high rate image compression," *SPIE Math. Imaging,* vol. 2034, pp. 194–204, 1993.
[133] E. A. B. da Silva, D. G. Sampson, and M. Ghanbari, "Image coding using successive approximation wavelet vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Detroit, MI, 1995, pp. 2201–2204.
[134] P. C. Cosman, S. M. Perlmutter, and K. O. Perlmutter, "Tree-structured vector quantization with significance map for wavelet image coding," J. A. Storer and M. Cohn, Eds., in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 1995.
[135] J. M. Shapiro, "An embedded wavelet hierarchical image coder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* San Francisco, CA, vol. IV, Mar. 23–26, 1992, pp. 657–660.
[136] ——, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing,* vol. 41, no. 12, pp. 3445–3462, Dec. 1994.
[137] A. Said and W. A. Pearlman, "A new fast and efficient image code based on set partitioning in hierarchical trees," submitted to *IEEE Trans. Circuits Syst. Video Technol.,* 1995.
[138] D. L. Donoho, "Unconditional bases are optimal bases for data compression and for statistical estimation," *Applied Comput. Harmonic Anal.,* vol. 1, no. 1, pp. 100–115, Dec. 1993.
[139] Z. Xiong, K. Ramchandran, M. T. Orchard, and K. Asai, "Wavelet packets-based image coding using joint space-frequency quantization," in *Proc. IEEE Int. Conf. Image Processing,* Austin, TX, vol. 3, Nov. 1994, pp. 324–328.
[140] Z. Xiong, N. P. Galatsanos, and M. T. Orchard, "Marginal analysis prioritization for image compression based on a hierarchical wavelet decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Minneapolis, MN, vol. V, Apr. 27–30, 1993, pp. 546–549.
[141] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint optimization of scalar and tree-structured quantization of wavelet image decompositions," in *Proc. Twenty-Seventh Asilomar Conf. Signals, Syst. Comput.,* vol. 1, Nov. 1993, pp. 891–895.
[142] K. M. Uz, J. M. Shapiro, and M. Czigler, "Optimal bit allocation in the presence of quantizer feedback," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Minneapolis, MN, vol. V, Apr. 27–30, 1993, pp. 385–388.
[143] P. C. Cosman, R. M. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Trans. Image Processing,* this issue, pp. 202–225.
[144] C. W. Barrett and R. L. Frost, "Lattice-based designs of direct sum codebooks for vector quantization," in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 28–30, 1995, p. 436.
[145] Y. Yamada and S. Tazaki, "Successive approximation vector quantization," *Electron. Commun. in Japan, Part 1,* vol. 69, no. 9, pp. 11–19, Sept. 1986.
[146] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory,* vol. 25, no. 4, pp. 373–380, July 1979.
[147] N. Phamdo, N. Farvardin, and T. Moriya, "A unified approach to tree-structured and multistage vector quantization for noisy channels," *IEEE Trans. Inform. Theory,* vol. 39, no. 3, pp. 835–850, May 1993.
[148] E. S. Jang and N. M. Nasrabadi, "Subband coding with multi-stage VQ for wireless image communications," *IEEE Trans. Video Technol.,* to appear.
[149] M. T. Orchard and K. Ramchandran, "An investigation of wavelet-based image coding using an entropy-constrained quantization framework," in *Proc. Data Compression Conf.,* Snowbird, UT, USA, Mar. 29–31, 1994, pp. 341–350.
[150] Y. H. Kim and J. M. Modestino, "Adaptive entropy coded subband coding of images," *IEEE Trans. Image Processing,* vol. 1, pp. 31–48, Jan. 1992.
[151] F. Kossentini and M. J. T. Smith, "Adaptive entropy-constrained residual vector quantization," *IEEE Signal Processing Lett.,* vol. 1, no. 8, pp. 121–123, Aug. 1994.
[152] S. Luttrell, "Hierarchical vector quantization," *Proc. Inst. Elec. Eng.,* vol. 136, pp. 405–413, 1989.
[153] H. Jafarkhani and N. Farvardin, "A scalable wavelet image coding scheme using multi-stage pruned tree-structured vector quantization," in *Proc. IEEE Int. Conf. Image Processing,* Washington, DC, Oct. 1995.



Christopher F. Barnes (S'85, M'89) received the B.S., M.S., and Ph.D. degrees from Brigham Young University, Provo, UT, USA, in 1986, 1987, and 1989, respectively.

He is employed by the Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, USA, working in the areas of sensor signal processing and data exploitation. He is an adjunct professor with the Communications Group of the Electrical and Computer Engineering Department of the Georgia Institute of Technology. He has also been a technical Japanese-to-English translator for Scripta Technica, Inc. since 1992. His research interests include source and channel coding, coded modulation, estimation, and pattern recognition.

Dr. Barnes is a member of Eta Kappa Nu, Sigma Xi, and Phi Kappa Phi.

Syed A. Rizvi (S'92) received the B.Sc. degree (with honors) from the University of Engineering and Technology, Lahore, Pakistan, in 1990 and the M.S. degree from the State University of New York (SUNY) at Buffalo, Bufaslo, NY, USA, in 1993, both in electrical engineering. He is currently a candidate for the Ph.D. degree in the Departrment of Electrical and Computer Engineering at SUNY at Buffalo.

From 1992 to 1995, he was a teaching assistant at SUNY at Buffalo. Since May 1995, he has been working with the U.S. Army Research Laboratory, Adelphi, MD, USA, as a research associate, where his responsibilities include the development of coding algorithms for FLIR images and video using wavelet decomposition and vector quantization. His research interests include image and video coding, applications of artificial neural networks in image processing, and computer vision.

Nasser M. Nasrabadi (SM'92), For photograph and biography, please see this issue, p. 200.