Necessary Conditions for the Optimality of Variable Rate Residual Vector Quantizers*

Faouzi Kossentini and Mark J. T. Smith
Digital Signal Processing Laboratory
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Christopher F. Barnes
Georgia Tech Research Institute
Georgia Institute of Technology
Atlanta, GA 30332

June 15, 1993

Abstract— **Residual vector quantization (RVQ), or multistage VQ, as it is also called, has recently been shown to be a competitive technique for data compression [1]. The competitive performance of RVQ reported in [1] results from the joint optimization of variable rate encoding and RVQ *direct-sum* codebooks. In this paper, necessary conditions for the optimality of variable rate RVQs are derived, and an iterative descent algorithm based on a Lagrangian formulation is introduced for designing RVQs having minimum average distortion subject to an entropy constraint. Simulation results for these *entropy-constrained* RVQs (EC-RVQs) are presented for memoryless Gaussian, Laplacian, and uniform sources. A Gauss-Markov source is also considered. The performance is superior to that of entropy-constrained scalar quantizers (EC-SQs) and practical entropy-constrained vector quantizers (EC-VQs), and is competitive with that of some of the best source coding techniques that have appeared in the literature.**

Index Terms—**Residual vector quantization, multistage vector quantization, entropy, source coding.**

*This work was supported in part by the National Science Foundation under contract MIP-9116113 and by NASA.

1 Introduction

Residual Vector Quantization (RVQ), or multistage VQ, as it is also called, was originally introduced in 1982 [2]. Its structure, which is shown in Figure 1, consists of a cascade of VQ stages (hence the name multistage VQ). For the pth stage VQ the input vector $\boldsymbol{x}_p$ is quantized resulting in the approximation $\hat{\boldsymbol{x}}_p$. The difference is then computed to form the residual $\boldsymbol{x}_{p+1} = \boldsymbol{x}_p - \hat{\boldsymbol{x}}_p$, which serves as an input to the next stage. This aspect of the structure motivates the name *residual* VQ or RVQ.

Perhaps the most striking benefit of RVQ is its memory efficient structure. An RVQ with P stages and N_p vectors per stage ($1 \leq p \leq P$) can uniquely represent $\prod_{p=1}^{P} N_p$ vectors with only $\sum_{p=1}^{P} N_p$ vectors needed for storage. Furthermore, similar savings in computation may be achieved by exploiting the RVQ tree structure.

Despite these attractive features, RVQ has received little attention until recently. Early assessments of its utility, as reported by Baker [3] and in a survey paper by Makhoul, et al. [4], were somewhat discouraging. In the former case, some preliminary investigations with RVQ structures having more than two stages (applied to image coding) led to the conclusion that it is not advantageous to iteratively vector quantize image waveform residuals [3, p. 102]. In the latter case, Makhoul, et al. observed a rapid degradation in performance for RVQ applied to speech coding as the number of stages was increased and suggested that RVQ be limited to not more than two or three stages.

In 1989, Barnes [5] introduced an analysis of RVQ in which the RVQ is optimized subject to the imposed structural constraint. The new design method led to an improvement in performance over previous design methods. Since then the technical literature has shown much activity in the area of RVQ and the application of RVQ to data compression has become more widespread [6, 7, 8, 9, 10, 11, 12, 13].

In this paper, we extend the theory and design methods of *fixed rate* RVQs to the case of *variable rate* RVQ. The first part of the paper (Section 2) follows the work presented in [14, 15] where necessary conditions for the optimality of fixed rate RVQ are derived. Here, however, we present a mathematical treatment of convergence for the RVQ design algorithm. The next part of the paper gives a derivation of optimality conditions for variable rate RVQ. It is well known that variable rate systems can yield a lower average rate than fixed rate systems. This property has been demonstrated in [16, 17] for entropy-constrained VQ (EC-VQ). EC-VQ has shown some of the best performance results among entropy coded quantization schemes. In our discussions, a theory for entropy constrained RVQ (EC-RVQ) is developed. In addition, a *locally* optimal design algorithm is introduced and convergence issues are addressed. The paper concludes with an evaluation and comparison of the performance of EC-RVQ on some well-known synthetic sources. Simulation results show that EC-RVQ achieves some of the best performance results reported to date.

2 Fixed Rate RVQ

The first approach introduced for the design of RVQs consists of using the LBG algorithm sequentially on each stage [2]. Although each of the stage codebooks is designed to minimize the average distortion introduced by that stage (given fixed prior stages), there is no guarantee that the overall average distortion introduced by the RVQ is minimized. A better design technique is one that designs the stage codebooks *jointly* to minimize the overall average distortion. The key to optimizing the RVQ stages jointly is to view the RVQ in terms of a structurally constrained *direct-sum* codebook (that is, a codebook that contains all possible ordered direct-sums of stage code vectors) and find necessary conditions for the optimality of that direct-sum codebook (i.e., joint optimality of all stage codebooks).

A direct-sum codebook may be depicted in several ways. Here we choose to view it diagramatically as a tree. To illustrate this, consider a three-stage RVQ with two vectors in each stage codebook: stage 1 contains vectors $\boldsymbol{y}_1(1), \boldsymbol{y}_1(2)$; stage 2 contains vectors $\boldsymbol{y}_2(1), \boldsymbol{y}_2(2)$; and stage 3 contains $\boldsymbol{y}_3(1), \boldsymbol{y}_3(2)$. Figure 2 shows a tree corresponding to this RVQ where the stages are delineated by the dashed lines and the stage code vectors appear inside the nodes. Eight nodes appear at the base of the tree, each one corresponding to a direct-sum code vector. The value of any one of the eight code vectors is obtained by tracing the unique path from bottom to top and summing the stage code vectors (shown inside the nodes) along the way. This simple tree interpretation is helpful for suggesting efficient RVQ encoder structures, and for understanding both the optimality conditions and the corresponding RVQ design algorithms.

Equally important to the discussion is the mathematical notation used to describe inputs, outputs, and the various components of the RVQ. Let $\boldsymbol{x}_1$ be a realization of the random k-dimensional vector $\boldsymbol{X}_1$ described by the probability density function (pdf) $f_{\boldsymbol{X}_1}(\boldsymbol{x}_1)$ on $\Re^k$. A P-stage RVQ (see Figure 1) consists of a finite sequence of P vector quantizers. For the pth stage VQ where $1 \leq p \leq P$, let us define the following symbols:

N_p	the pth stage codebook size
j_p	the pth stage index: $\{1 \leq j_p \leq N_p\}$
J_p	the pth set of all possible values for j_p: i.e. $\{1, 2, \ldots, N_p\}$
$\boldsymbol{y}_p(j_p)$	the j_pth code vector of the pth stage
$S_p(j_p)$	the j_pth partition cell of the pth stage
$V_p(j_p)$	the j_pth stage-removed residual equivalent class of the pth stage
$\mathcal{C}_p$	the pth stage codebook $\{\boldsymbol{y}_p(j_p) : j_p \in J_p\}$
$\mathcal{P}_p$	the pth stage partition $\{S_p(j_p) : j_p \in J_p\}$
Q_p	the pth stage quantizer mapping

The pth stage VQ quantizes the residual vector $\boldsymbol{x}_p$ and outputs $Q_p(\boldsymbol{x}_p)$. The pth stage quantizer mapping $Q_p : \Re^k \mapsto C_p$ can be realized by a composition of a fixed length encoder mapping $E_p : \Re^k \mapsto J_p$ where

$$E_p(\boldsymbol{x}_p) = j_p \text{ if and only if } \boldsymbol{x}_p \in S_p(j_p),$$

and a fixed length decoder mapping $D_p : J_p \mapsto C_p$ where

$$D_p(j_p) = \boldsymbol{y}_p(j_p).$$

As stated in the previous section, a P-stage RVQ can be represented by a tree as illustrated in Figure 2. The associated "single-stage" direct-sum VQ codebook and the tree-structured RVQ codebook are identical in the sense that they produce the same representation of the source output, and thus, have the same expected distortion. For the direct-sum VQ, let us define the following symbols:

N	direct-sum codebook size ($N = \prod_{p=1}^{P} N_p$)
$\boldsymbol{J}$	direct-sum P-tuple index set, $\boldsymbol{J} = J_1 \times J_2 \times \cdots \times J_P$
$\boldsymbol{j}$	a P-tuple index in $\boldsymbol{J}$
$\boldsymbol{y}(\boldsymbol{j})$	$\boldsymbol{j}$th direct-sum code vector
$\boldsymbol{V}(\boldsymbol{j})$	$\boldsymbol{j}$th direct-sum partition cell
$\boldsymbol{C}$	direct-sum codebook $\{\boldsymbol{y}(\boldsymbol{j}) : \boldsymbol{j} \in \boldsymbol{J}\}$
$\boldsymbol{P}$	direct-sum partition $\{\boldsymbol{V}(\boldsymbol{j}) : \boldsymbol{j} \in \boldsymbol{J}\}$
$\boldsymbol{Q}$	direct-sum mapping

The direct-sum codebook contains all possible ordered sums of the stage code vectors, i.e., $\boldsymbol{C} = C_1 + C_2 + \ldots + C_P$. The direct-sum code vectors are given by

$$\boldsymbol{y}(\boldsymbol{j}) = \sum_{p=1}^{P} \boldsymbol{y}_p(j_p),$$

where j_p is the pth member of the ordered P-tuple index $\boldsymbol{j}$. The direct-sum VQ quantizes the source vector $\boldsymbol{x}_1$ and outputs the representation $\hat{\boldsymbol{x}}_1 = \boldsymbol{Q}(\boldsymbol{x}_1)$ given by

$$\boldsymbol{Q}(\boldsymbol{x}_1) = \sum_{p=1}^{P} Q_p(\boldsymbol{x}_p),$$

where

$$\boldsymbol{x}_p = \boldsymbol{x}_1 - \sum_{i=1}^{p-1} Q_i(\boldsymbol{x}_i), \quad p > 1,$$

is the pth stage *causal residual.* The term *causal* refers to the sequential process used to compute the residual, i.e., the stage residuals are computed sequentially starting from the first stage to the pth stage.

2.1 Necessary Conditions for Optimal Fixed Rate RVQ

Let the distortion that results from representing $\boldsymbol{x}$ with $\boldsymbol{y}$ be expressed by $d(\boldsymbol{x}, \boldsymbol{y})$. The distortion measure $d(\boldsymbol{x}, \boldsymbol{y})$ is assumed to be a non-negative real-valued function that satisfies the following requirements:

1. For any fixed $\boldsymbol{x} \in \Re^k$, $d(\boldsymbol{x}, \boldsymbol{y})$ is a continuously differentiable function of $\boldsymbol{y} \in \Re^k$.

2. $d(\boldsymbol{x}, \boldsymbol{y})$ is translation invariant.

3. For any fixed $\boldsymbol{x} \in \Re^k$, $d(\boldsymbol{x}, \boldsymbol{y})$ is a strictly convex function of $\boldsymbol{y}$, that is, $\forall \boldsymbol{y}_1, \boldsymbol{y}_2 \in \Re^k$ and $\lambda \in (0,1), d(\boldsymbol{x}, \lambda \boldsymbol{y}_1 + (1-\lambda)\boldsymbol{y}_2) < \lambda d(\boldsymbol{x}, \boldsymbol{y}_1) + (1-\lambda) d(\boldsymbol{x}, \boldsymbol{y}_2)$.

A P-stage RVQ is said to be optimal if it gives at least a locally minimum value of the average distortion incurred in representing $\boldsymbol{x}_1$ with $\hat{\boldsymbol{x}}_1$,

$$D(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1) = E\left\{ d\left[\boldsymbol{x}_1, \sum_{p=1}^{P} Q_p(\boldsymbol{x}_p)\right]\right\}. \tag{1}$$

For stage codebook and partition optimality, (1) should be minimized with respect to stage codebook and partition parameters. However, this minimization is complicated by the fact that knowledge of the joint pdf $f_{\boldsymbol{X}_1 \ldots \boldsymbol{X}_P}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P)$ is required, which, in turn, depends in a complicated fashion upon the sequence of stage codebooks and partitions. This optimization problem can be made tractable by viewing the RVQ product code as a single-stage VQ with a structurally constrained direct-sum codebook (i.e., the direct-sum code vectors are structurally dependent). By minimizing the average distortion of the direct-sum quantizer,

$$D(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1) = E\left\{d\left(\boldsymbol{x}_1, Q(\boldsymbol{x}_1)\right)\right\},$$

the problem of dealing explicitly with the complicated structural interdependencies that exist among the stages of the RVQ is avoided.

First, to derive optimality conditions for a fixed rate RVQ direct-sum partition, assume that the stage codebooks $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P\}$ are fixed, which implies that the direct-sum codebook $\boldsymbol{C}$ is also fixed. Then

$$E\left\{d[\boldsymbol{x}_1, Q(\boldsymbol{x}_1)]\right\} \geq E\left\{\min_{\boldsymbol{y}(\boldsymbol{j}) \in \boldsymbol{C}} d[\boldsymbol{x}_1, \boldsymbol{y}(\boldsymbol{j})]\right\}.$$

That is, no direct-sum partition can yield lower average distortion than the partition obtained by the nearest-neighbor mapping. Accordingly, we have the nearest-neighbor encoding rule,

$$\boldsymbol{x}_1 \in \boldsymbol{V}^*(\boldsymbol{j}) \quad \textit{iff} \quad d[\boldsymbol{x}_1, \boldsymbol{y}(\boldsymbol{j})] \leq d[\boldsymbol{x}_1, \boldsymbol{y}(\boldsymbol{k})] \quad \text{for all } \boldsymbol{k} \in \boldsymbol{J}. \tag{2}$$

The optimal direct-sum partition cells are denoted with asterisks, $\boldsymbol{V}^*(\boldsymbol{j})$.

The next step is to determine necessary conditions for optimal stage code vectors. For the derivation that follows it is useful to introduce the *stage-removed* index mapping $\beta_p : \boldsymbol{J} \mapsto \tilde{\boldsymbol{J}}_p$, $\tilde{\boldsymbol{J}}_p = J_1 \times J_2 \times \cdots \times J_{p-1} \times J_{p+1} \times \cdots \times J_P$, defined by

$$\beta_p(\boldsymbol{j}) = (j_1, j_2, \ldots, j_{p-1}, j_{p+1}, \ldots, j_P)$$

for $\boldsymbol{j} \in \boldsymbol{J}$. Note that $\beta_p(\boldsymbol{j})$ includes all members of $\boldsymbol{j}$ except the pth member, hence the name *stage-removed* index. This index represents a shortened path through the RVQ tree where the pth level branch has been removed, and the remainder of the path starting with the $(p+1)$th level branch has been added or grafted back into the tree structure. Hence, each direct-sum code vector $\boldsymbol{y}(\boldsymbol{j})$, where $\boldsymbol{j} = (j_1, j_2, \ldots, j_{p-1}, j_p, j_{p+1}, \ldots, j_P) \in \boldsymbol{J}$, can be written as

$$\boldsymbol{y}(\boldsymbol{j}) = \boldsymbol{g}(\beta_p(\boldsymbol{j})) + \boldsymbol{y}_p(j_p),$$

where

$$\boldsymbol{g}(\beta_p(\boldsymbol{j})) = \sum_{\substack{i=1 \\ i \neq p}}^{P} \boldsymbol{y}_i(j_i)$$

is the pth *stage-removed* direct-sum path of the RVQ tree.

Given a particular $\boldsymbol{x}_1 \in \Re^k$ and a fixed RVQ encoding rule, there exists a pth *stage-removed* residual vector defined by

$$\gamma_p = \boldsymbol{x}_1 - \boldsymbol{g}(\beta_p(\boldsymbol{j})).$$

This residual vector is the difference between the input and the stage-removed direct-sum vector. Because the stage-removed residual γ_p is a translation (conditioned on the pth stage) of the realization $\boldsymbol{x}_1$ of the random vector $\boldsymbol{X}_1$, it is also a realization of a random vector $\boldsymbol{\Gamma}_p$ with associated stage-removed residual probability density function $f_{\boldsymbol{\Gamma}_p}(\gamma_p)$.

In addition, let $H_p(j_p)$ be the set of P-tuple indices corresponding to all direct-sum code vectors $\boldsymbol{y}(\boldsymbol{j})$ that contain $\boldsymbol{y}_p(j_p)$ in their construction. In other words, $H_p(j_p) \subset \boldsymbol{J}$ is the set of all indices such that $j_p \in J_p$ is the pth element of $\boldsymbol{j}$. The set $H_p(j_p)$ can be used to describe the j_pth *stage-removed* residual equivalence class $V_p(j_p)$ by

$$V_p(j_p) = \bigcup_{\boldsymbol{j} \in H_p(j_p)} (\boldsymbol{V}(\boldsymbol{j}) - \boldsymbol{g}(\beta_p(\boldsymbol{j}))), \tag{3}$$

where $\boldsymbol{V}(\boldsymbol{j}) - \boldsymbol{g}(\beta_p(\boldsymbol{j}))$ indicates that all $\boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})$ have been translated by $\boldsymbol{g}(\beta_p(\boldsymbol{j}))$. If $\boldsymbol{V}(\boldsymbol{j})$ is assumed to be an optimal partition, i.e. $\boldsymbol{V}(\boldsymbol{j}) = \boldsymbol{V}^*(\boldsymbol{j})$, then $V_p(j_p) = V_p^*(j_p)$ is an optimal stage-removed residual equivalence class.

To determine necessary conditions for optimal stage code vectors, assume that the stage partitions $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P\}$ are fixed, which implies that the direct-sum partition $\boldsymbol{P}$ is fixed. Now let K_p be the set of all possible pth stage codebooks $\mathcal{C}_p$ with N_p vectors, and let $\mathbf{K}$ be the set of all possible direct-sum codebooks formed from the K_p's with $1 \leq p \leq P$. Also let $F : \mathbf{K} \mapsto [0, \infty)$ be the function given by

$$F(C) \;=\; \sum_{\boldsymbol{j} \in \boldsymbol{J}} E_{\boldsymbol{X}_1} \{ d(\boldsymbol{x}_1, \boldsymbol{y}(\boldsymbol{j})) \,|\boldsymbol{x}_1 \in V(\boldsymbol{j}) \} \;\; \mathrm{pr}\{\boldsymbol{x}_1 \in V(\boldsymbol{j})\}, \tag{4}$$

for $\boldsymbol{y}(\boldsymbol{j}) \in C$ and $C \in \mathbf{K}$. To find a minimum for the average distortion (4), it suffices to find a sequence of codebooks $(\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_P^*) \;\in\; K_1 \times K_2 \times \ldots \times K_P$ and corresponding direct-sum codebook $C^* \in \mathbf{K}$ that minimizes F. Coordinate descent algorithms can be used to find such a minimum. These algorithms are based on the following procedure: we hold fixed all stage codebooks, except for the pth stage codebook, and then we minimize F with respect to $\mathcal{C}_p$. This is an iterative procedure and is performed for each stage (i.e. all values of p) until $F(C)$ converges to a minimum. There are two common forms of implementation [18]. In the first, often called the nonlinear Jacobi algorithm, the minimizations with respect to the different codebooks $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P\}$ are carried out simultaneously. Mathematically, the nonlinear Jacobi algorithm is described by

$$\mathcal{C}_p(t+1) = \arg\min_{\mathcal{C}_p} F\left(\mathcal{C}_1(t), \ldots, \mathcal{C}_{p-1}(t), \mathcal{C}_p, \mathcal{C}_{p+1}(t), \ldots, \mathcal{C}_P(t)\right), \tag{5}$$

for $1 \leq p \leq P$. In the second approach, often called the nonlinear Gauss-Seidel algorithm, the minimizations are carried out successively for each codebook and may be described mathematically by

$$\mathcal{C}_p(t+1) = \arg\min_{\mathcal{C}_p} F\left(\mathcal{C}_1(t+1), \ldots, \mathcal{C}_{p-1}(t+1), \mathcal{C}_p, \mathcal{C}_{p+1}(t), \ldots, \mathcal{C}_P(t)\right), \tag{6}$$

for $1 \leq p \leq P$. Let us assume all stage codebooks (except for the pth stage codebook) are fixed. Also, let us modify (4) by writing

$$F(C) = \sum_{j_p \in J_p} \sum_{\beta_p(\boldsymbol{j}) \in \boldsymbol{J}_p} E_{\boldsymbol{X}_1} \left\{ d\left[\boldsymbol{x}_1, \boldsymbol{g}(\beta_p(\boldsymbol{j})) + \boldsymbol{y}_p(j_p) \mid \boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})\right]\right\} \mathrm{pr}\left\{\boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})\right\}.$$

Using the assumption that the distortion measure is translation invariant, and also using (3) together with the law of total probability, we can rewrite the above equation as

$$\begin{aligned} F(C) &= \sum_{j_p \in J_p} E_{\Gamma_p|j_p} \left\{ d\left[\gamma_p, \boldsymbol{y}_p(j_p)\right] |\gamma_p \in V_p(j_p)\right\} \ \mathrm{pr}\left\{\gamma_p \in V_p(j_p)\right\} \\ &\geq \sum_{j_p \in J_p} \inf_{\boldsymbol{u} \in \Re^k} E_{\Gamma_p|j_p} \left\{ d\left(\gamma_p, \boldsymbol{u}\right) |\gamma_p \in V_p(j_p)\right\} \ \mathrm{pr}\left\{\gamma_p \in V_p(j_p)\right\}. \end{aligned} \qquad (7)$$

In [19], it is shown that provided $\mathrm{pr}\left\{\gamma_p \in V_p(j_p)\right\} \neq 0$, there exist $\boldsymbol{y}_p^*(j_p) \in \Re^k$ (which we call *stage-removed residual centroids*) for the stage-removed residual equivalence classes $V_p(j_p)$ such that

$$\int d\left[\gamma_p, \boldsymbol{y}_p^*(j_p)\right] f_{\Gamma_p|j_p}(\gamma_p) d\gamma_p = \inf_{\boldsymbol{u} \in \Re^k} \int d(\gamma_p, \boldsymbol{u}) f_{\Gamma_p|j_p}(\gamma_p) d\gamma_p < \infty, \qquad (8)$$

and that the set of all solutions $\boldsymbol{y}_p^*(j_p)$ to (8) is convex, closed, and bounded. Since the distortion measure $d(\boldsymbol{x}, \boldsymbol{y})$ is assumed to be strictly convex in $\boldsymbol{y}$, the solution is unique. In (8) the pdf $f_{\Gamma_p|j_p}(\gamma_p)$ is related to the source pdf $f_{\boldsymbol{X}_1}(\boldsymbol{x}_1)$ according to

$$f_{\Gamma_p|j_p}(\gamma_p) = \frac{\sum_{\boldsymbol{j} \in H_p(j_p)} \boldsymbol{I}[V(\boldsymbol{j})] f_{\boldsymbol{X}_1}\left[\boldsymbol{g}(\beta_p(\boldsymbol{j})) + \gamma_p\right]}{\mathrm{pr}\left\{\gamma_p \in V_p(j_p)\right\}}, \qquad (9)$$

where $\boldsymbol{I}[\boldsymbol{V}(\boldsymbol{j})]$ is an indicator function for the direct-sum partition cell $\boldsymbol{V}(\boldsymbol{j})$, that is, $\boldsymbol{I}[\boldsymbol{V}(\boldsymbol{j})] = 1$ if $\boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})$ and $\boldsymbol{I}[\boldsymbol{V}(\boldsymbol{j})] = 0$ otherwise. The $\boldsymbol{y}_p(j_p)$'s which satisfy (8) are generalized centroids of stage-removed residual vectors (i.e., residual vectors formed from the encodings of all *prior* and *subsequent* RVQ stages). Hence, the second condition will be referred to as the *stage-removed residual centroid condition.*

Convergence of the nonlinear Gauss-Seidel algorithm applied to RVQ can now be established using a descent approach.

Proposition 1: Suppose F is continuously differentiable and convex on $K_1 \times K_2 \times \ldots \times K_P$. Furthermore, suppose that for each $p \in \{1, 2, \ldots, P\}$, F is a strictly convex function of $\mathcal{C}_p$ when the other codebooks are held fixed. Let $\{(\mathcal{C}_1(t), \ldots, \mathcal{C}_P(t))\}$ with $t = 0, 1, 2, \ldots$ be a sequence of stage codebooks generated by the nonlinear Gauss-Seidel algorithm. Then, every limit point of $\{(\mathcal{C}_1(t), \ldots, \mathcal{C}_P(t))\}$ minimizes F over $K_1 \times K_2 \times \ldots \times K_P$.

Details of the convergence proof are given in [20]. The proof is based on a descent approach. In particular, successive minimizations cannot increase the value of $F[\mathcal{C}_1(t), \ldots, \mathcal{C}_P(t)]$. This shows that $F[\mathcal{C}_1(t+1), \ldots, \mathcal{C}_P(t+1)] \leq F[\mathcal{C}_1(t), \ldots, \mathcal{C}_P(t)]$ and implies the convergence of $F[\mathcal{C}_1(t), \ldots, \mathcal{C}_P(t)]$ provided that F is bounded below. It should be noted that if F is not differentiable, the Gauss-Seidel algorithm may fail to converge to a minimum.

The proof outlined above does not apply to the Jacobi algorithm. Even though minimizations with respect to each stage cannot increase the value of F, the fact that these minimizations are carried out simultaneously allows the possibility that $F[\mathcal{C}_1(t+1), \ldots, \mathcal{C}_P(t+1)] > F[\mathcal{C}_1(t), \ldots, \mathcal{C}_P(t)]$. However, convergence of the nonlinear Jacobi algorithm can be established under suitable assumptions on the new codebook selection rule or mapping R: $K_1 \times K_2 \times \ldots \times K_P \mapsto K_1 \times K_2 \times \ldots \times K_P$, given by

$$R(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P) = (\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P) - c\nabla F(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P), \tag{10}$$

where c is a positive real number and ∇F denotes the gradient of F [21].

Proposition 2: Let F be a continuously differentiable function, let c be a real number, and suppose that the mapping $R(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P)$ given by (10) is a contraction mapping with respect to the block-max norm $\mathbf{B}(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P) = \max_p ||\mathcal{C}_p||_p / w_p$, where each $||\cdot||_p$ is the Euclidean norm on K_p and each w_p is a positive real number. Then, there exists a unique vector $(\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_P^*)$ that minimizes F over $K_1 \times K_2 \times \ldots \times K_P$. Moreover, the sequence $\{(\mathcal{C}_1(t), \ldots, \mathcal{C}_P(t))\}$ generated by either of the two algorithms (described by (5) and (6)) converges to $(\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_P^*)$ geometrically. For proof, see [20].

A common distortion measure is the *squared error distortion measure* defined by

$$d(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||^2 = \sum_{i=1}^{k}(x_i - y_i)^2,$$

where $||\cdot||$ denotes the Euclidean norm and x_i and y_i are elements of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. This distortion measure can be written in the form

$$d(\boldsymbol{x}, \boldsymbol{y}) = \rho(||\boldsymbol{x} - \boldsymbol{y}||)$$

where $\rho(\alpha) = \alpha^2$. Obviously, ρ is a continuously differentiable and strictly convex function on $[0, \infty)$ with $\rho(0) = 0$. It follows that the squared error distortion measure satisfies the requirements (1)-(3) in Section 2.1. Therefore, it can be easily shown that F is continuously differentiable and convex on $K_1 \times K_2 \times \ldots \times K_P$, and that F is a strictly convex function of $\mathcal{C}_p$. Thus, Proposition 1 guarantees that when the squared error distortion is used, the nonlinear Gauss-Seidel algorithm converges to a minimum.

A necessary condition for $R(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P)$ to be a contraction mapping is that $x - c[\rho(x)]'$ be a contraction mapping for any positive real number c. It is clear that the function $\rho(x) = x^2$ does not satisfy such a requirement, and Proposition 2 cannot be used to guarantee the convergence of the Jacobi algorithm (when the squared error distortion measure is used). In fact, computer simulations confirm the Jacobi algorithm is not guaranteed to converge, even when the initial vector is close to $(\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_P^*)$.

2.2 The RVQ Design Algorithm

The RVQ design algorithm, introduced in [5], attempts to jointly optimize all stage codebooks to minimize the overall reconstruction error of the RVQ subject to a constraint on the number of direct-sum code vectors. It is an iterative procedure that is similar to the LBG algorithm. However, unlike the LBG algorithm, the optimization of the decoder (assuming the encoder is fixed) is an embedded iterative procedure that guarantees that new stage codebooks minimize the overall average distortion introduced by the RVQ. Therefore, there are two interlaced iterative procedures: one for optimization of the encoder/decoder pair, and another to simultaneously satisfy the stage-removed residual centroid condition in all stages.

Assuming that all stage codebooks are held fixed, the first optimality condition (given by (2)) implies that only exhaustive search encoders are guaranteed, *in general*, to generate an optimal direct-sum Voronoi partition. However, exhaustive search encoding is usually too expensive. An alternative (but generally sub-optimal) encoder is the stage-sequential encoder. Although fast, this encoder is often unable to find the best direct-sum code vector, thereby resulting in what may be a significant increase in average distortion. Another sub-optimal, but *efficient* and *effective* encoder is the M-search encoder. The M-search technique, introduced in [22] for tree searching, was shown to be very efficient when used to search the RVQ tree [23, 15]. The M-search algorithm proceeds one level deeper into the RVQ tree by extending all branches from M surviving nodes, and only the best M of these extended branches survive to the next level. This procedure continues until the last stage of the codebook is reached, and then the code vector of the best path among the final M paths is used. Employing M-search during the optimization of the encoder usually leads to a relatively small complexity, but to close-to-optimal performance [23, 15].

Assuming a fixed direct-sum partition $\boldsymbol{P}$, or equivalently, a fixed set of stage partitions $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P\}$, the Gauss-Seidel algorithm is used to find the constituent codebooks $\{\mathcal{C}_1^*, \mathcal{C}_2^*, \ldots, \mathcal{C}_P^*\}$ with stage code vectors that simultaneously satisfy the stage-removed residual centroid condition (8). It is shown above that, for the squared error distortion measure, the Gauss-Seidel algorithm always converges to a minimum. Therefore, the "decoder-only" iteration used to find a minimizing set of stage codebooks can only reduce or leave unchanged the average distortion.

It is shown in [20] that if the encoder yields a Voronoi partition (in the squared error distortion sense) with respect to the direct-sum codebook and the Gauss-Seidel algorithm is used in the decoder optimization step, the fixed rate RVQ design algorithm converges monotonically to a fixed point which satisfies necessary conditions for minimum squared error distortion. However, it should be emphasized that if a sub-optimal encoder is used, then the encoder optimization step may actually increase the average distortion and monotonic convergence cannot be guaranteed. The possibility of a nonmonotonic average squared error distortion raises the issue of how to effectively terminate the iterative process. Fortunately, experimental results show that the stage-sequential search RVQ design algorithm effectively reduces the average distortion with only occasional deviations from monotonicity. Furthermore, the M-search RVQ design algorithm converged monotonically in all our experiments to a fixed point.

3 Variable Rate RVQ

An optimal variable rate RVQ can be constructed by incorporating the entropy constraint directly into the RVQ design loop. In [1], it is shown that the direct-sum codebook constraint can generally be expected to lead to both an increased average distortion and a decreased output entropy. This motivates an RVQ design algorithm which finds stage code vectors that minimize the average distortion subject to a constraint on the output entropy of the RVQ. Necessary conditions for optimality of variable rate RVQ are derived in the next section, and an *entropy-constrained* RVQ design algorithm which satisfies these conditions is discussed in the following section.

3.1 Necessary Conditions for Optimal Variable Rate RVQ

For the direct-sum VQ, let $\mathcal{J}$ be set of variable length indices $\{c(\boldsymbol{j}) : \boldsymbol{j} \in \boldsymbol{J}\}$. The direct-sum VQ mapping, $\boldsymbol{Q} : \Re^k \mapsto \boldsymbol{C}$, may be realized by a composition of a variable length encoder mapping $\mathcal{E} : \Re^k \mapsto \mathcal{J}$, where

$$\mathcal{E}(\boldsymbol{x}_1) = c(\boldsymbol{j}) \text{ if and only if } \boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j}),$$

and a variable length decoder mapping $\mathcal{D} : \mathcal{J} \mapsto \boldsymbol{C}$ where

$$\mathcal{D}(c(\boldsymbol{j})) = \boldsymbol{y}(\boldsymbol{j}).$$

The variable length encoder can be further decomposed into two mappings, $\mathcal{E} = \mathbf{L} \circ \mathbf{E}$, where $\mathbf{E} : \Re^k \mapsto \boldsymbol{J}$ and $\mathbf{L} : \boldsymbol{J} \mapsto \mathcal{J}$, and o denotes composition. Similarly, one can decompose the variable length decoder into two mappings, $\mathcal{D} = \mathbf{D} \circ (\mathbf{L})^{-1}$, where $(\mathbf{L})^{-1} : \mathcal{J} \mapsto \boldsymbol{J}$, and $\mathbf{D} : \boldsymbol{J} \mapsto \boldsymbol{C}$. Note that the mapping $\mathbf{L}$ is one-to-one and onto, and hence, is an invertible mapping with inverse $(\mathbf{L})^{-1}$.

Let the distortion that results from representing $\boldsymbol{x}_1$ with $\hat{\boldsymbol{x}}_1$, $d(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1)$, be a non-negative real-valued function that satisfies requirements (1)-(3) of Section 2.1. According to distortion-rate theory [24],[25], [26], the kth-order distortion function (where k is the vector size)

$$D_k(R) = \inf_{\text{pr}\{\hat{\boldsymbol{x}}_1 | \boldsymbol{x}_1\}} \{E[d(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1)] \mid I(\boldsymbol{x}_1; \hat{\boldsymbol{x}}_1) \leq R\}$$

is a lower bound to the kth-order *operational* distortion-rate function

$$\hat{D}_k(R) = \inf_{(\mathcal{E},\mathcal{D})} \{E[d(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1)] \mid E[l(\boldsymbol{x}_1)] \leq R\}$$

where $l(\boldsymbol{x}_1) = |\mathcal{E}(\boldsymbol{x}_1)|$ is the length of the codeword representing $\boldsymbol{x}_1$ and $I(\boldsymbol{x}_1; \hat{\boldsymbol{x}}_1)$ is the mutual information between $\boldsymbol{x}_1$ and $\hat{\boldsymbol{x}}_1$. The *convex hull* of $\hat{D}_k(R)$ can be found [16] by minimizing the functional

$$J(\mathcal{E},\mathcal{D}) = E[d(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1)] + \lambda E[l(\boldsymbol{x}_1)]$$

where λ can be interpreted as the slope of a line supporting the convex hull of the operational distortion-rate function $\hat{D}_k(R)$.

A variable rate P-stage RVQ (with an average rate no greater than R) is said to be optimal for $f_{\boldsymbol{X}_1}(\cdot)$ if it gives at least a locally minimum value of the average distortion. The design problem can be stated as follows: Choose the codebook $\boldsymbol{C}$, partition $\boldsymbol{P}$, and variable-length mapping L that minimize the average distortion

$$D(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1) = E\{d(\boldsymbol{x}_1, \boldsymbol{Q}(\boldsymbol{x}_1))\}$$

subject to

$$E\{l(\boldsymbol{x}_1)\} \;\leq\; R,$$

where $l : \Re^k \mapsto \Re$ is the variable length of the codeword representing $\boldsymbol{x}_1$, and is defined by

$$l(\boldsymbol{x}_1) = |\mathcal{E}(\boldsymbol{x}_1)| = |\mathbf{L}(\mathbf{E}(\boldsymbol{x}_1))| = |\mathbf{L}(\boldsymbol{j})|.$$

This constrained minimization problem can be replaced by the following unconstrained minimization problem: Choose the codebook $\boldsymbol{C}$, partition $\boldsymbol{P}$, and variable length mapping $\boldsymbol{L}$ that minimize the Lagrangian

$$J_\lambda(\mathbf{E},\mathbf{L},\mathbf{D}) = E\left\{d(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1) + \lambda\, |\mathbf{L}(\boldsymbol{j})|\right\}. \tag{11}$$

Proceeding, assume the codebooks $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P\}$ are fixed. This implies the direct-sum codebook $\boldsymbol{C}$ is fixed. Also, assume the lengths $|\mathbf{L}(\boldsymbol{j})|$ of the channel codewords associated with the direct-sum code vectors are fixed. Then, a partition $\boldsymbol{P}$ that minimizes (11) is one that minimizes the integrand $d(\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1) + \lambda\ |\mathbf{L}(\boldsymbol{j})|$ almost everywhere. That is,

$$\boldsymbol{x}_1 \in \boldsymbol{V}^*(\boldsymbol{j}) \ \ \textit{iff} \ d[\boldsymbol{x}_1, \boldsymbol{y}(\boldsymbol{j})] + \lambda\ |\mathbf{L}(\boldsymbol{j})| \ \leq \ d[\boldsymbol{x}_1, \boldsymbol{y}(\boldsymbol{k})] + \lambda\ |\mathbf{L}(\boldsymbol{k})| \ \text{ for all } \boldsymbol{k} \in \boldsymbol{J}. \quad (12)$$

Note that (2) is a special case of (12) when $\lambda = 0$.

Next, assume the codebooks $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P\}$ and the partitions $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_P\}$ are fixed. This implies that both the direct-sum codebook $\boldsymbol{C}$ and the direct-sum partition $\boldsymbol{P}$ are fixed. Then, note that (11) can be expressed as

$$J_\lambda(\mathbf{E}, \mathbf{L}, \mathbf{D}) = \sum_{\boldsymbol{j} \in \boldsymbol{J}} E\{d[\boldsymbol{x}_1, \boldsymbol{y}(\boldsymbol{j})] + \lambda |\mathbf{L}(\boldsymbol{j})| \mid \boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})\} \ \mathrm{pr}(\boldsymbol{j}) \qquad (13)$$

where $\mathrm{pr}(\boldsymbol{j}) = \mathrm{pr}\{\boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})\}$. A mapping $\mathbf{L}$ that minimizes (13) is one that minimizes the expected codeword length

$$R = \sum_{\boldsymbol{j} \in \boldsymbol{J}} |\mathbf{L}(\boldsymbol{j})| \ \mathrm{pr}(\boldsymbol{j}).$$

Setting the codeword length $|\mathbf{L}(\boldsymbol{j})|$ to

$$|\mathbf{L}^*(\boldsymbol{j})| = -\log_2 \mathrm{pr}(\boldsymbol{j}) = -\log_2 \mathrm{pr}(j_1, j_2, \ldots, j_P) \qquad (14)$$

results in an average rate which is equal to the output entropy of the direct-sum quantizer.

The probability $\mathrm{pr}(j_1, j_2, \ldots, j_P)$ of a path in the RVQ can also be written as the product of conditional probabilities, i.e.,

$$\mathrm{pr}(j_1, j_2, \ldots, j_P) = \mathrm{pr}(j_P | j_{P-1}, \ldots, j_1)\ \mathrm{pr}(j_{P-1} | j_{P-2}, \ldots, j_1) \ldots \mathrm{pr}(j_2 | j_1)\ \mathrm{pr}(j_1)$$

Therefore, we can write

$$\begin{aligned} |\mathbf{L}^*(\boldsymbol{j})| &= -\log_2 \mathrm{pr}(j_P|j_{P-1},\ldots,j_1) - \log_2 \mathrm{pr}(j_{P-1}|j_{P-2},\ldots,j_1) \\ &\quad -\ldots - \log_2 \mathrm{pr}(j_2|j_1) - \log_2 \mathrm{pr}(j_1) \end{aligned} \tag{15}$$

and the output entropy of the optimal direct-sum RVQ can be written as

$$H^*(J_1, J_2, \ldots, J_P) = \sum_{p=1}^{P} H(J_p|J_{p-1},\ldots,J_1).$$

Finally, assume the stage partitions $\{\mathcal{P}_1, \mathcal{P}_2 \ldots, \mathcal{P}_P\}$ are fixed. This implies the direct-sum partition $\boldsymbol{P}$ is fixed. Also, assume that the lengths $|\mathbf{L}(\boldsymbol{j})|$ of the channel codewords associated with the direct-sum code vectors are fixed. Then, rewrite (11) as

$$\begin{aligned} J_\lambda(\mathbf{E},\mathbf{L},\mathbf{D}) = \sum_{\boldsymbol{j}\in\boldsymbol{J}} \; & E\left\{d[\boldsymbol{x}_1, \mathbf{D}(\boldsymbol{j})] \mid \boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})\right\} \; \mathrm{pr}(\boldsymbol{j}) + \\ & \lambda \sum_{\boldsymbol{j}\in\boldsymbol{J}} E\left\{|\mathbf{L}(\boldsymbol{j})| \mid \boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})\right\} \; \mathrm{pr}(\boldsymbol{j}). \end{aligned} \tag{16}$$

Clearly, a mapping $\mathbf{D}$ that minimizes (16) is one that minimizes

$$\sum_{\boldsymbol{j}\in\boldsymbol{J}} E\left\{d[\boldsymbol{x}_1, \mathbf{D}(\boldsymbol{j})] \mid \boldsymbol{x}_1 \in \boldsymbol{V}(\boldsymbol{j})\right\} \; \mathrm{pr}(\boldsymbol{j}).$$

To achieve this minimum, the multistage code vectors $\boldsymbol{y}_p(j_p)$ at the pth stage must satisfy (8), i.e.,

$$\int d\left[\gamma_p, \boldsymbol{y}_p^*(j_p)\right] f_{\Gamma_p|j_p}(\gamma_p) d\gamma_p = \inf_{\boldsymbol{u}\in\Re^k} \int d(\gamma_p, \boldsymbol{u}) f_{\Gamma_p|j_p}(\gamma_p) d\gamma_p, \tag{17}$$

where $\gamma_p = \boldsymbol{x}_1 - \boldsymbol{g}(\beta_p(\boldsymbol{j}))$, and $f_{\Gamma_p|j_p}(\gamma_p)$ is defined by (9).

3.2 The EC-RVQ Design Algorithm

The EC-RVQ design algorithm proposed here is an iterative descent algorithm similar to the one used for the design of EC-VQ codebooks. Each iteration consists of applying the transformation

$$(\mathbf{E}(t+1),\mathbf{L}(t+1),\mathbf{D}(t+1)) = T(\mathbf{E}(t),\mathbf{L}(t),\mathbf{D}(t))$$

where

$$\begin{aligned}
\mathbf{E}(t+1) &= \arg\min_{\mathbf{E}}(\mathbf{E},\mathbf{L}(t),\mathbf{D}(t)) && \text{(optimum partitions)}\\
\mathbf{L}(t+1) &= \arg\min_{\mathbf{L}}(\mathbf{E}(t+1),\mathbf{L},\mathbf{D}(t)) && \text{(optimum codeword lengths)}\\
\mathbf{D}(t+1) &= \arg\min_{\mathbf{D}}(\mathbf{E}(t+1),\mathbf{L}(t+1),\mathbf{D}) && \text{(optimum code vectors)}
\end{aligned}$$

Following the lines of argument of [27], one can show that every limit point of the sequence $(\mathbf{E}(t),\mathbf{L}(t),\mathbf{D}(t))$, $t = 0,1,\ldots,$ generated by the transformation T minimizes the Lagrangian $J_\lambda(\mathbf{E},\mathbf{L},\mathbf{D})$ (as given by (11)). Therefore, the EC-RVQ design algorithm is guaranteed to converge to a local minimum.

To find several points on the convex hull of the operational rate-distortion curve, the minimization of $J_\lambda(\mathbf{E},\mathbf{L},\mathbf{D})$ is repeated for various λ's. Starting with $\lambda = 0$ (which corresponds to the RVQ codebook designed by the fixed rate RVQ design algorithm), the EC-RVQ design algorithm uses a pre-determined sequence of λ's to design locally optimal variable rate EC-RVQ codebooks.

For optimal performance, the EC-RVQ design algorithm must generally employ an exhaustive-search encoder, a jointly optimized direct-sum decoder, and an optimal entropy coder as described by (14). Unfortunately, the computational complexity and memory requirements associated with optimal EC-RVQs are usually prohibitive, and sub-optimal design procedures are usually used to generate practical EC-RVQs.

As with fixed rate RVQ design algorithms, the encoder does not necessarily have to be optimal to be useful. Sub-optimal tree-structured searching techniques such as stage-sequential searching or multipath searching can be employed, leading to relatively fast encoder implementations. Experimental results indicate that stage-sequential searching usually leads to a significant increase in average distortion, but multipath M-searching can result in a close-to-optimal performance, even with values of M as small as 2 or 3 [23, 15].

Ideally, all stage codebooks in the RVQ should be jointly optimized. However, since the complexity of the joint optimization design process increases rapidly (quadratically) with increasing number of stages, the RVQ design effort can become excessive. The complexity of the design can be greatly reduced by using conventional stage-sequential optimization, but the resulting performance can also be significantly reduced. The performance gap between sequential and joint optimization can be bridged by local joint optimization of the stage codebooks. The optimization is local in the sense that the stages are partitioned into overlapping blocks and the joint optimization process is restricted to only the stages of each block. This technique was previously employed to accelerate the design of large-block fixed rate RVQ codebooks with a relatively large number of stages [28]. However, we also note that, unlike fixed rate RVQ, EC-RVQ (with a modest number of stages) is shown experimentally to generally perform quite well when sequential stage-wise optimization is used. This encouraging result implies that, at moderate bit rates, the EC-RVQ design speed can be substantially increased without significantly impairing performance.

A unique complexity reducing feature of EC-RVQ is its potential to use *stage-conditional* (i.e., conditioned on previous stages) entropy tables of relatively small sizes. Equation (15) shows that the optimal length (given by (14)) of the variable length codeword associated with an index $\boldsymbol{j} \in \boldsymbol{J}$ is also the sum of P stage-conditional self-information components. During the design process, the lengths of the stage-conditional entropy codewords can be estimated by using a sufficiently large training set. Clearly, the aggregate number of tables of stage-conditional entropy codes can become extremely large as the number of stages increases, which may offset the memory savings obtained by using the RVQ structure. However, the number of tables can be made relatively small by limiting the number m of previous stages upon which conditioning is based. In other words, the direct-sum codeword length $|\mathbf{L}(\boldsymbol{j})|$ is approximated by

$$\begin{aligned}|\mathbf{L}(\boldsymbol{j})|_m &= -\log_2 \mathrm{pr}(j_P|j_{P-1},\ldots,j_{P-m}) - \log_2 \mathrm{pr}(j_{P-1}|j_{P-2},\ldots,j_{P-m}) \\ &\quad -\ldots - \log_2 \mathrm{pr}(j_2|j_1) - \log_2 \mathrm{pr}(j_1). \qquad (18)\end{aligned}$$

Obviously, since $H(J_p|J_{p-1},\ldots,J_1) \leq H(J_p|J_{p-1},\ldots,J_{p-m})$ for each $p = 1,2,\ldots,P$ and $m < p-1$, it is easy to show that $H_m(\boldsymbol{J}) = \sum_{p=1}^{P} H(J_p|J_{p-1},\ldots,J_{p-m}) \;\geq\; H(\boldsymbol{J})$. Experimental results show that the value of m that results in a good complexity/performance tradeoff increases with both increasing number of stages and vector size, but decreases with increasing stage codebook size. Recent results also show that the best value for m depends heavily on the source. For sources with memory, the best value of m is usually small ($0 \leq m \leq 2$). For memoryless sources, however, a larger value of m is usually needed for a good tradeoff, which results in increased memory requirements.

While the sub-optimal EC-RVQ design algorithms discussed above are not guaranteed to converge to local minima, they provide good complexity/performance tradeoffs, and they facilitate the design of practical EC-RVQs. We also point out that the sub-optimal algorithms employed in all EC-RVQ experiments performed in this work converged monotonically to a fixed point, and occasional deviations from monotonicity were observed only when stage-sequential searching was used during the encoding step of the EC-RVQ design.

4 Experimental Results

Many quantization techniques have been used to code Gaussian, Laplacian, and uniform memoryless sources, as well as Gauss-Markov sources. Table 1 shows some of the well-known coders as compared qualitatively with EC-RVQ in terms of encoder complexity and memory. For the class of VQ-based coders, EC-RVQ is less demanding in terms of both memory and encoding complexity. It has comparable encoding complexity and memory requirements to that of EC-TCQ but does not suffer from the relatively large coding delays associated with large trellises. Finally, it should be noted that when the dimension is one, EC-RVQ, or entropy-constrained residual scalar quantization (EC-RSQ), has the simplest encoding complexity and the smallest memory requirements.

In this paper we report on the relative performance of these coding techniques for memoryless Gaussian, Laplacian, and uniform sources as well as a Gauss-Markov source. Experimental results demonstrate the performance of EC-RVQ and show its advantages and disadvantages when compared to some of the competitive coding techniques that have appeared in the literature. In particular, EC-RVQ performance is compared to that of scalar quantization (SQ), entropy-constrained SQ (EC-SQ), entropy-constrained VQ (EC-VQ), trellis coded quantization (TCQ), entropy-constrained TCQ (EC-TCQ), and lattice-based VQs. For each of the sources considered here, the EC-RVQs, the EC-RSQs, and the EC-SQs, which are described in Table 2, were designed on training sequences rather than on the underlying distributions, and were used to encode a test sequence of 40,000 samples taken from the same source. The performance results for EC-VQ [16], TCQ [29], EC-TCQ, predictive EC-SQ (PEC-SQ), predictive EC-TCQ (PEC-TCQ) [30], and lattice-based VQs [31, 16] are taken from the literature.

Experimental results for a Gaussian random variable with zero mean and unit variance are shown in Figure 3 (top) and Table 3. Figure 3 (top) shows the rate-distortion performance for the various EC-RVQs and EC-SQ relative to the R(D) curve. Signal-to-noise ratio (SNR) values for EC-RVQ, EC-VQ, EC-SQ, D4 lattice, A2 lattice, TCQ, EC-TCQ, and R(D) at 0.5, 1.0, 1.5, and 2.0 bits per sample (bps) are given in Table 3. It can be seen that the performance of EC-RVQ increases with increased vector size, and that practical EC-RVQs outperform practical EC-VQs with the same vector size, even while maintaining relatively small encoding complexity and memory requirements. EC-RVQ is also competitive with both TCQ and EC-TCQ.

The next set of experiments considers the Laplacian source with zero mean and unit variance. Figure 3 (bottom) shows the rate-distortion performance of several EC-RVQs and EC-SQ relative to a curve linearly interpolated from well-known R(D) points. Numerical values are given in Table 4 for EC-RVQ, EC-RSQ, EC-SQ, TCQ, EC-TCQ, SQ, VQ, and R(D) at 0.5, 1.0, and 2.0 bps. Unlike the case of the Gaussian source, increasing the vector size does not improve the EC-RVQ rate-distortion performance significantly. This is explained by the fact that, as the vector size increases, encoding complexity and memory requirements limit the size of the initial codebook (or the *peak* bit rate) that can be used to design practical EC-RVQs. This leads to a reduction in rate-distortion performance because the Laplacian source (which has a peaked distribution) requires a very large output alphabet size (i.e., number of levels or code vectors), which is difficult to attain in practice. In fact, EC-RSQ is very competitive with EC-RVQ because the former has the potential to use an expanded set of direct-sum code vectors. When compared to other coding techniques, EC-RVQ (including the special case where the vector size k is equal to 1) outperforms the other coders at low bit rates and is competitive with EC-TCQ at high rates.

Simulation results for the encoding of a memoryless uniform source are shown in Figure 4 (top) with numerical values given in Table 5. As stated in [29], entropy coding does not lead to any performance gains in the case of scalar or trellis coded quantization. However, although the source is uniform, RVQ outputs are *generally* not equiprobable, and entropy coding usually leads to a slight performance gain. As can be seen, increasing the vector size leads to an increase in rate-distortion performance. However, EC-RVQ performance generally falls below that of TCQ [29], but becomes competitive when the the vector size is relatively large (e.g., $k = 16$).

Finally, results for a Gauss-Markov source with correlation coefficient $\rho = 0.9$ are shown in Figure 4 (bottom) and Table 6. Again, Figure 6 shows the rate-distortion performance of several EC-RVQs and EC-SQ relative to R(D) while Table 6 shows the SNRs for a number of predictive coding techniques as well as EC-RVQ and EC-VQ at bit rates of 0.5, 1.0, 1.5, 2.0 and 2.5 bps. It should be noted that for rates $R > 0.926$, the R(D) curve in Figure 4 (bottom) is actually an upper bound on the true derived R(D) curve. As expected, there is a clear advantage of VQ-based coders over most of the other non-predictive scalar coders. Although EC-VQ is expected to *theoretically* outperform all VQ-based coders for such a source, practical EC-VQs do not meet that expectation, mainly because the encoding complexity and memory requirements associated with such coders severely limit the initial codebook size (or the peak bit rate). In fact, EC-VQ is significantly outperformed by EC-RVQ with the same vector size. For vector sizes larger than 6, EC-RVQ outperforms PEC-SQ at all bit rates between 0.5 and 2.0 bits/sample, and is competitive with PEC-TCQ, especially at relatively large vector sizes (e.g., $k = 16$). It should be noted that the memory inherent in both the state and the predictor gives PEC-TCQ an effective vector size which is usually larger than the vector sizes used by the VQ-based coders.

5 Summary

Necessary conditions for optimal variable rate RVQ have been derived, and an iterative descent algorithm for designing locally optimal variable rate EC-RVQ codebooks has been introduced. The RVQ structure is exploited to facilitate the implementation of practical EC-RVQs, which perform well even while maintaining very low encoding complexity and memory requirements.

Experimental results for three memoryless sources and a Gauss-Markov source indicate that practical EC-RVQs have performance advantages over other VQ-based coders, including practical EC-VQs. Although EC-RVQ outperforms TCQ-based coders only at some relatively low bit rates for the Laplacian source, it is usually competitive and has the potential of increased rate-distortion performance when the peak bit rate is increased. Furthermore, encoding complexity and memory requirements of EC-RVQ are comparable to those of TCQ-based coders, but EC-RVQ does not have the disadvantage of the long encoding delays associated with large trellises.

References

[1] F. Kossentini, M. Smith, and C. Barnes, "Image coding using entropy-constrained RVQ," *Submitted to Transactions on Image Processing*, Apr. 1993.

[2] B. H. Juang and A. H. Gray, "Multiple stage vector quantization for speech coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 597–600, April 1982.

[3] R. L. Baker, *Vector Quantization of Digital Images.* PhD thesis, Stanford University, 1984.

[4] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proceedings of the IEEE*, vol. 73, pp. 1551–1581, Nov. 1985.

[5] C. F. Barnes, *Residual Quantizers.* PhD thesis, Brigham Young University, Provo, Utah, Dec. 1989.

[6] F. Kossentini, M. Smith, and C. Barnes, "Image coding with variable rate RVQ," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. III, (San Fransisco, CA, USA), pp. 369–372, Mar. 1992.

[7] F. Kossentini, M. Smith, and C. Barnes, "Finite-state residual vector quantization," *Journal of Visual Communication and Image Representation*, July 1993.

[8] F. Kossentini, M. Smith, and C. Barnes, "Entropy-constrained residual vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. V, (Mineapolis, MN, USA), pp. 598–601, Apr. 1993.

[9] W. Chan, S. Gupta, and A. Gersho, "Enhanced multistage vector quantization by joint codebook design," *IEEE Trans. on Communications*, vol. COM-40, pp. 1693–1697, Nov. 1992.

[10] W. Y. Chan, A. Gersho, and S. W. Soong, "Joint codebook design for summation product-code vector quantizers," in *Data Compression Conference*, (Snowbird, UT, USA), pp. 42–51, Mar. 1992.

[11] B. Bhattacharya, W. Leblana, S. Mahmound, and V. Cuperman, "Tree searched multistage vector quantization of LPC parameters for 4 kb/s speech coding," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 1, (San Fransisco, CA, USA), pp. 105–108, Mar. 1992.

[12] D. Miller and K. Rose, "An improved sequential search multistage vector quantizer," in *Data Compression Conference*, (Snowbird, UT, USA), pp. 12–21, Mar. 1992.

[13] J. A. Rodriguez-Fonollosa and E. Masgrau, "Adaptive multistage vector quantization," in *IEEE Proc. MELECON*, pp. 225–228, Apr. 1990.

[14] C. F. Barnes and R. L. Frost, "Necessary conditions for the optimality of residual vector quantizers," in *Proceedings of the IEEE International Symposium on Information Theory*, (San Diego, CA, USA), Jan. 1990.

[15] C. F. Barnes and R. L. Frost, "Vector quantizers with direct sum codebooks," *IEEE Trans. on Information Theory*, vol. 39, pp. 565–580, Mar. 1993.

[16] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-37(1), pp. 31–42, January 1989.

[17] T. Lookabaugh, E. A. Riskin, P. A. Chou, and R. M. Gray, "Variable rate vector quantization for speech, image, and video compression," *IEEE Trans. on Communications*, vol. 41, pp. 186–199, Jan. 1993.

[18] D. G. Luenberger, *Linear and Nonlinear Programming.* Menlo Park, California: Addison-Wesley Publishing Company, 1984.

[19] R. M. Gray, J. C. Kieffer, and Y. Linde, "Locally optimal block quantizer design," *Inform. and Control*, vol. 45, pp. 178–198, May 1980.

[20] F. Kossentini, M. Smith, and C. Barnes, "Locally optimal RVQ design algorithms: Convergence," *In preparation*, 1993.

[21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computations.* New Jersey: Prentice-Hall, Inc., 1989.

[22] F. Jelinek and J. B. Anderson, "Instrumentable tree encoding of information sources," *IEEE Transactions on Information Theory*, vol. IT-17, pp. 118–119, Jan. 1971.

[23] F. Kossentini, M. Smith, and C. Barnes, "Large block RVQ with mutipath searching," in *Proc. IEEE Int. Sym. Circuits and Systems*, vol. 5, (San Diego, CA, USA), pp. 2276–2279, May 1992.

[24] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE National Convention Record, Part 4*, pp. 142–163, 1959. Also in *Information and Decision Processes*, R. E. Machol, Ed. New York, NY: McGraw-Hill, 1960, pp. 93-126.

[25] T. Berger, *Rate Distortion Theory.* New Jersey: Prentice-Hall, Inc., 1971.

[26] R. G. Gallager, *Information Theory and Reliable Communication.* NY: John Wiley & Sons, 1968.

[27] M. J. Sabin and R. M. Gray, "Global convergence and empirical consistency of the generalized Lloyd algorithm," *IEEE Transactions on Information Theory,* vol. IT-32, pp. 148–155, Mar. 1986.

[28] F. Kossentini, M. Smith, and C. Barnes, "A perspective view of finite state binary residual VQ," in *Proc. IEEE Int. Sym. Circuits and Systems,* (Singapore), pp. 300–303, June 1991.

[29] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. on Communications,* vol. 38, pp. 82–93, Jan. 1990.

[30] T. R. Fisher and M. Wang, "Entropy-constrained trellis-coded quantization," *IEEE Trans. on Information Theory,* vol. 38, pp. 415–426, Mar. 1992.

[31] J. Conway and N. Sloane, "A lower bound on the average error of vector quantizers," *IEEE Trans. on Information Theory,* vol. IT-31, pp. 106–109, Jan. 1985.

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

List of Figures

List of Tables

System	Block(Vector) Size	Encoding	Memory	Entropy Coder
EC-SQ	1	simple	very small	very simple
EC-RSQ	1	very simple	very small	very simple
A2 Lattice	2	moderate	small	simple
D4 Lattice	4	moderate	small	simple
EC-VQ	4	complex	large	complex
EC-VQ	8	very complex	large	complex
EC-RVQ	4	simple	small	simple
EC-RVQ	8	moderate	small	simple
EC-RVQ	16	complex	moderate	moderate
EC-TCQ(s=8)	1	simple	small	simple
EC-TCQ(s=8)	4	moderate	small	moderate
EC-TCQ(s=128)	1	moderate	moderate	moderate

Table 1: Qualitative comparison of several entropy-coded quantization systems

	EC-RVQ					EC-RSQ	EC-SQ
	k=4	k=6	k=8	k=12	k=16		
TSS	250	300	400	500	750	200	200
NS	4	5	5	6	8	3	1
SCS	16	16	16	16	16	4	16
PBR	4.00	3.33	2.50	2.0	2.0	6.0	4.0
NSP	2	2	2	3	3	1	1
MMO	1	2	2	2	2	1	0
NVDC	128	160	160	288	384	12	16
CM	1.02	1.92	2.56	4.61	8.19	0.48	0.64
TM	0.39	6.28	6.28	8.33	12.42	0.09	0.08

Table 2: Training set size (TSS) in thousands of vectors, number of stages (NS), stage codebook size (SCS) in vectors, peak bit rate (PBR) in bits/sample, number of search paths (NSP), Markov model order (MMO), number of vector distortion calculations (NVDC) per input vector, codebook memory (CM) in kilobytes, and maximum memory requirements for entropy tables (TM) in kilobytes for EC-RVQ, EC-RSQ, and EC-SQ.

	EC-RVQ		EC-VQ	EC-SQ	D4	A2	TCQ	EC-TCQ	R(D)
Rate	k=4	k=12	k=4				s=256	s=128	
0.5	2.21	2.50	2.20	2.09	2.05	2.17	2.78	N/A	3.01
1.0	5.10	5.38	4.80	4.64	4.55	4.78	5.56	5.50	6.02
1.5	7.80	8.21	7.70	7.57	6.95	7.60	N/A	8.79	9.00
2.0	10.68	N/A	N/A	10.55	N/A	N/A	11.04	11.83	12.04

Table 3: Performance (SNR in dB) of various source coding schemes for the memoryless Gaussian source at 0.5, 1.0, 1.5, and 2.0 bits per sample.

	EC-RVQ		EC-RSQ	EC-SQ	TCQ	EC-TCQ	SQ	VQ	R(D)
Rate	k=4	k=6			s=256	s=128		k=6	
0.5	3.23	3.27	3.15	3.14	2.20	N/A	N/A	1.97	N/A
1.0	5.91	5.92	5.90	5.79	5.54	4.82	3.01	4.96	6.62
2.0	11.38	11.58	11.50	11.31	11.22	12.35	7.54	N/A	12.66

Table 4: Performance (SNR in dB) of various source coding schemes for the memoryless Laplacian source at 0.5, 1.0, and 2.0 bits per sample.

	EC-RVQ		EC-SQ	TCQ		SQ	R(D)
Rate	k=4	k=16		s=4	s=256		
0.5	3.12	3.20	3.08	2.84	3.24	N/A	N/A
1.0	6.27	6.39	6.04	6.22	6.58	6.02	6.79
2.0	12.27	12.79	12.08	12.62	13.00	12.04	13.21
3.0	18.58	N/A	18.10	18.83	19.23	18.06	19.42

Table 5: Performance (SNR in dB) of various source coding schemes for the memoryless uniform source at 0.5, 1.0, and 2.0 and 3.0 bits per sample.

	EC-RVQ			EC-VQ		PEC-SQ	PEC-TCQ	R(D)
Rate	k=4	k=6	k=16	k=4	k=8		s=8	
0.5	7.45	8.43	9.32	7.10	8.15	N/A	N/A	10.22
1.0	10.64	11.58	12.36	10.40	11.15	N/A	N/A	13.23
1.5	13.38	14.29	15.29	12.15	N/A	13.86	15.30	16.26
2.0	16.15	17.23	N/A	15.80	N/A	17.22	18.38	19.25
2.5	19.14	20.13	N/A	N/A	N/A	20.48	21.41	22.26

Table 6: Performance (SNR in dB) of various source coding schemes for the Gauss-Markov source at 0.5, 1.0, 1.5, 2.0 and 2.5 bits per sample.

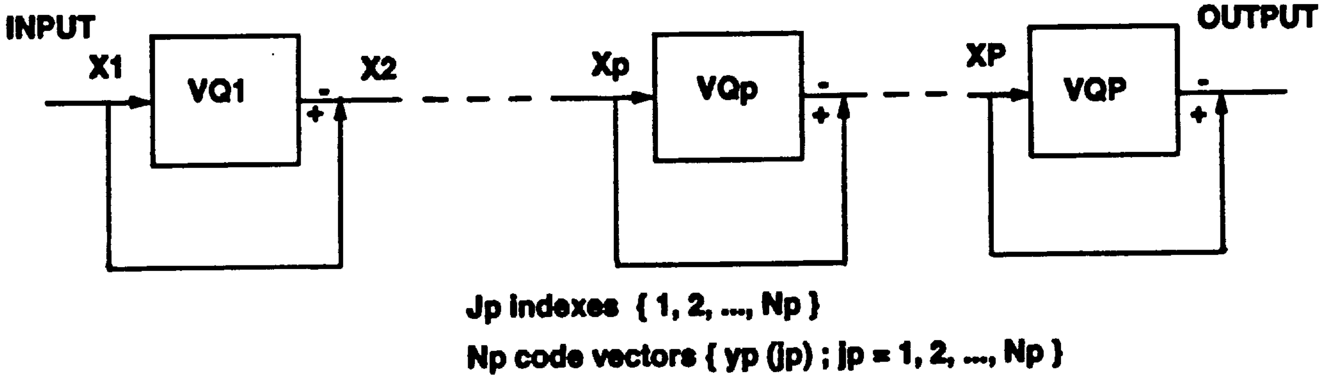
INPUT
X1
VQ1
-
+
X2
Xp
VQp
-
+
XP
VQP
-
+
OUTPUT
Jp indexes { 1, 2, ..., Np }
Np code vectors { yp (jp) ; jp = 1, 2, ..., Np }

Figure 1: A *P*-stage residual vector quantizer

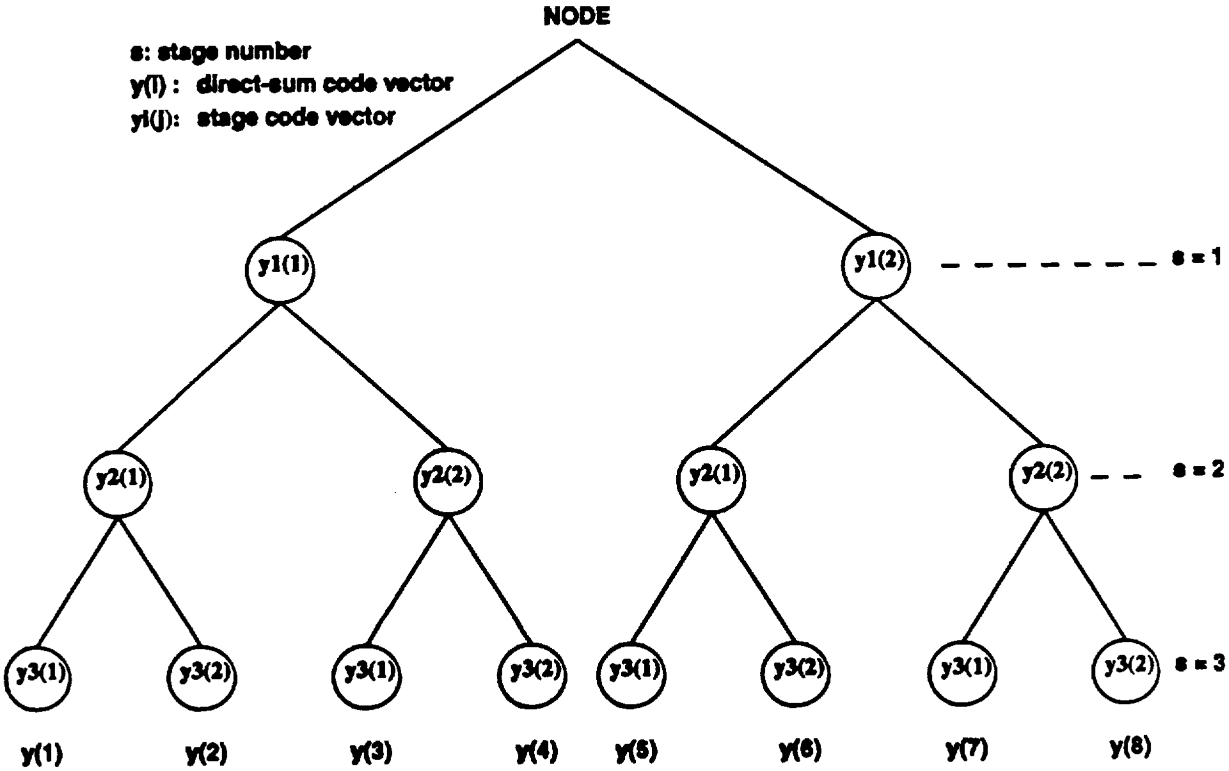

Figure 2: A 3-level RVQ tree

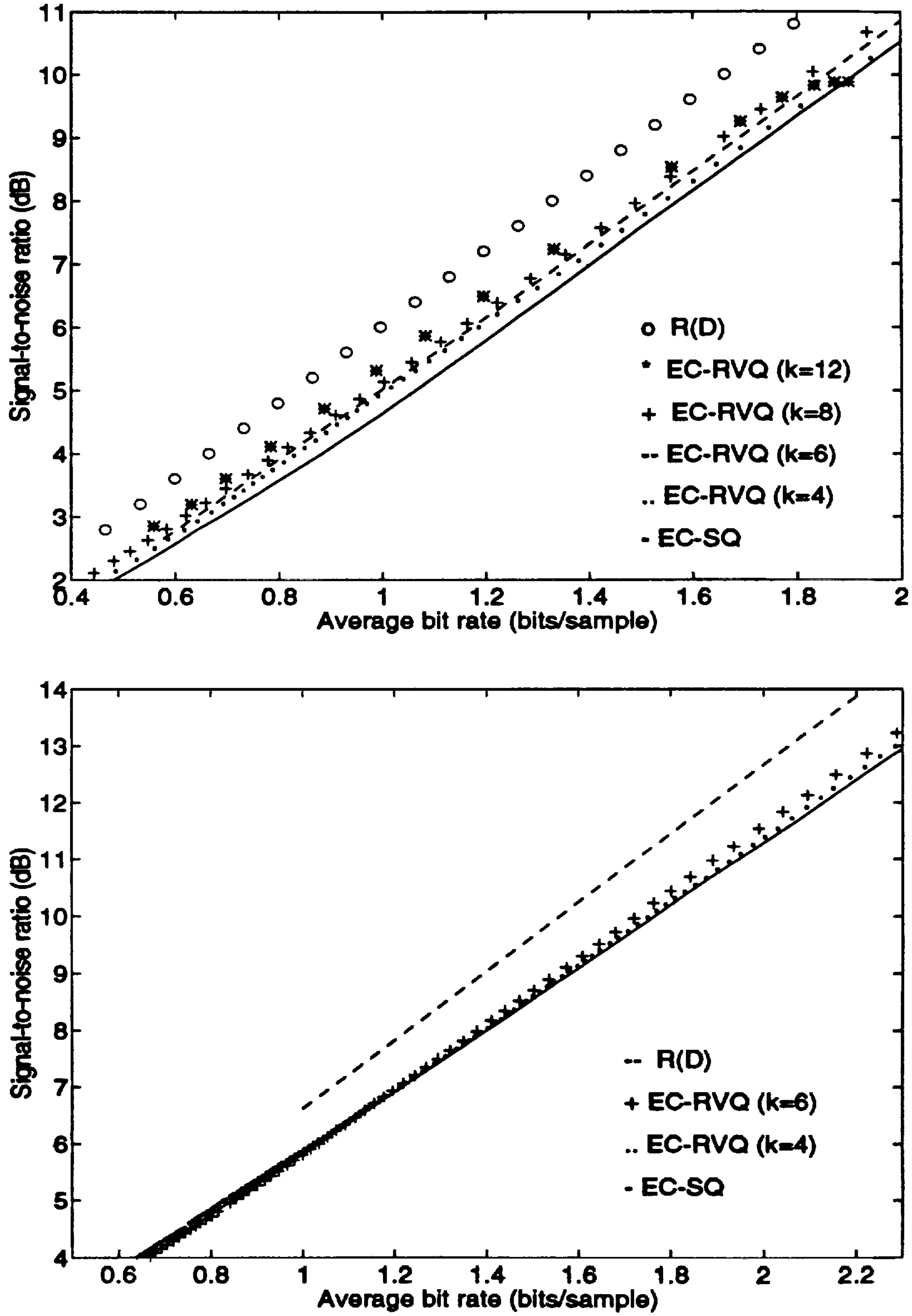

Figure 3: The R(D) performance of several EC-RVQs and EC-SQ relative to the true R(D) curve for the Gaussian (Top) and the Laplacian (Bottom) memoryless sources.

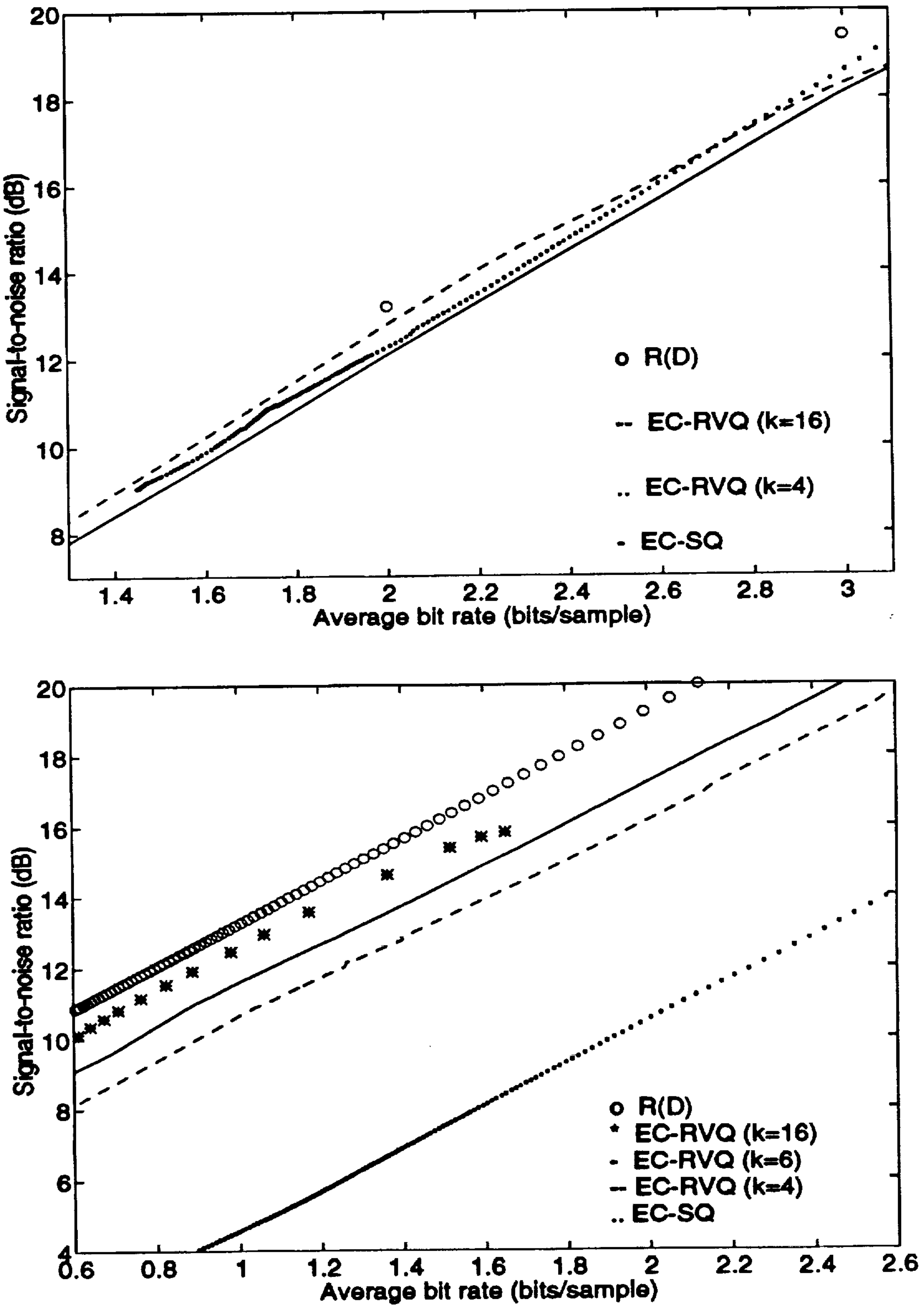

Figure 4: The R(D)performance of several EC-RVQs and EC-SQ relative to the true R(D) curve for the uniform source (Top) and the Gauss-Markov source (Bottom).