Entropy-Constrained Vector Quantization

PHILIP A. CHOU, TOM LOOKABAUGH, AND ROBERT M. GRAY, FELLOW, IEEE

Abstract—**An iterative descent algorithm based on a Lagrangian formulation is introduced for designing vector quantizers having minimum distortion subject to an entropy constraint. These entropy-constrained vector quantizers (ECVQ's) can be used in tandem with variable rate noiseless coding systems to provide locally optimal variable rate block source coding with respect to a fidelity criterion. Experiments on sampled speech and on synthetic sources with memory indicate that for waveform coding at low rates (about 1 bit/sample) under the squared error distortion measure, about 1.6 dB improvement in the signal-to-noise ratio can be expected over the best scalar and lattice quantizers when block entropy-coded with blocklength 4. Even greater gains are made over other forms of entropy-coded vector quantizers. For pattern recognition, it is shown that the ECVQ algorithm is a generalization of the *k*-means and related algorithms for estimating cluster means, in that the ECVQ algorithm estimates the prior cluster probabilities as well. Experiments on multivariate Gaussian distributions show that for clustering problems involving classes with widely different priors, the ECVQ outperforms the *k*-means algorithm in both likelihood and probability of error.**

I. Introduction

CONSIDER the quantization or data compression scheme of blocking a discrete-time stationary ergodic source into contiguous vectors of length n, and representing each vector by one of m reproduction vectors. For transmission of the source across a binary channel (or storage in a binary medium), the index of each reproduction vector can be encoded into binary in a straightforward manner requiring $(\log_2 m)/n$ bits per sample. For this encoding scheme, the natural design problem for a fixed block length n is to find the set of m reproduction vectors, known as the reproduction codebook, which minimizes the average distortion between the source and its reproduction, subject to a constraint on the maximum rate of bit transmission. Distortion-rate theory [1]–[3] guarantees that if n is sufficiently large, there exists a finite set of m reproduction vectors with transmission rate $R = (\log_2 m)/n$ bits per sample, such that the average distortion per sample D is arbitrarily close to the minimum distortion $D(R)$, over all possible compression schemes with average transmission rate less than or equal to R bits per sample. This provides a theoretical basis for the use of such vector quantizers; for sufficiently long block lengths, they are theoretically optimal. Unfortunately, practical schemes cannot use arbitrarily long block lengths. For a fixed block length of practical size, a scheme that achieves the same distortion with lower average transmission rate than straightforward binary coding of the indexes is one that "entropy encodes" its index sequence. Entropy encoding reduces the average transmission rate from $(\log_2 m)/n$ bits per sample to the index entropy or, with additional effort, even to the entropy rate of the index sequence. The key word here, however, is "average," since known practical schemes for achieving this lower rate all have a variable rate nature, and hence, an additional amount of complexity. If we are willing to deal with the added complexity, the natural design problem is to find the set of reproduction vectors which minimizes the average distortion between the source and its reproduction, subject to a constraint on the index entropy.

This is a classic problem in the quantization literature, often developing in parallel with the sister problem of quantizer design for minimum distortion with a fixed number of indexes. One broad area of inquiry has been based on calculus approximations that can be made if the probability distribution of the source is smooth and the number of indexes per dimension is large. Investigations in this "asymptotic" or "high resolution" quantization theory began with the scalar case ($n = 1$), in which regular quantizers composed of thresholds and scalar reproductions (the thresholds determining to which reproduction a given source symbol will be mapped) are generally the systems of interest. Gish and Pierce [4] showed, among other things, that the optimal high-resolution entropy-constrained scalar quantizer (ECSQ) i) has uniformly spaced thresholds regardless of the source probability density function and ii) for the Gaussian iid case and squared-error distortion has an index entropy only 0.255 bits/sample greater than the rate-distortion curve. The combination of these results suggests that scalar uniform quantization should be quite effective for this source—a result that had already been noted experimentally by Goblick and Holsinger [5]. Zador [6] developed bounds for the index entropy of a high-resolution entropy-constrained vector quantizer (ECVQ) ($n \geq 1$). Based on the fact that one of these bounds is optimized by a uniform distribution of reproductions, Gersho [7] conjectured the optimal high resolution ECVQ should have the form of a lattice (a lattice in $\boldsymbol{R}^n$ is composed of all integral combinations of a set of linearly independent vectors which span the space). Investigations of the properties of lattices suggest that certain lattices should perform better than others;

Manuscript received September 8, 1987; revised April 8, 1988. This work was supported by the National Science Foundation under NSF Grant IST-8509860; by a Hughes Doctoral Fellowship; by an IBM Doctoral Fellowship; and by ESL, a subsidiary of TRW Inc. This work was performed while all the authors were with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University.

P. A. Chou is with AT&T Bell Laboratories, Murray Hill, NJ 07974.

T. Lookabaugh is with Compression Labs, Inc., San Jose, CA 95134.

R. M. Gray is with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

IEEE Log Number 8824508.

0096-3518/89/0100-0031$01.00 © 1989 IEEE

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

Conway and Sloane [8]–[10] have determined the best known lattices for several dimensions as well as useful encoding and indexing schemes, and Sayood *et al.* have applied lattices to transform image coding [11].

A second avenue of investigation has centered on efforts to determine the optimal ECSQ directly (without high-resolution approximations). Expressions for the entropy and distortion performance of ECSQ typically also include as a parameter the number of reproductions m. Ideally, we would like to find the optimum performance over all m; however, in many analytic approaches and all practical implementations, it is in fact necessary to limit m to some maximum value. Wood [12] provided the first numerical ECSQ design, using a descent algorithm to show that the optimal ECSQ for the Gaussian iid case was only slightly superior to the uniform scalar quantizer even for moderate values of m. Berger [13] described the necessary conditions for an optimal ECSQ under squared-error distortion (and later for the rth power distortion [14]), and first presented the Lagrangian formulation that we will make use of in Section II. Based on this formulation, he developed an iterative algorithm for determining thresholds of the ECSQ. Netravali and Saigal [15] presented another ECSQ design algorithm based on fixed-point considerations. Noll and Zelinski [16] applied Berger's algorithm to a variety of common probability density functions. Farvardin and Modestino [17] extended the necessary conditions for optimality to general distortion measures, and outlined two algorithms for ECSQ design. The first, which they actually used in their experiments, is similar to Berger's algorithm. The second is a fixed-point algorithm which can be considered the scalar version of the ECVQ design algorithm of this paper. Although they reported difficulties with the convergence of the algorithm, we encountered no such difficulties in the present investigation. For the squared-error distortion, Ziv [18] presented an interesting result which shows that a certain dithered uniform scalar quantizer followed by entropy coding of a block of n indexes will have index entropy (conditioned on the dithering variable) within 0.754 bits/sample of the optimal ECVQ of block length n; Gutman [19] extended the result to a variety of other distortions. Unfortunately, this bound is loose at low rates, and tighter bounds from the high-resolution theory are available at high rates.

In this paper we present a descent algorithm for the ECVQ design problem. The algorithm begins with a Lagrangian formulation like Berger's, but in implementation is quite similar to the generalized Lloyd algorithm [20], [21] developed for the constrained number-of-indexes vector quantizer design problem. We conduct a variety of experiments based on the use of a training sequence rather than the integration of probability density functions characteristic of previous investigations. We discover that, generally at the cost of higher complexity, ECVQ outperforms many other entropy-coded quantization schemes including scalar uniform threshold, lattice, constrained number-of-indexes vector quantization, and a recently introduced tree based variable rate vector quantization scheme [22]; performance gains are especially significant for sources with memory such as speech. Finally, we explore applications of the algorithm in clustering and classifier design for pattern recognition.

II. Theoretical Background

Consider the variable rate communications system shown in Fig. 1. In a typical implementation, the encoder blocks the source $\{X_i\}$ into vectors of length n, and maps each vector $X^n \in \mathcal{X}^n$ into a variable-length codeword $c \in \mathcal{C}$. The codeword c is then sent through the channel, which is assumed to be noiseless, and the decoder maps the codeword into a reproduction vector $Y^n \in \mathcal{Y}^n$. The reproduction alphabet $\mathcal{Y}$ is usually, but not necessarily, equal to the source alphabet $\mathcal{X}$, but, in general, the two alphabets are arbitrary and need not be finite or even countable. The channel codeword alphabet Σ is finite, and is usually taken to be binary, $\Sigma = \{0, 1\}$. This shall be assumed throughout the remainder of the paper. The set of channel codewords $\mathcal{C} = \{c_i\}_{i \in \mathcal{I}}$ is known as the channel codebook, and is a subset of Σ^*, the set of all finite length strings from Σ. Thus, $\mathcal{C}$ is finite or countable depending on its index set $\mathcal{I}$, and contains codewords of varying lengths, in general.

Let $\boldsymbol{\alpha}_n$: $\mathcal{X}^n \to \mathcal{C}$ and $\boldsymbol{\beta}_n$: $\mathcal{C} \to \mathcal{Y}^n$ denote the encoder and decoder, respectively. It is possible to decompose the encoder into two parts, $\boldsymbol{\alpha}_n = \gamma \circ \alpha$, where α: $\mathcal{X}^n \to \mathcal{I}$ and γ: $\mathcal{I} \to \mathcal{C}$, and $\circ$ denotes composition. The first part, α, maps a source vector into an index, and is many-to-one and information lossy, in general. The second part, γ, maps the index into a channel codeword, and is one-to-one and information lossless.

Similarly, the decoder can be decomposed as $\boldsymbol{\beta}_n = \beta \circ \gamma^{-1}$, where γ^{-1}: $\mathcal{C} \to \mathcal{I}$ and β: $\mathcal{I} \to \mathcal{Y}^n$. The mapping γ^{-1} is the inverse of γ, and maps each channel codeword back to its index. The mapping β outputs a reproduction vector for an index.

The pair $(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)$ will be referred to as a variable rate block coder because a block of input is represented by a single codeword in $\mathcal{C}$, and the codewords in $\mathcal{C}$ are generally not all of the same length.

To be information lossless, it is not enough that γ be invertible; it must also be *uniquely decodable*. This means that if c is a string in Σ^*, and c is the concatenation of codewords $\gamma(i_1), \cdots, \gamma(i_m)$ and also the concatenation of codewords $\gamma(i'_1), \cdots, \gamma(i'_{m'})$, then $m = m'$ and $i_1 = i'_1, \cdots, i_m = i'_m$. It can be shown [2, p. 49] that if the mapping γ: $\mathcal{I} \to \mathcal{C}$ is uniquely decodable, then there exists a *prefix-free* mapping γ': $\mathcal{I} \to \mathcal{C}'$ with the same codeword lengths. That is, there is a codebook $\mathcal{C}' = \{c'_i\}_{i \in \mathcal{I}}$ in which no codeword is the prefix of any other codeword, and in which $|c'_i| = |c_i|$ for all $i \in \mathcal{I}$, where $|c|$ denotes the length of the codeword c. Without loss of generality, then, γ will be assumed to be prefix-free. This places a constraint on the minimum average codeword length, as we shall see.

Assume the source $\{X_i\}_{i=0}^{\infty}$ is a strictly stationary ergodic random process with process distribution P. In what follows, the strict stationarity condition could be replaced

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

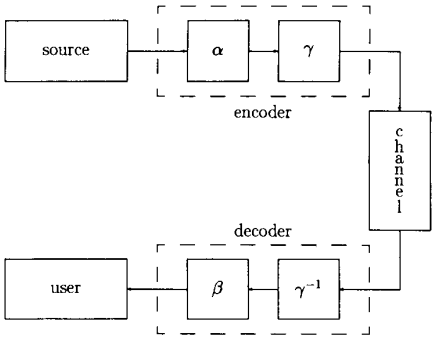


Fig. 1. A variable rate communications system.

by an asymptotic mean stationarity condition, but strict stationarity will be assumed for simplicity.

Let X^n denote the vector $(X_0, \cdots, X_{n-1})$, and let $Y^n = \boldsymbol{\beta}_n(\boldsymbol{\alpha}_n(X^n))$ be its reconstruction. If $l_n(X^n) = |\boldsymbol{\alpha}_n(X^n)|$ is the length of the codeword representing X^n, then

$$R_n = \frac{1}{n} E\big[l_n(X^n)\big]$$

is the average *rate*, or average number of channel symbols (here, bits) used to represent each source symbol. If $\rho_n(X^n, Y^n)$ is the distortion between the source vector X^n and its reproduction Y^n, then

$$D_n = \frac{1}{n} E\big[\rho_n(X^n, Y^n)\big]$$

is the average distortion between each source letter and its reproduction.

According to distortion-rate theory [2], [3], [23], if the distortion measure $\rho_n(x^n, y^n)$ and the source distribution P_{X^n} are fixed, then the nth-order distortion-rate function

$$D_n(R) = \inf_{P_{Y^n|X^n}} \left\{ \frac{1}{n} E\big[\rho_n(X^n, Y^n)\big] \middle| \frac{1}{n} I(X^n; Y^n) \le R \right\}$$

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

$$\hat{D}_n(R) = \inf_{(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)} \left\{ \frac{1}{n} E\big[\rho_n(X^n, Y^n)\big] \middle| \frac{1}{n} E\big[l_n(X^n)\big] \le R \right\}.$$

If ρ_n is a single-letter fidelity criterion, that is, if $\rho_n(x^n, y^n) = \sum_{i=0}^{n-1} \rho_1(x_i, y_i)$, then as $n \to \infty$, the bound becomes tight, so that the distortion-rate function

$$D(R) = \lim_{n \to \infty} D_n(R)$$

characterizes the region in the distortion-rate plane achievable by some deterministic variable rate block code $(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)$.

The nth-order distortion-rate function $D_n(R)$ is nonincreasing and convex, and its computation is a straightforward convex programming problem. In the case where the source and reproduction alphabets are finite, the elegant Blahut algorithm [24] can be used to compute $D_n(R)$.

The Blahut algorithm and more general convex programming algorithms rely on minimizing the Lagrangian

$$J(P_{Y^n|X^n}) = E\big[\rho_n(X^n, Y^n)\big] + \lambda I(X^n; Y^n),$$

where the Lagrange multiplier λ has an interpretation as the slope of a line supporting the graph of $D_n(R)$.

The nth-order *operational* distortion-rate function $\hat{D}_n(R)$ is nonincreasing, but it is unfortunately not necessarily convex or even continuous, particularly if P has a discrete component, e.g., if P is a sample distribution from a training sequence. (See Fig. 2.) Hence, Lagrangian methods cannot be used to find $\hat{D}_n(R)$. However, they can be used to find the *convex hull* of $\hat{D}_n(R)$ by minimizing the functional

$$J(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n) = E\big[\rho_n(X^n, Y^n)\big] + \lambda E\big[l_n(X^n)\big]. \quad (1)$$

Here, λ has an interpretation as the slope of a line supporting the convex hull.

Fortunately, any point on the convex hull can be achieved by timesharing between two variable rate block coders $(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)$ and $(\boldsymbol{\alpha}'_n, \boldsymbol{\beta}'_n)$, that achieve points on $\hat{D}_n(R)$. In practice, timesharing is usually not necessary because there often exists a single variable rate block coder $(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)$ that achieves a point on $\hat{D}_n(R)$ and on its convex hull with average rate (or distortion) sufficiently close to the desired average rate (or distortion).

In any case, our objective is to find the convex hull of $\hat{D}_n(R)$, and to find variable rate block coders $(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)$ achieving points on the convex hull. These coders are in some sense optimal, and can be used in practical data compression systems. We solve this problem by explicitly minimizing the Lagrangian functional (1).

III. The ECVQ Algorithm

Recall that $\boldsymbol{\alpha}_n = \gamma \circ \alpha$ and $\boldsymbol{\beta}_n = \beta \circ \gamma^{-1}$, where $\alpha: \mathfrak{X}^n \to \mathcal{I}$ quantizes an input vector into an index, $\gamma: \mathcal{I} \to \mathcal{C}$ noiselessly encodes the index into a binary string for transmission across the channel, $\gamma^{-1}: \mathcal{C} \to \mathcal{I}$ decodes the string back into its original index, and $\beta: \mathcal{I} \to \mathcal{Y}^n$ reproduces the index as an output vector which represents the original input vector in a minimum distortion sense.

Hence, we may rewrite the coder $(\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n)$ as (α, γ, β), and reexpress the Lagrangian functional (1) as

$$J_\lambda(\alpha, \gamma, \beta) = E\Big[\rho_n\big(X^n, \beta(\alpha(X^n))\big) + \lambda \big|\gamma(\alpha(X^n))\big|\Big]. \quad (2)$$

Our objective is to find the coder (α, γ, β) which minimizes this functional.

We employ an iterative descent algorithm similar to the generalized Lloyd algorithm. Starting with an arbitrary initial coder $(\alpha^{(0)}, \gamma^{(0)}, \beta^{(0)})$, we repeatedly apply a transformation

$$(\alpha^{(t+1)}, \gamma^{(t+1)}, \beta^{(t+1)}) = T(\alpha^{(t)}, \gamma^{(t)}, \beta^{(t)})$$

such that $J_\lambda(\alpha^{(t)}, \gamma^{(t)}, \beta^{(t)})$ is decreasing in t.

Since $J_\lambda(\alpha, \gamma, \beta)$ is bounded below by zero, the sequence of real numbers $J^{(t)} = J_\lambda(\alpha^{(t)}, \gamma^{(t)}, \beta^{(t)})$ is guar-

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

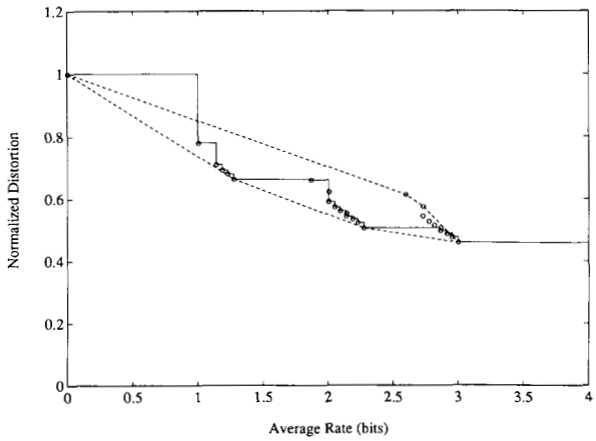


Fig. 2. A typical operational distortion-rate function $\hat{D}(R)$ when the underlying probability measure is discrete. Each circle is a (rate, distortion) pair; the dashed line is their convex hull. The solid staircase line is the operational distortion-rate function.

anteed to converge. As a convergence criterion for the algorithm, we use the simple stopping rule

$$(J^{(t)} - J^{(t+1)})/J^{(t+1)} < \epsilon,$$

where $\epsilon = 0.005$ is typical.

Following the lines of argument of Sabin and Gray [25], it appears that under suitable regularity conditions, the convergence of the coders $(\alpha^{(t)}, \gamma^{(t)}, \beta^{(t)})$ themselves (to a set) is also guaranteed, in a well-defined sense. This aspect, however, will not be investigated in this paper.

The transformation T operates as follows. For fixed $\gamma^{(t)}$ and $\beta^{(t)}$, $\alpha^{(t+1)}$ is chosen to minimize $J_\lambda(\alpha^{(t+1)}, \gamma^{(t)}, \beta^{(t)})$. Then, for fixed $\alpha^{(t+1)}$ and $\beta^{(t)}$, $\gamma^{(t+1)}$ is chosen to minimize $J_\lambda(\alpha^{(t+1)}, \gamma^{(t+1)}, \beta^{(t)})$. Finally, for fixed $\alpha^{(t+1)}$ and $\gamma^{(t+1)}$, $\beta^{(t+1)}$ is chosen to minimize $J_\lambda(\alpha^{(t+1)}, \gamma^{(t+1)}, \beta^{(t+1)})$. This procedure guarantees that $J^{(t)}$ is nonincreasing.

Let us consider the minimizations in more detail.

Fix γ and β, thereby fixing the reproduction codewords and the lengths of their associated channel codewords. A mapping $\alpha: \mathcal{X}^n \to \mathcal{I}$ that minimizes (2) is one that minimizes the integrand $\rho_n(x^n, \beta(\alpha(x^n))) + \lambda|\gamma(\alpha(x^n))|$ almost everywhere. That is, α maps x^n to the ith cell if the distance from x^n to $\beta(i)$, biased by the length of $\gamma(i)$, is minimized. Thus,

$$\alpha(x^n) = \operatorname*{argmin}_{i \in \mathcal{I}} \left[\rho_n(x^n, \beta(i)) + \lambda|\gamma(i)|\right]. \quad (3)$$

This α need not be unique since ties may be broken arbitrarily and (3) need hold only for almost all x^n. Equation (3) is analogous to nearest neighbor encoding in standard vector quantization.

Next express (2) as

$$J_\lambda(\alpha, \gamma, \beta) = \sum_{i \in \mathcal{I}} p(i)\, E\big[\rho_n(X^n, \beta(i)) + \lambda|\gamma(i)|\,\big|\,\alpha(X^n) = i\big], \quad (4)$$

where $p(i) = P\{\alpha(X^n) = i\}$.

For fixed α and β, and hence for fixed p, a prefix-free code $\gamma: \mathcal{I} \to \mathcal{C}$ that minimizes (4) is one that minimizes the expected codeword length $R = \Sigma_{i \in \mathcal{I}}\, p(i)|\gamma(i)|$. A prefix-free code of minimum expected codeword length can be found, in the case of finite $\mathcal{I}$, by the Huffman algorithm [26], for example. The Huffman algorithm could be incorporated into the design algorithm at this stage. However, for simplicity, we allow the fiction that codewords can have noninteger lengths, and assign

$$|\gamma(i)| = \log_2 (1/p(i)). \quad (5)$$

The "average rate" of the resulting code is exactly equal to its index entropy. Experiments have shown that following this system with a Huffman code produces an overall system nearly identical in performance to a system produced by a procedure that includes the Huffman algorithm within the design loop. Indeed, the average rate of a Huffman code must satisfy bounds such as $H(p) \le R \le H(p) + \max_i p(i) + 0.086$ [27], where $H(p) = \Sigma_{i \in \mathcal{I}}\, p(i) \log_2 (1/p(i))$ is the index entropy. On the other hand, experiments have also shown that if the Huffman code is to be followed by a buffering scheme, it is best to incorporate the code and the buffering scheme into the design loop, so that the resulting codebook will be optimized for the buffer. Normally, however, we use (5) so as not to tie our results to a particular entropy code, since there are a number of noiseless codes, e.g., arithmetic codes [28], [29] and Ziv–Lempel codes [30], that also achieve average rates quite close to the codeword entropy.

The last step in the transformation T is to fix α and γ, and hence p. Then the mapping $\beta: \mathcal{I} \to \mathcal{Y}^n$ that minimizes (4) is one that for each $i \in \mathcal{I}$, minimizes $E[\rho_n(X^n, \beta(i)) + \lambda|\gamma(i)|\,|\,\alpha(X^n) = i]$, i.e.,

$$\beta(i) = \operatorname*{arg\,min}_{y^n \in \mathcal{Y}^n} E\big[\rho_n(X^n, y^n)\,\big|\,\alpha(X^n) = i\big].$$

The reproduction codeword $\beta(i)$ is known as the *centroid* of the cell $\{\alpha(X^n) = i\}$, and is the same as the centroid of a cell in standard vector quantization; it need not be unique. Computing the centroid can be performed easily for a number of common distortion measures [21]. For example, if ρ_n is the squared-error distortion $\rho_n(x^n, y^n) = \Sigma_{i=0}^{n-1} (x_i - y_i)^2$, then the centroid is the conditional average $\beta(i) = E[X^n\,|\,\alpha(X^n) = i]$.

A summary of the algorithm is given in Fig. 3.

When the distribution is unknown, it can be estimated from the sample $X_0^n, X_1^n, \cdots, X_{N-1}^n$, where the vector $X_k^n = (X_{nk}, X_{nk+1}, \cdots, X_{n(k+1)-1})$ is the kth block from the original source $\{X_i\}_{i=0}^{\infty}$. The sample distribution

$$P_{X^n}^N(F) = \frac{1}{N} \sum_{k=0}^{N-1} 1_F(X_k^n)$$

converges on events F almost everywhere since P is stationary. Furthermore, if P is n-ergodic, $P_{X^n}^N$ converges to P_{X^n} in distribution. Here, $1_F(x) = 1$ if $x \in F$ and 0 otherwise. The algorithm for this case is identical to the algorithm of Fig. 3, with probabilities and expectations replaced by sample averages in the obvious way. For $\lambda =$

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

(0) Initialization: Given
a distribution P_{X^n},
a distortion measure ρ_n,
a Lagrange multiplier λ,
a convergence threshold ϵ,
an index set $\mathcal{I}$,
an initial reproduction codebook $\{\beta^{(0)}(i)\}_{i\in\mathcal{I}}$,
and initial codeword lengths $\{\gamma^{(0)}(i)\}_{i\in\mathcal{I}}$,
set $t = 0$ and $J^{(0)} = \infty$.

(1) $\alpha(x^n) = \text{argmin}_{i\in\mathcal{I}}\rho_n(x^n, \beta(i)) + \lambda|\gamma(i)|$.

(2) $|\gamma(i)| = -\log_2 P_{X^n}\{\alpha(X^n) = i\}$.

(3) $\beta(i) = \text{argmin}_{y^n\in\mathcal{Y}^n} \mathrm{E}\,[\rho_n(X^n, y^n)\,|\alpha(X^n) = i]$.

(4) $J^{(t+1)} = \mathrm{E}[\rho_n(X^n, \beta^{(t+1)}(\alpha^{(t+1)}(X^n))) + \lambda|\gamma^{(t+1)}(\alpha^{(t+1)}(X^n))|]$.

(5) If $(J^{(t)} - J^{(t+1)})/J^{(t+1)} > \epsilon$,
set $t = t + 1$ and go to (1).
Otherwise, quit.

Fig. 3. Descent algorithm to minimize the Lagrangian functional $J_\lambda(\alpha, \beta, \gamma)$ over all variable rate block coders.

0, the algorithm reduces to the generalized Lloyd algorithm [21].

When the algorithm converges, we obtain the variable rate coder (α, γ, β) that (locally) minimizes the functional

$$\frac{1}{n} J_\lambda(\alpha, \gamma, \beta) = D(\alpha, \gamma, \beta) + \lambda R(\alpha, \gamma, \beta),$$

where $D(\alpha, \gamma, \beta) = (1/n)E[\rho_n(X^n, \beta(\alpha(X^n)))]$ is the average distortion per letter, and $R(\alpha, \gamma, \beta) = (1/n)E[|\gamma(\alpha(X^n))|]$ is the average codeword length per letter. Thus, the line in the distortion-rate plane with slope $-\lambda$ that passes through the point $(R(\alpha, \gamma, \beta), D(\alpha, \gamma, \beta))$ supports the convex hull of the operational distortion-rate function (modulo the assumption that the minimization was global).

Therefore, to find the entire convex hull, it is necessary to repeat the minimization of $J_\lambda(\alpha, \gamma, \beta)$ for various λ's.

Two points on the convex hull are easy to find. The first point (R_∞, D_∞) is obtained by minimizing $J_\lambda(\alpha, \gamma, \beta)$ in the limit of large λ, hence, the notation R_∞ and D_∞. This point corresponds to the rate-zero codebook, in which there is only one channel codeword, the empty string. In this case, $\alpha(X^n) = 0$, $|\gamma(0)| = 0$, and $\beta(0) = \text{argmin}_{y^n\in\mathcal{Y}^n} E[\rho_n(X^n, y^n)]$. Hence, $R_\infty = 0$ and $D_\infty = (1/n)E[\rho_n(X^n, \beta(0))]$, where $\beta(0)$ is the centroid for the entire distribution.

The second point (R_0, D_0), obtained at $\lambda = 0$, corresponds to the full-rate codebook designed by the generalized Lloyd algorithm. For this point to have a finite average rate and positive distortion, we insist that the index set be finite, say $\mathcal{I} = \{0, 1, \cdots, m-1\}$. This constrains the rate (or resolution) of the code to be at most $1/n \log_2 m$. The average length R_0, or the entropy, of the code will of course be less than this. One could say that our resulting codes are both *rate* and *entropy* constrained. The distortion D_0 for the code is computed by the generalized Lloyd algorithm.

One of many ways to obtain the remainder of the points on the convex hull is to "walk up" the curve starting from the point (R_0, D_0), which corresponds to $\lambda = 0$. The first point visited is the point (R_1, D_1) which minimizes $J_{\lambda_1} = D + \lambda_1 R$, where $-\lambda_1 = \epsilon(D_0 - D_\infty)/(R_0 - R_\infty)$ is some fraction of the slope between (R_∞, D_∞) and (R_0, D_0). The next point visited (R_2, D_2) corresponds to some slope λ_2, etc. Each λ_i in the increasing sequence $\lambda_1, \lambda_2, \cdots$, can be determined *a priori*, e.g., by a geometric sequence, or it can be determined from a prediction of the behavior of the convex hull up to that point. In our experiments, the formula used was

$$\lambda_{n+1} = \frac{\lambda_n - \lambda_{n-1}}{R_{n-1} - R_n}\frac{R_0}{M}, \qquad n \geq 1, \tag{6}$$

with λ_0 and λ_1 initialized as above, and M being the desired number of (equally spaced in R) points on the curve. This procedure generally performed quite well, although in some experiments (on synthetic data such as the Gaussian memoryless source), R was extremely sensitive to λ so that backtracking was necessary. A computational advantage is accrued by using the final codebook for λ_n as the starting codebook for λ_{n+1}: in an experiment, we realized a savings in computation time of a factor of 2.4 over the alternative of using the λ_0 codebook to initialize the algorithm for each λ.

One issue of some importance is the complexity of the algorithm. Each pass through steps 1)–4) of the training sequence based algorithm of Fig. 3 requires roughly $3Nnm$ additions, Nnm multiplications, and m logarithms, with N the training sequence size in vectors, n the vector dimension, and m the codebook size. For M of (6) chosen as 20, we found that in the experiments reported in the next section, convergence for each λ took an average of three passes giving a total effort of about 60 times the "per pass" figures.

IV. Experimental Results

In this section we examine the implementation of the ECVQ design algorithm and compare its performance against several other entropy-coded quantization systems. Implementational issues include the effect of codebook size and the effect of placing specific entropy coding schemes inside and outside the design loop. Experimental comparisons were made against scalar uniform quantizers with entropy coding of single quantizer outputs and blocks of outputs, two- and four-dimensional lattice quantizers, full search, complete tree structured, and entropy-pruned tree structured vector quantizers (all with entropy-coded outputs).

In Fig. 4 we show the signal-to-quantization noise ratio (SQNR) curves for squared-error distortion for an eight-dimensional ECVQ as we varied the codebook size. Not surprisingly, the algorithm performance improves as codebook size increases. The training sequence was 20 000 samples of speech data, and the performance for this example was measured on the training sequence.

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

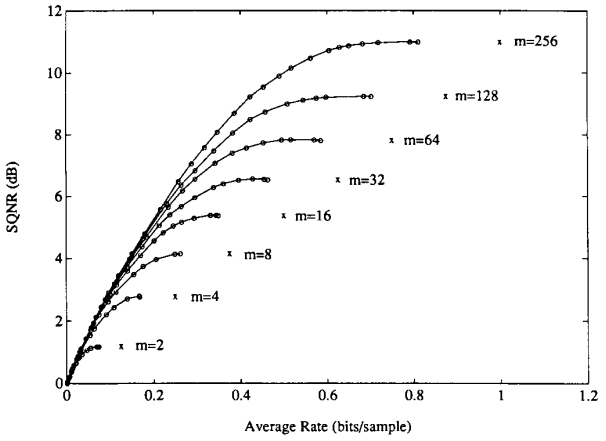


Fig. 4. ECVQ performance versus codebook size. Solid lines and circles are ECVQ performance as λ is varied; the x's are full search VQ performance without entropy coding. Codebook size is given by m.

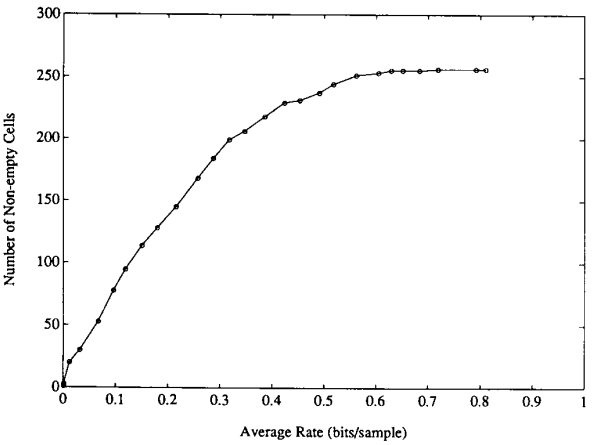


Fig. 5. Effective codebook size for the speech coding experiment of Fig. 4, with initial codebook size $m = 256$.

The algorithm itself generally reduces the codebook size as λ is increased, since during the course of the algorithm, a particular cell (say the ith cell) may become unpopulated (i.e., $\alpha(X_k^n)$ never equals i during step 1) of the algorithm). In this case, step 2) will set $|\gamma(i)| = \infty$ so that the cell will never be chosen as the best codeword by an encoder, or equivalently will never become populated as the design algorithm continues. Unlike the generalized Lloyd algorithm, there is no convincing rationale for attempting to repopulate empty cells; splitting a highly populated cell may indeed reduce distortion, but it may also increase the entropy of the quantizer. Hence, we remove empty cells from the codebook and continue the algorithm with the smaller codebook size. Fig. 5 shows the *effective* codebook size versus average rate for the experiment of Fig. 4 with an initial codebook size of 256.

In Fig. 6 we demonstrate the fact (mentioned in Section III) that not integerizing the codeword lengths inside the design loop leads to only a minor penalty in performance compared to the case where we replace step 2) of the design algorithm with an actual Huffman code design to determine the best integral word lengths. Not integerizing the word lengths, which leads to a minimum output entropy for α, is simpler and also advantageous when contemplating other entropy coding algorithms for γ.

On the other hand, including an entropy encoding scheme within the design loop tailors the ECVQ quantizer to that particular scheme. Consider a system in which the variable rate code is used over a fixed rate channel, so that buffering becomes necessary. Although the decoder can track the encoder buffer state if the channel code is prefix-free, buffer overflows and underflows cause a decrease in the effective average bit rate. This causes a decrease in SQNR at the nominal bit rate. By explicitly incorporating the overflow and underflow buffer strategies into the ECVQ design loop, the codewords can be designed to compensate for overflows and underflows. The standard strategy for handling overflows (for a zero mean

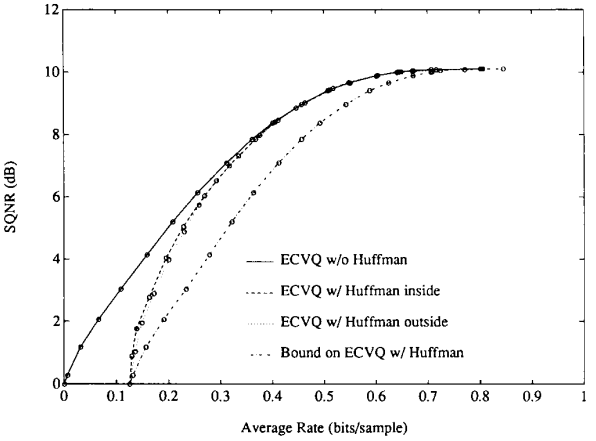


Fig. 6. Effect of Huffman coding inside and outside the ECVQ design algorithm. The solid line is ECVQ performance without Huffman coding (i.e., with noninteger word lengths), the dashed line is performance with Huffman coding inside the design algorithm, the dotted line is performance with a Huffman code applied after the ECVQ algorithm has finished. The dot-dash line shows a bound on Huffman code redundancy due to Gallager [27].

source) is to output a block of zeros instead of the reproduction codeword whose channel codeword would have overflowed the buffer. A better strategy, due to Berger *et al.* [31] and called buffer adapted Huffman (BAH) coding, is to output the best reproduction codeword possible from among those whose channel codewords fit into the remaining buffer space. The standard strategy for handling underflows is to pad the channel codeword with noninformation carrying bits. A better strategy is to output the best reproduction codeword possible from among those whose channel codewords will underflow the buffer or empty it exactly.

In Fig. 7 we show test sequence SQNR performance versus buffer length on speech data described later in this section, for several 256 codeword, 8-dimensional ECVQ systems with codeword entropies equal to 2.926 bits/vector. All are Huffman coded, buffered, and at-

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

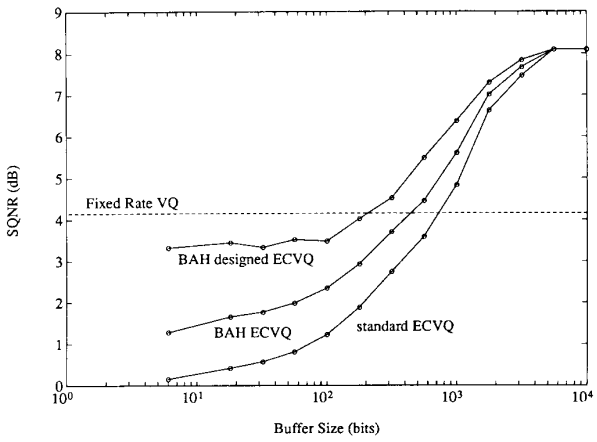


Fig. 7. Effect of buffer length on ECVQ with Huffman coding for a speech source. A fixed rate VQ and an "output zeros during overflow" standard strategy are compared to buffer adapted Huffman coding applied to an independently designed ECVQ (BAH) or included in the ECVQ design loop (BAH designed).

tached to a fixed rate, 3 bits/vector channel. At any buffer size, the ECVQ designed with the BAH overflow strategy inside the design loop outperforms the usual ECVQ system followed by either the standard overflow strategy or the BAH strategy. A fixed rate, 3 bits/vector VQ is shown for comparison; it is outperformed by all the variable rate systems at moderate to high buffer lengths.

We next compare, in a series of experiments, several quantizers in terms of their index entropy versus squared-error distortion. Table I shows several of the systems, giving a qualitative evaluation of their encoding complexity (source-to-index mapping) as well as noting m/n—a measure of the complexity of the required entropy coder for encoding the index. The scalar system considered is a simple uniform threshold quantizer with 17 thresholds and 19 cells, whose centroids serve as the reproduction levels, as in [17, Section VI, B]. The quantizer outputs are individually entropy encoded. The scalar system with block entropy coding is a scalar system with 4 thresholds and 5 cells, whose output indexes have been blocked into four dimensions before entropy encoding the 625 possible index blocks. This system is close in spirit to Ziv's universal quantizer [18], although it is not dithered. The A_2 lattice system employed has as its reproduction levels the innermost 37 points of a two-dimensional hexagonal lattice [32]. For high-resolution entropy-constrained two-dimensional quantization, the hexagonal lattice is optimal [7], [33]. The D_4 lattice system uses the 256 point Voronoi code enumerated by Conway and Sloane [9], [10]. The D_4 lattice, or the set of points in $\boldsymbol{R}^4$ with integral coordinates whose sum is even, is the best-known lattice for high-resolution entropy-constrained four-dimensional quantization [8]. The outputs of both the A_2 and D_4 systems are entropy coded as usual. Performance curves in the distortion-rate plane are traced out by scaling the various lattices. The ECVQ systems of four and eight dimensions were designed using the algorithms of Section III, with an initial codebook size of 256. All of these systems were designed on training sequences rather than on the underlying distributions.

TABLE I
COMPARISON OF ENTROPY-CODED QUANTIZATION SYSTEMS FOR THE EXPERIMENTS OF FIGS. 8, 9, AND 10

System	Block Size (n)	Number of Cells (m)	m/n	Encoding Complexity
Scalar	1	19	19	very simple
Scalar with Block Entropy Coding	4	625	156	simple
A_2 Lattice	2	37	19	moderate
D_4 Lattice	4	256	64	moderate
ECVQ	4	256	64	complex
ECVQ	8	256	32	complex

Results for the Gaussian iid source are shown in Fig. 8. We use training and test sequences of 40 960 samples each. The rate-distortion function for this source is [3, p. 99]

$$R(D) = \frac{1}{2}\log_2 \frac{\sigma^2}{D} \quad \text{bits/sample},$$

where σ^2 is the source variance and D is the average squared-error distortion. We note that, although ECVQ shows a slight improvement over the other systems, the gain is most likely not worth the additional complexity involved. Indeed, for this source our 8-dimensional ECVQ operating at 0.5 bits/sample with an SQNR of 2.3 dB is outperformed by several easily instrumentable codes of large blocklength such as the Golay code of length 24 evaluated by Adoul and Lamblin [34] as having an SQNR of 2.53 dB, and a length 32 Hadamard scrambling/permutation code due to Schroeder and Sloane [35] with an SQNR of 2.51 dB as measured by Adoul and Lamblin. Fischer's pyramid VQ represents a similar approach to the Laplacian iid source [36]. Observe, however, that in our results the D_4 lattice quantizer is uniformly outperformed by all the other systems. This suggests that we should use high-resolution quantizer conclusions about optimal lattices with care when the number of indexes is small. A similar case, in which an ECSQ outperforms a lattice quantizer based on the A_4^* lattice for the Laplacian memoryless source, was reported in Sayood *et al.* [11]. (The fact that the ECSQ and uniform threshold scalar quantizers have essentially identical performance for this source was demonstrated by Berger [14] and Farvardin and Modestino [17].)

Results for a Gauss–Markov source with correlation coefficient $a = 0.9$ are shown in Fig. 9. Again, the training and test sequences were 40 960 samples each. The rate-distortion function for this source for $R > 0.926$ bits/sample is [3, p. 113]

$$R(D) = \frac{1}{2}\log_2 \frac{0.19}{D} \quad \text{bits/sample}.$$

For $R < 0.926$ bits/sample, this is a lower bound on the rate-distortion curve (and hence the resulting SQNR curve

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

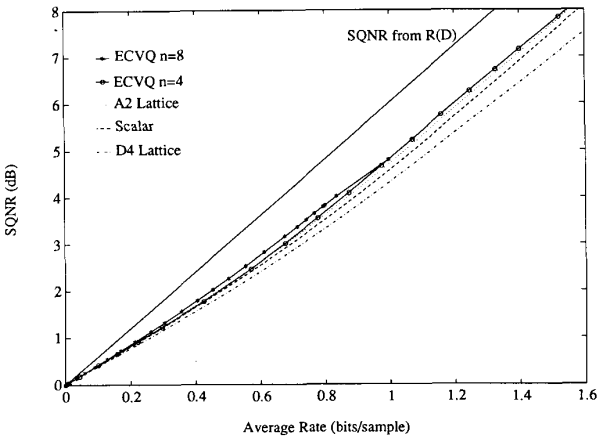


Fig. 8. Performance of some of the systems of Table I on a Gaussian iid source.

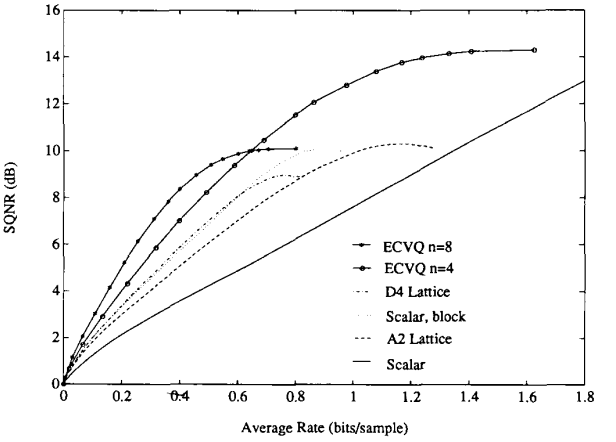


Fig. 10. Performance of the systems of Table I on a speech waveform source.

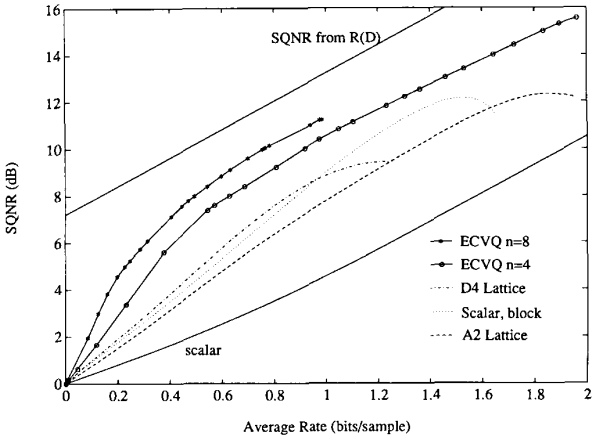


Fig. 9. Performance of the systems of Table I on a Gauss–Markov source with correlation coefficient $a = 0.9$. The $R(D)$ derived SQNR curve is an upper bound for $R < 0.926$ bits/sample.

is an upper bound on the true rate-distortion derived SQNR curve). In this experiment, there is a clear advantage for the ECVQ coders over the other systems; in many cases, rate can be cut in half with no increase in distortion. At 0.75 bits per sample for a blocklength of $n = 4$, ECVQ has a 1.6 dB advantage over the entropy-coded D_4 lattice—its nearest competitor. However, a predictive trellis code with SQNR as high as 12.58 dB at 1 bit/sample has been designed for this source [37]. Note, however, that the memory inherent in both the state and the predictor gives such a code an effective block length much larger than dimension 8.

In Fig. 10 we show results for a speech waveform coding experiment. The training sequence was 20 s of 8 kHz sampled speech (160 000 samples) and the test sequence was 10 s (80 000 samples) from the same (male) speaker. Again, the ECVQ systems have a clear advantage over the other systems. At 0.75 bits per sample for a blocklength of $n = 4$, ECVQ again has a 1.6 dB advantage over its nearest competitor—this time the block entropy-coded scalar system. Admittedly, direct vector quantization of waveform speech has had limited success in producing good subjective quality at such rates. More fruitful applications of ECVQ to real sources will probably come in the form of transform, subband, and image vector quantization. Preliminary results indicate that the technique cannot be used to full advantage on LPC-VQ, however.

Our second set of experiments compares ECVQ to full-search vector quantizers, complete tree-structured vector quantizers, and entropy-pruned tree-structured vector quantization, again for waveform speech coding. The full-search vector quantizers are designed by the generalized Lloyd algorithm [21] to have a minimum distortion subject to a constraint on the number of indexes. However, the indexes are subsequently entropy encoded. Similarly, the binary tree-structured vector quantizers are designed recursively using the generalized Lloyd algorithm, as in [38], but the output indexes are subsequently encoded. The entropy-pruned tree-structured vector quantizers are designed by a recent technique [22] that seeks to prune complete tree-structured quantizers so as to minimize the distortion subject to a constant on the index entropy.

The training sequence for these experiments was locally generated, consisting of 2 min of 8 kHz sampled speech from 3 males and 3 females. The test sequence was 40 s of speech from a male and female not in the training sequence. All vectors had 8 dimensions, so that the training and test sequences consisted of 120 000 and 40 000 vectors, respectively. Generalized Lloyd algorithm and complete tree-structured vector quantizers were designed for a 1.5 bit/sample rate (4096 codewords) and used as initial codebooks ($\lambda = 0$) for the ECVQ and entropy-pruned tree-structured vector quantizers, respectively. Additional generalized Lloyd algorithm and complete tree-structured vector quantizers were designed for lower rates. The results, which again demonstrate a performance advantage for ECVQ, are shown in Fig. 11. At 0.75 bits per sample, ECVQ has a 1.3 dB advantage in SQNR over entropy-pruned tree-structured vector quan-

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

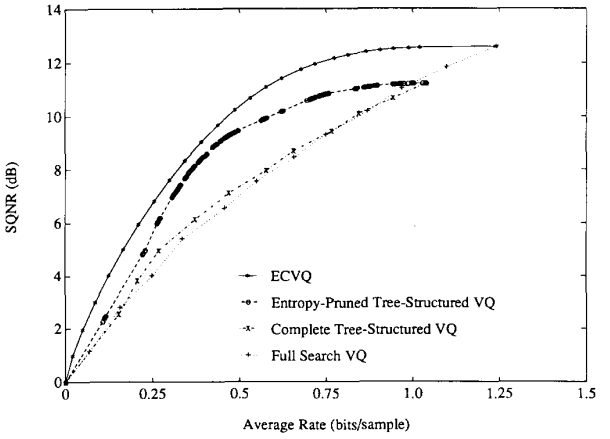


Fig. 11. Performance of entropy-coded vector quantizers in a speech waveform coding experiment.

tization, which in turn has a 1.6 dB advantage over other entropy-coded vector quantization systems. It is important, however, to note the advantage of the pruned tree-structured system in terms of complexity: like ECVQ, it realizes significant gains over the other vector quantizer systems, but it is much easier to search. For instance, the 0.75 bit/sample system requires an average of 6 distortion calculations per input vector compared to 4096 for ECVQ.

V. Applications to Pattern Recognition

The generalized Lloyd algorithm has more than a passing similarity to several clustering algorithms of the statistical pattern recognition community, in particular, an algorithm due to Forgey [39], the k-means algorithm [40], and the ISODATA algorithm [41]. The basic operation of all these algorithms is to assign data to cluster centers on the basis of which center is closest (generally in Euclidean or Mahalanobis distance), next choose new cluster centers by centroiding, and then iterate these two steps until convergence is reached. In the context of clustering, we will refer to these collectively as k-means-type algorithms. In this section, we suggest that the ECVQ algorithm may be useful in applications where it is desired to separate clusters of widely varying population, since it is a k-means-type algorithm that adjusts the distance to a cluster center by a measure of its population.

If the clusters are modeled by a Gaussian mixture, the optimal (minimum risk) classifier is straightforward to design, at least in principle, using Bayes' theorem. Let us assume that n-dimensional observed vectors $x_1, x_2, \cdots$, are drawn independently from a Gaussian mixture:

$$p(x) = \sum_{i=0}^{m-1} p(x|\omega_i)\, P(\omega_i),$$

where ω_i, $i = 0, \cdots, m-1$ represents the class, $P(\omega_i)$ is the prior probability of class i, and $p(x|\omega_i)$ is Gaussian with mean μ_i and covariance matrix Σ_i. The minimum probability of error rule for determining ω_i from an observation x follows directly from Bayes' theorem [42, ch. 2]:

$$\begin{aligned}\hat{i} &= \operatorname*{argmax}_i \left[p(\omega_i|x)\right] \\ &= \operatorname*{argmax}_i \left[\log p(x|\omega_i) + \log P(\omega_i)\right],\end{aligned}$$

which, under the assumption $\Sigma_i = \Sigma\ \forall i$, is equivalent to

$$\hat{i} = \operatorname*{argmin}_i \left[\rho_M(x, \mu_i) - \lambda \log P(\omega_i)\right], \qquad (7)$$

where $\rho_M(x, \mu_i) = (x - \mu_i)'\Sigma^{-1}(x - \mu_i)$ is the squared Mahalanobis distance, μ' denotes the transpose of μ, and $\lambda = 2$. We recognize (7) as identical to the calculation (3) performed by α, the first stage of the ECVQ encoder of Fig. 1, in mapping a vector to its index, providing we make the identification $\beta(i) = \mu_i$ and $|\gamma(i)| = -\log P(\omega_i)$. (This is precisely the choice of $|\gamma(i)|$ determined by the ECVQ design algorithm in (5) if $P(\omega_i)$ is assumed proportional to the population of cell i.) Hence, the ECVQ encoder first stage corresponds to an optimal classifier for a Gaussian mixture when the covariance matrices are the same for each class. If the prior probabilities are also the same for each class, i.e., the prior probabilities are uniform, then the standard vector quantizer encoder corresponds to the optimal classifier. For $\Sigma = \sigma^2 I$, we can further simplify (7) to write

$$\hat{i} = \operatorname*{argmin}_i \left[\rho(x, \mu_i) - \lambda^* \log P(\omega_i)\right], \qquad (8)$$

where $\rho(x, \mu_i)$ is squared-error distortion and λ^* is $2\sigma^2$.

The question still remains as to how to determine the parameters of the probability model from a set of training data $x_0, x_1, \cdots, x_{N-1}$. A typical approach is to determine the maximum likelihood estimate of the parameters. Necessary conditions for the estimate $\hat{p} = (\hat{p}_0, \cdots, \hat{p}_{m-1})$ of the prior probabilities and $\hat{\mu} = (\hat{\mu}_0, \cdots, \hat{\mu}_{m-1})$ of the means to be stationary points of the likelihood are [42, p. 193]

$$\hat{p}_i = \frac{1}{N} \sum_{k=0}^{N-1} \hat{p}(\omega_i|x_k, \hat{\mu}) \qquad (9)$$

$$\hat{\mu}_i = \frac{\sum_{k=0}^{N-1} \hat{p}(\omega_i|x_k, \hat{\mu})\, x_k}{\sum_{k=0}^{N-1} \hat{p}(\omega_i|x_k, \hat{\mu})} \qquad (10)$$

$$\hat{p}(\omega_i|x_k, \hat{\mu}) = \frac{p(x_k|\hat{\mu}_i)\,\hat{p}_i}{\sum_{j=0}^{m-1} p(x_k|\hat{\mu}_j)\,\hat{p}_j}, \qquad (11)$$

where $p(x|\hat{\mu}_i)$ is the multivariate Gaussian density with mean $\hat{\mu}_i$ and covariance matrix Σ_i.

Equations (9), (10), and (11) form the basis for an iterative algorithm which attempts to find the maximum likelihood estimates of the means and prior probabilities by first fixing $\hat{p}$ and $\hat{\mu}$ and determining the $\hat{p}(\omega_i|x_k, \hat{\mu})$ from (11), then fixing $\hat{p}(\omega_i|x_k, \hat{\mu})$ and determining $\hat{p}$ and

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

$\hat{\mu}$ from (9) and (10). This algorithm, which we will term the maximum likelihood algorithm, was first proposed by Wolfe [43]. Duda and Hart approximate (11) by

$$\hat{p}(\omega_i|x_k, \hat{\mu}) = \begin{cases} 1 & \text{if } \rho_M(x_k, \hat{\mu}_i) < \rho_M(x_k, \hat{\mu}_j), \quad j \neq i, \\ 0 & \text{otherwise.} \end{cases}$$

This is indeed reasonable if the μ_i are well separated and the $\hat{p}_i$ are all equal. In this case, the resulting iterative algorithm is equivalent to the k-means-type algorithms. However, if we allow the $\hat{p}_i$ to vary, then a more reasonable approximation is

$$\hat{p}(\omega_i|x_k, \hat{\mu}) = \begin{cases} 1 & \text{if } \rho_M(x_k, \hat{\mu}_i) - \lambda \log \hat{p}_i \\ & \quad < \rho_M(x_k, \hat{\mu}_j) - \lambda \log \hat{p}_j, \quad j \neq i, \\ 0 & \text{otherwise.} \end{cases}$$

When applied to (9), (10), and (11), this leads to the ECVQ algorithm. Hence, the ECVQ algorithm is an approximation to the maximum likelihood algorithm for estimating means *and* prior probabilities in the same way that the k-means-type algorithms are an approximation to the maximum likelihood algorithm for estimating means with the classes assumed equiprobable.

To compare the behavior of several of these algorithms, we simulated a 40 000 sample training sequence drawn from a four component bivariate Gaussian mixture with different prior probabilities but all covariance matrices equal to the identity matrix. Several thousand samples from the density are shown in Fig. 12. We used the maximum likelihood algorithm, the generalized Lloyd algorithm, and the ECVQ algorithm (with $\lambda = 2$) to estimate means and prior probabilities. (The ECVQ algorithm and maximum likelihood algorithm used the final estimate of the generalized Lloyd algorithm as the starting point for their iterations.) We then used the Bayes' classifier corresponding to each algorithm to classify data for which we (but not the algorithms) knew the labels. The results of the experiment are shown in Tables II and III. Not surprisingly, the maximum likelihood algorithm provided the best estimates of the prior probabilities and the best classifier performance. It also provided the best estimates of the means (not shown). While not achieving the performance of the maximum likelihood algorithm, the ECVQ algorithm both provided better mean estimates and significantly outperformed the generalized Lloyd algorithm in classification accuracy. Although not shown, we were able to improve the ECVQ algorithm's performance by varying λ from the "optimal" value of 2. In fact, the choice $\lambda = 1$ resulted in a slightly lower misclassification rate than the maximum likelihood algorithm. Coupled with the fact that λ in general depends on knowledge of the cluster variances [see (8)], which may not be readily available, this suggests that a classifier design might consider several different λ's. The curve tracing algorithm of Section III would be useful for selecting such a set. However, dependence of the ECVQ algorithm on the starting codebook, training sequence size, and choice of λ appears to be much more serious in pattern recognition applications than in source coding applications, so that the algorithm should be applied with care, particularly in circumstances where an appeal to maximum likelihood becomes even more difficult.

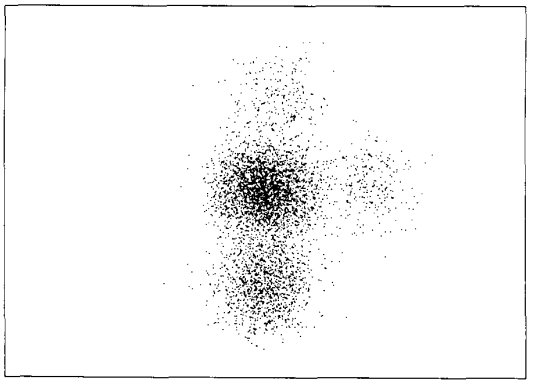


Fig. 12. Four component bivariate Gaussian mixture. The component probabilities are nonuniform.

TABLE II
TRUE AND ESTIMATED CLASS PROBABILITIES FOR A GAUSSIAN MIXTURE

Class	True Probabilities	Maximum Likelihood Estimated Probabilities	ECVQ Estimated Probabilities
1	0.700	0.699	0.725
2	0.050	0.052	0.041
3	0.050	0.050	0.045
4	0.200	0.199	0.189

TABLE III
PERFORMANCE OF VARIOUS ALGORITHMS FOR PARAMETER ESTIMATION AND CLASSIFICATION OF A GAUSSIAN MIXTURE

Algorithm	Maximum Likelihood	Generalized Lloyd Algorithm	ECVQ
Log Likelihood	−72459	−79504	−72633
Misclassification Rate	0.0317	0.1640	0.0356

VI. DISCUSSION AND CONCLUSION

In this paper we have introduced an iterative descent algorithm for designing locally optimal variable rate coders for use in a block quantization or data compression scheme. In essence, the algorithm designs vector quantizers which are optimized to perform well when followed by a variable rate entropy coder, such as a Huffman, arithmetic, or Ziv–Lempel coder. The choice of which entropy coder to use is not made by the algorithm, but is instead left to the implementor. Any of these entropy coders will

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

produce an average rate approximately equal to the codeword entropy. The algorithm simply ensures that the codeword entropy and the distortion are low.

For high rates (high resolution), asymptotic quantization theory predicts that, for the squared error distortion measure and other difference distortion measures expressible as the rth power of a seminorm, the advantage in performance of the optimum ECVQ system over simpler scalar and lattice systems will be small. Our experimental investigations have focused on the low-rate region (on the order of one bit per sample), where it is shown that for memoryless sources, such as the Gaussian iid source, the performance improvement of ECVQ over simple scalar quantization is indeed negligible. However, for sources with memory, such as the first-order Gauss–Markov source with correlation coefficient $a = 0.9$, as well as for real speech, the improvement in mean squared error is shown to be as much as 1.6 dB at 0.75 bits/sample. Gains in the performance of ECVQ over other forms of entropy-coded vector quantization are shown to be even more substantial.

A real-time ECVQ implementation would most likely be no more complex than an entropy-coded VQ implementation. The principal VQ encoder calculation for many interesting distortion measures can be simplified to an inner product. For such distortion measures, the first stage calculation for the ECVQ encoder also has an inner product form (with $-\lambda \log_2 p(i)$ added to the constant); hence, several new inner product based systolic array vector quantizer architectures are also attractive for implementing the first stage of the ECVQ encoder, allowing large codebooks to be searched in real time [44], [45]. The remaining stages of the ECVQ can be implemented by lookup tables. Thus, ECVQ is an attractive option for a number of real-time low rate data compression systems.

The ECVQ algorithm also plays a role in statistical pattern recognition, where it generalizes the "k-means" algorithm and its variants used in clustering, in that the mixture is not assumed to be equiprobable. That is, the ECVQ algorithm estimates both the means *and the prior probabilities* of the unknown mixture. It is shown experimentally that in the case of widely different priors, the ECVQ algorithm outperforms the equivalent k-means-type of algorithm in likelihood as well as in probability of error. In addition, the estimated means and prior probabilities are close to their true values. It is conjectured that the ECVQ algorithm may be useful even in exploratory data analysis if it is suspected that the class probabilities may not be uniform.

References

[1] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE Nat. Conv. Rec.*, 1959, pp. 142–163.

[2] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

[3] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[4] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 676–683, Sept. 1968.

[5] T. J. Goblick and J. L. Holsinger, "Analog source digitization: A comparison of theory and practice," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 323–326, Apr. 1967.

[6] P. Zador, "Asymptotic quantization error of continuous signals and their quantization dimensions," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 139–149, Mar. 1982. (Previously an unpublished Bell Lab memo, 1966.)

[7] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 373–380, July 1979.

[8] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 211–226, Mar. 1982.

[9] ——, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227–231, Mar. 1982.

[10] ——, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 820–824, Nov. 1983.

[11] K. Sayood, J. D. Gibson, and M. C. Rost, "An algorithm for uniform vector quantizer design," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 805–814, Nov. 1984.

[12] R. C. Wood, "On optimum quantization," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 248–252, Mar. 1969.

[13] T. Berger, "Optimum quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 759–765, Nov. 1972.

[14] ——, "Minimum entropy quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 149–157, Mar. 1982.

[15] A. N. Netravali and R. Saigal, "Optimum quantizer design using a fixed-point algorithm," *Bell Syst. Tech. J.*, vol. 55, pp. 1423–1435, Nov. 1976.

[16] P. Noll and R. Zelinski, "Bounds on quantizer performance in the low bit-rate region," *IEEE Trans. Commun.*, vol. COM-26, pp. 300–304, Feb. 1978.

[17] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 485–497, May 1984.

[18] J. Ziv, "On universal quantization," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 344–347, May 1985.

[19] M. Gutman, "On uniform quantization with various distortion measures," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 169–171, Jan. 1987.

[20] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–136, Mar. 1982. (Previously an unpublished Bell Lab Tech. Note, 1957.)

[21] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[22] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree structured source coding and modeling," *IEEE Trans. Inform. Theory*, to appear.

[23] M. B. Pursley and L. D. Davisson, "Variable rate coding for nonergodic sources and classes of ergodic sources subject to a fidelity constraint," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 324–337, May 1976.

[24] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 460–473, July 1972.

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

[26] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Pro. IRE*, vol. 40, pp. 1098–1101, Sept. 1952.

[27] R. G. Gallager, "Variations on a theme by Huffman," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 668–674, Nov. 1978.

[28] J. Rissanen and G. G. Langdon, Jr., "Arithmetic coding," *IBM J. Res. Develop.*, vol. 23, pp. 149–162, Mar. 1979.

[29] F. Rubin, "Arithmetic stream coding using fixed precision registers," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 672–675, Nov. 1979.

[30] J. Ziv, "Coding theorems for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 405–412, July 1978.

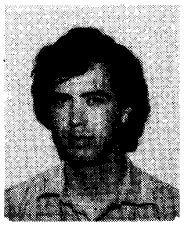
[31] T. Berger, M. U. Chang, and S. Y. Tung-Kleinberg, "Quantization-permutation codes and buffer-adapted Huffman codes," in *Proc. 18th Annu. Allerton Conf. Commun., Contr.*, Comput., Monticello, IL, 1980, pp. 443–436.

[32] K. D. Rines and N. C. Gallagher, Jr., "The design of two-dimensional quantizers using pre-quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 232–238, Mar. 1982.

[33] D. J. Newman, "The hexagon theorem," *IEEE Trans. Inform. The-*

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.

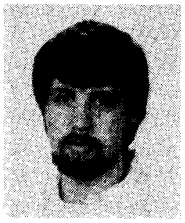
ory, vol. IT-28, pp. 137–139, Mar. 1982. (First appeared as a Bell Lab Tech. Memo, May 1964.)
[34] J. Adoul and C. Lamblin, "A comparison of some algebraic structures for CELP coding of speech," in *Proc. ICAASP*, Dallas, TX, Apr. 1987, pp. 45.8.1–45.8.4.
[35] M. R. Schroeder and N. J. A. Sloane, "New permutation codes using Hadamard unscrambling," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 144–146, Jan. 1987.
[36] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 568–583, July 1986.
[37] E. Ayanoğlu and R. M. Gray, "The design of predictive trellis waveform coders using the generalized Lloyd algorithm," *IEEE Trans. Commun.*, vol. COM-34, pp. 1073–1080, Nov. 1986.
[38] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 562–574, Oct. 1980.
[39] E. W. Forgey, "Cluster analysis of multivariate data: Efficiency versus interpretability of classifications," *Biometrics*, vol. 21, no. 3, p. 768, 1965.
[40] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probabil.* Berkeley, CA: University of California Press, 1967, pp. 281–297.
[41] G. H. Ball and D. J. Hall, "ISODATA, A novel method of data analysis and pattern classification," Tech. Rep. AD 699616, Stanford Res. Inst., Menlo Park, CA, 1965.
[42] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[43] J. H. Wolfe, "Pattern clustering by multivariate mixture analysis," *Multivariate Behavior. Res.*, vol. 5, pp. 329–350, 1970.
[44] P. Cappello, G. Davidson, A. Gersho, C. Koc, and V. Somayazulu, "A systolic vector quantization processor for real-time speech coding," in *Proc. ICASSP*, 1986, pp. 2143–2146.
[45] R. Dianysian and R. L. Baker, "A VLSI chip set for real time vector quantization of image sequences," in *Proc. ISCS*, May 1987.



Philip A. Chou was born in Stamford, CT, on April 17, 1958. He received the B.S.E. degree from Princeton University, Princeton, NJ, in 1980 and the M.S. degree from the University of California at Berkeley in 1983, both in electrical engineering and computer science. In 1988 he received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

From 1977 through the present he has worked for IBM, Bell Laboratories, Princeton Plasma Physics Lab, Telesensory Systems, Speech Plus, and Hughes Aircraft Corporation, where he was involved variously in office automation, motion estimation in television, optical character recognition, LPC speech compression and synthesis, text-to-speech synthesis by rule, and the compression of digitized terrain. His research interests are pattern recognition, data compression, and speech and image processing.

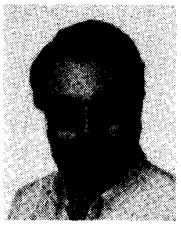
Dr. Chou received the National Merit Scholarship in 1976, the Regents Fellowship in 1981, the Lockheed Leadership Fellowship in 1983, and the Hughes Fellowship in 1984. He is a member of Phi Beta Kappa, Tau Beta Pi, Sigma Xi, and the IEEE Computer, Information Theory, and Acoustics, Speech, and Signal Processing Societies.



Tom Lookabaugh was born in Billings, MT, on July 21, 1961. He received the B.S. degree in engineering physics from Colorado School of Mines in 1983, the M.S. degrees in electrical engineering (1984), engineering management (1986), and statistics (1987), and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1988, where he was an NSF graduate Fellow.

In June 1988 he joined Compression Labs, Inc. His areas of interest include quantization and data compression, speech and image processing, and efficient architectures for signal processing.

Dr. Lookabaugh is a member of Sigma Pi Sigma and Tau Beta Pi.



Robert M. Gray (S'68–M'69–SM'77–F'80) was born in San Diego, CA, on November 1, 1943. He received the B.S. and M.S. degrees from M.I.T., Cambridge, MA, in 1966 and the Ph.D. degree from the University of Southern California, Los Angeles, in 1969, all in electrical engineering.

Since 1969 he has been with Stanford University, Stanford, CA, where he is currently a Professor of Electrical Engineering. His research interests are the theory and design of data compression and classification systems, oversampled analog-to-digital conversion, speech and image coding and recognition, and ergodic and information theory. He is the coauthor, with L. D. Davisson, of *Random Processes* (Englewood Cliffs, NJ: Prentice-Hall, 1986), and the author of *Probability, Random Processes, and Ergodic Properties* (New York: Springer-Verlag, 1988).

Dr. Gray was a member of the Board of Governors of the IEEE Information Theory Group (1974–1980, 1985–1988). He was an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY from September 1977 through October 1980, and Editor of that journal from October 1980 through September 1983. He is currently an Associate Editor of *Mathematics of Control, Signals, and Systems*. He has been on the program committee of several IEEE International Symposia on Information Theory and was an IEEE Delegate to the Joint IEEE/USSR Workshop on Information Theory in Moscow in 1975. He was co-recipient with L. D. Davisson of the 1976 IEEE Information Theory Group Paper Award, and co-recipient with A. Buzo, A. H. Gray, Jr., and J. D. Markel of the 1983 IEEE ASSP Senior Award. He was a Fellow of the Japan Society for the Promotion of Science (1981) and the John Simon Guggenheim Memorial Foundation (1981–1982). In 1984 he was awarded an IEEE Centennial medal. He is a member of Sigma Xi, Eta Kappa Nu, SIAM, IMS, AAAS, AMS, and the Societé des Ingenieurs et Scientifiques de France. He holds an Advanced Class Amateur Radio License (KB6XQ).

Authorized licensed use limited to: China University of Petroleum. Downloaded on August 21,2024 at 06:47:19 UTC from IEEE Xplore. Restrictions apply.