Integer Programming Approach to Fixed-rate Entropy-coded Quantization

Sasan Nikneshan, Erik Hons, and Amir K. Khandani
Dept. of Elec. and Comp. Engineering
University of Waterloo
Waterloo, Ontario N2L 3G1
Tel. (519) 885-1211 x5324, Fax. (519) 746-3077

Abstract— **This paper describes two new fixed-rate entropy-coded quantization methods for stationary memoryless sources where the structure of code-words are derived from a variable-length scalar quantizer. In the first method, we formulate the quantization as a zero-one integer optimization problem. We show that the resulting integer program can be closely approximated by solving a simple linear program. The result is a Lagrangian formulation which adjoin the constraint (length) to total distortion. Unlike the previous methods with a fixed Lagrangian multiplier (fixed-slope, and variable rate output), we use an iterative algorithm to optimize Lagrangian function while updating the slope of the function until the cost constraint is satisfied with equality (ensure to be fixed-rate). In order to achieve some part of packing gain, we combine the process of trellis encoding with that of quantizer shaping using linear programming. This results in an iterative use of Viterbi algorithm for optimizing the Lagrangian function. For the important class of sources with a monotonically decreasing density, we present another fixed-rate method with negligible complexity. Numerical results show an excellent performance with a small complexity for the proposed schemes as compared to previously known methods.**

I. Introduction

Optimum fixed-rate scalar quantizers, introduced by Max [1] and Lloyd [2] (LMQ), minimize the average distortion for a given number of reconstruction levels. LMQ performs worse than the optimal Entropy Constrained Scalar Quantizer (ECSQ) in the absence of channel noise. ECSQ is known to perform close to the rate-distortion bound for a large class of memoryless sources [3], [4]. Gish and Pierce showed that the optimal ECSQ has uniformly spaced reconstruction levels regardless of the source probability density function [4].

The design of an entropy-constrained vector quantizer is generally based on the minimization of the functional

$$J = D + \lambda H$$

where D is the distortion between input and output, λ is the Lagrange multiplier, and H is the entropy of the output. The problem of convex optimization in information theory was first presented by Blahut [5]. The Blahut algorithm for finding the rate-distortion function is based on minimizing a Lagrangian (adjoining the distortion and the mutual information) where the Lagrangian multiplier is interpreted as the slope of the hyper-plane supporting the convex achievable region. Chou and *et al.,* [6] presented an algorithm for the entropy constrained vector quantization (ECVQ). Their implementation is similar to generalized Lloyd algorithm [7]. The generalized Lloyd algorithm [7] is a time consuming approach. In order to alleviate this problem, Equitz [8] proposed a recursive algorithm, called pairwise nearest neighbor (PNN). Recently, an entropy-constrained version of the PNN design algorithm was proposed by Garrido, Pearlman, and Finamore [9] and is called entropy-constrained pairwise nearest neighbor (ECPNN).

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Communications and Information Technology Ontario (CITO).

All of these methods have a variable-rate output with its concomitant difficulties. To take advantage of entropy coding, while avoiding the dis-advantages associated with conventional methods based on using variable rate codes (including error propagation and buffering problems), one can use *fixed-rate* entropy-coded vector quantization (FEVQ).

The pyramid vector quantizer (PVQ), introduced by Fischer (for Laplacian sources) [10], is a fixed-rate VQ in which the code-vectors are located on the intersection of a cubic lattice and a pyramid in N-dimensional space. For Laplacian sources this quantizer is asymptotically optimal and achieves the performance of ECSQ, but for other sources it does not approach the ECSQ performance, even for large N. Hung and *et al.,* investigated the application of PVQ for compressed image transmission over noisy channel, where the fixed-rate quantization reduces the susceptibility to bit error corruption [11]. They also proposed a new method of deriving the indices of the lattice points of the multi-dimensional pyramid and described how theses techniques could also improve the channel noise immunity of general symmetric lattice quantizer.

One class of schemes are based on selecting the N-fold symbols with the lowest additive self-information (typical set). This scalar-vector quantizer (denoted by SVQ) is a fixed-rate vector quantizer derived from a variable-rate scalar quantizer. A method for exploiting this structure based on using a dynamic programming approach with the states corresponding to the length of the code-words is used by Laroia and Farvardin in [12]. The core idea in the schemes of [12] is to use a state diagram with the transitions corresponding to the one-D symbols. This results in a trellis composed of N stages where N is the space dimensionality. Then, the Viterbi algorithm is used to find the path of the minimum overall additive distortion through the trellis. Reference [13] uses a different approach to dynamic programming showing improvement with respect to the schemes of [12]. The key point in [13] is to decompose the underlying operations into the lower dimensional subspaces. This decomposition avoids the exponential growth of the complexity.

Trellis coded quantization (TCQ), introduced by Marcellin and Fischer [14], is based on applying the Ungerboeck notion of set partitioning to the partitions of a scalar quantizer where a trellis structure is used to prune the expanded number of quantization levels down to the desired encoding rate.

Entropy-constrained TCQ [15,16] is based on using a distance measure which is a linear combination of the codeword length and quantization distortion. This is a generalization of the entropy coded quantization method [6] to include TCQ. Laroia and Farvardin combine the SVQ idea [12] with TCQ and proposed a fixed-rate quantizer which they call Trellis-based Scalar Vector Quantizer (TB-SVQ) [17]. The resulting quantizer shows an excellent performance assuming error free transmission.

In this paper, we propose an integer programming approach to fixed-rate entropy-coded vector quantization (FEVQ) for stationary memoryless sources. We use a zero-one integer optimization formulation for quantization problem, which was introduced in [18]. In order to solve the resulting zero-one integer program, we approximated it to a simple linear program. The result is a Lagrangian formulation adjoining the distortion and length of codewords. In order to achieve some packing gain, we combine the trellis encoding [14] and the new proposed FEVQ. For the important special case of a source with a monotonically decreasing density, we present a second method with negligible complexity,

The rest of article is organized as follows: Section II contains a brief description of the linear program formulation and the approach to solve this problem. Starting from two initial points, we derive an equation to find a chain of improving solution. The derivation is discussed in detail. We show the resulting equation is in the form of a Lagrangian function. In the end of the section, the idea of fixed-rate entropy-constrained trellis-coded quantization using linear programming is introduced. In the Section III, we talk about a simple approach to fixed-rate quantization of a special class of source with monotonically decreasing source density. Finally, in Section IV, we conclude the paper by presenting the numerical results and a comparison between proposed methods with some other quantization schemes.

II. Formulation of Encoding as a Linear Program

Consider an N dimensional vector quantizer derived from N variable length scalar quantizers. Each scalar quantizer consists of M partitions with reconstruction levels $(q_1, q_2, ..., q_M)$, where $q_1 < q_2 < ... < q_M$, having self information of $\vec{c} = \{c(1), c(2), ..., c(M)\}$. There is a variable-length, binary, prefix code associated with each quantizer, where codeword corresponding to level q_j has a length of $\ell(j)$.

To formulate the decoding problem as an integer program, the jth partition of the scalar quantizer along the ith dimension is identified by a binary variable $\delta_i(j)$, $i = 1, ..., N$, $j = 1, ..., M$ where $\delta_i(j) = 0, 1$ and $\sum_j \delta_i(j) = 1$, $i = 1, ..., N$. To select the element indexed by j_0 along the ith dimension, we set $\delta_i(j_0) = 1$ and $\delta_i(j) = 0$, $j \neq j_0$. This results in,

$$\begin{aligned} &\text{Minimize} && \sum_{i=1}^{N}\sum_{j=1}^{M} \delta_i(j) d_i(j) \\ &\text{Subject to:} && \sum_{i=1}^{N}\sum_{j=1}^{M} \delta_i(j) c(j) + s_c = C_{max}, \quad s_c \geq 0, \\ & && \textstyle\sum_{j=1}^{M} \delta_i(j) = 1, \quad \forall i, \qquad \delta_i(j) = 0, 1, \quad \forall i, j, \end{aligned} \tag{1}$$

where s_c is the slack variable of the cost constraint. Each of the equalities $\sum_j \delta_i(j) = 1$, $i = 1, ..., N$, is called an *indicator constraint.*

The immediate problem in solving (1) is that the variables $\delta_i(j)$ are restricted to be integer numbers (0 or 1). We show that one can relax the zero-one constraint and then round the result and this may only cause a small degradation in the quality of the solution. It [18] a solution based on generalized upper bounding of linear programming is suggested to solve Eq. 1. In the current article, we use an improved solution method with a much lower complexity. In addition, the proposed solution has an interpretation in terms of the conventional Lagrangian method (where the corresponding Lagrange multiplier is iteratively optimized). This provides a natural framework to combine the method with a trellis structure to achieve some extra packing gain.

To solve the resulting linear program, we find a chain of improving solutions each expressed in terms of a linear interpolation between two intermediate solutions. The interpolation coefficients are computed such that the cost constraint is satisfied with equality. At each iteration, one of the two points involved in the interpolation is updated in a way that the resulting improvement in the objective function is maximized. The updating is achieved by solving an LP problem which is solely subject to the indicator constraints and has a trivial complexity.

Consider the solution $\mathbf{x} = \{x_i(j), i = 1, \cdots, N, j = 1, \cdots, M\}$ and assume that the objective function value and the cost associated with $\mathbf{x}$ are equal to,

$$\begin{aligned} D &= \sum_{i=1}^{N}\sum_{j=1}^{M} x_i(j) d_i(j) \\ C &= \sum_{i=1}^{N}\sum_{j=1}^{M} x_i(j) c(j), \end{aligned} \tag{2}$$

respectively. We look at $\mathbf{x}$ as providing a tradeoff between D and C. The main idea is to find a linear interpolation between an appropriate set of such $\mathbf{x}$'s that: (i) the cost constraint is satisfied with equality, and (ii) the overall tradeoff is optimized.

Consider a solution obtained by interpolating between two points, say $\mathbf{x}^1$, $\mathbf{x}^2$, and assume that we try to improve the solution by bringing a third point, say $\mathbf{x}^3$, into the interpolation procedure. The following LP problem is used to compute the interpolation coefficients:

$$\begin{array}{ll} \text{Minimize:} & \lambda_1 D_1 + \lambda_2 D_2 + \lambda_3 D_3 \\ \text{Subject to:} & \lambda_1 C_1 + \lambda_2 C_2 + \lambda_3 C_3 = C_{Max} \\ & \lambda_1 + \lambda_2 + \lambda_3 = 1 \end{array} \tag{3}$$

where D_1, D_2, D_3 and C_1, C_2, C_3 are the objective function and cost associated with points $\mathbf{x}^1$, $\mathbf{x}^2$, $\mathbf{x}^3$ respectively. As the LP problem in (3) has two constraints, only two of the corresponding λ's will be non-zero. This simply means that it suffices to use only two points for the interpolation. This will be the case even if we try to provide an interpolation among a larger number of such points.

For a given value of λ_3, the LP problem in (3) is equivalent to:

$$\begin{array}{ll} \text{Minimize:} & \lambda_1 D_1 + \lambda_2 D_2 + \lambda_3 D_3 \\ \text{Subject to:} & \lambda_1 C_1 + \lambda_2 C_2 = C_{Max} - \lambda_3 C_3 \\ & \lambda_1 + \lambda_2 = 1 - \lambda_3 \end{array} \tag{4}$$

Solving for λ_1, λ_2, we obtain,

$$\lambda_1 = \frac{-C_{Max} + \lambda_3 C_3 + (1-\lambda_3) C_2}{C_2 - C_1} \quad \text{and} \tag{5}$$

$$\lambda_2 = \frac{C_{Max} - \lambda_3 C_3 - (1-\lambda_3) C_1}{C_2 - C_1}. \tag{6}$$

Substituting in (3), results in,

$$\lambda_1 D_1 + \lambda_2 D_2 + \lambda_3 D_3 = \lambda_3 (D_3 + \pi_1 C_3 + \pi_2) + \frac{(C_2 - C_{Max}) D_1 + (C_{Max} - C_1) D_2}{C_2 - C_1}$$

where,

$$\pi_1 = \frac{D_1 - D_2}{C_2 - C_1} \quad \text{and} \quad \pi_2 = \frac{C_1 D_2 - C_2 D_1}{C_2 - C_1} \tag{7}$$

The term $[(C_2 - C_{Max})D_1 + (C_{Max} - C_1)D_2]/(C_2 - C_1)$ in (7) is the best value of the objective function obtained by interpolating between only $\mathbf{x}^1$ and $\mathbf{x}^2$. The point $\mathbf{x}^3$ is selected to minimize the effect of the related term in (7). This results in the following LP problem for the selection of $\mathbf{x}^3$:

$$\begin{array}{ll} \text{Minimize:} & D_3 + \pi_1 C_3 = \sum_{i=1}^{N} \sum_{j=1}^{M} [d_i(j) + \pi_1 c(j)]\, x_i^3(j) \\ \text{Subject to:} & \text{Indicator constraints} \end{array} \tag{8}$$

where $x_i^3(j)$ are the component of $\mathbf{x}^3$. Note that the LP in (8) is decomposable, and consequently, has a trivial complexity.

If the minimum value of (8) satisfies $(D_3 + \pi_1 C_3)_{\min} + \pi_2 < 0$, it means that the inclusion of $\mathbf{x}^3$ results in a decrease in the objective function value in which case the iteration will continue. After this, the whole procedure is repeated for the resulting two points until it merges to a stationary condition which considering the linearity of the function will be the global optimum solution of the linear program.

In summary, given the points $\mathbf{x}^1$, $\mathbf{x}^2$, the algorithm computes the value of π_1 using (2), (7) and then finds the optimum solution of (8). Then, one of the two points $\mathbf{x}^1$ or $\mathbf{x}^2$ is updated and the procedure is repeated until no change in the value of π_1 in two subsequent iterations is observed.

A. Linear Programming Approach to Fixed-rate Entropy-coded Trellis Coded Quantization

Consider an N-dimensional TCQ (N as a block length) with $\nu = 2^\mu$ states. The corresponding scalar quantizer is specified by an alphabet (set of quantization level) $Q = \{q_1, q_2, \cdots, q_{2M}\}$ where $M = 2^m$ [14]. The quantizer points is partitioned to 4 subsets, S_l, $l = 1, 2, \cdots, 4$, where each subset consists of $M/2$ codewords. Given a data sequence $\mathbf{x}$, the Viterbi algorithm is used to find the allowable sequence of output symbols $\hat{\mathbf{x}}$. The sequence of output symbols chosen by the Viterbi algorithm can be represented by the bit sequence specifying the corresponding trellis path (sequence of subsets) in addition to the sequence of $m-1$ bit codewords to specify symbol from chosen subsets (plus μ bits specifying the starting state).

The entropy-constrained trellis coded quantizer (ECTCQ) is a trellis coded quantizer derived from a variable length scalar quantizer. Assume that there is a variable-length, binary, prefix code associated to each subset[1]. Therefore, each codeword is written as $r_{k,l} \in S_k$ with k and l denoting the lth codeword in the kth subset. Corresponding to each $r_{k,l}$, there is binary string $c_{k,l} \in \mathcal{C}$ with length $l_{k,l}$ (in bits). The codewords with the same first subscript must be uniquely decodable (i.e., the corresponding reproduction levels of these index codewords are in the same subset). The entropy-constrained trellis-coded quantizer uses a Lagrange multiplier to adjoin the distortion measure to the constraint, and its objective is to minimize the that functional while satisfying the trellis constraint. To minimize the functional (J_λ) the encoder uses Viterbi algorithm with trellis branches labeled with appropriate subsets and the branch metric (the biased squared distortion $\rho(x, r_{k,l}) = (x - r_{k,l})^2 + \lambda l_{k,l}$). The branch metric is sum of two terms, one term is the squared error between the current input and its closest codeword in the subset associated with that branch, and the other is λ times the length of the index codeword corresponding to that closest reproduction codeword.

For a fixed-rate entropy-constrained trellis-coded quantizer, we have to impose a constraint on the total length of codeword. Eq. 1 shows the formulation of fixed-rate entropy-constrained quantizer suitable for applying the linear programming approach. The formulation of the fixed-rate entropy constrained TCQ is derived by adding the extra constraint regarding the trellis structure to Eq. 1. In order to have the same rate as Eq. 1 we should take into the account that μ bits will be used to specify the starting

[1] These variable-length codes can also be designed for $S_1 \cup S_3$ and $S_2 \cup S_4$ [16].

state . Therefore Eq. 1 will be modified as follow;

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}\delta_i(j)d_i(j) \\ \text{Subject to:} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}\delta_i(j)\cdot\ell(j) \leq L_{max} - \mu \\ & \text{Trellis constraint \& Indicator constraint} \end{aligned} \tag{9}$$

We still find a chain of improving solutions each expressed in terms of a linear interpolation between two intermediate solutions. Starting with two points, say $\mathbf{x}^1$, and $\mathbf{x}^2$, we try to improve the solution by bringing a third point, say $\mathbf{x}^3$, into the interpolation procedure. By following the same procedure, from Eq. 1 to Eq. 7, we reach to a similar equation as Eq. 8. The updating is achieved by solving a new LP problem which is subject to the trellis and indicator constraints. This results in the following LP problem for the selection of $\mathbf{x^3}$:

$$\begin{aligned} \text{Minimize:} \quad & D_3 + \pi_1 C_3 = \sum_{i=1}^{N}\sum_{j=1}^{M}\left[d_i(j) + \pi_1 c(j)\right] x_i^3(j) \\ \text{Subject to:} \quad & \text{Indicator and trellis constraints} \end{aligned} \tag{10}$$

Noting that the indicator constraint are decomposable, the encoder uses the Viterbi algorithm to find the solution of Eq. 10. In this case, the branch metric is the sum of two terms, one term is the squared distance between the input and the closest codeword to the input in the subset associated with that branch, and the other is π_1 times the length of codeword $(d_i(j) + \pi_1 c(j))$. Therefore, given two points $\mathbf{x}^1$, $\mathbf{x}^2$, the algorithm computes the value of π_1, and then using Viterbi algorithm finds the solution of Eq. 10. Then, one of the points $\mathbf{x}^1$ or $\mathbf{x}^2$ is updated. We continue updating until no further change in value of π_1 is shown.

III. Problem Formulation of decoding as an integer program

In the following, we assume that if the length of the Huffman codes corresponding to different quantizer partitions are ordered, then two subsequent values differ in at most one unit. This means that the structure of the Huffman code is decided for the quantizer. Following our earlier formulation, for a given input vector $\mathbf{a} = (a_1, \cdots, a_N)$, we define a discrete function $R(l_j, a_j)$ (l_j is a discrete variable and a_j is a continuous one), which maps the jth component of $\mathbf{a}$, i.e., a_j to the closest reconstruction level with length of l_j. The distortion associated with this mapping is $(a_j - R(l_j, a_j))^2$. We define a new function for the jth coordinate called $\phi(a_j, l_j) = -(a_j - R(l_j, a_j))^2$. Using above definitions, we can have the following formulation for a fixed-rate entropy-coded quantization:

$$\max\{\Phi(\mathbf{a},\mathbf{l}) : \mathbf{l} \in S, L(\mathbf{l}) \leq L_{Max}\} \tag{11}$$

where S is the set of N-tuples of nonnegative integers of allowed codeword lengths,

$$L(\mathbf{l}) = \sum_{j=1}^{N} l_j$$

$L_{Max} > 0$, and

$$\Phi(\mathbf{a},\mathbf{l}) = \sum_{j=1}^{N} \phi(a_j, l_j)$$

The following algorithm can be used to solve this optimization problem. First, we review the algorithm and then we consider the necessary conditions for its optimality. In the following, the superscript k is added to our notation to specify the iteration index of the algorithm. The procedure is as follows:

1. Start with the allocation $\mathbf{l}^0 = 0$.
2. $k = 1$.
3. $\mathbf{l}^k = \mathbf{l}^{k-1} + e_j$, where e_j is the jth unit vector and i is any index for which

$$\phi(a_j, l_j^{k-1} + 1) - \phi(a_j, l_j^{k-1})$$

is maximum.
4. If $L(\mathbf{l}^k) > L_{Max}$, terminate; otherwise $k \rightarrow k+1$ and go to step 3.

The algorithm starts from a zero allocation. At each step, one bit will be allocated to the coordinate which has the most $\phi(a_j, l_j^{k-1}+1) - \phi(a_j, l_j^{k-1})$ (result in the most reduction of total distortion). The procedure will be continued til we spend all of the bits.

Theorem: If $\phi(a, \ell)$ is concave and strictly increasing of ℓ then the procedure generates an optimal allocation (for proof refer to [19]).

We derive a set of conditions on the quantizer structure to assure the optimality of the incremental bit allocation. These conditions can be easily integrated in the iterative design algorithm for the quantizer design. We also present a dynamic programming formulation for the optimum 1 bit allocation and show that in practice the dynamic programming approach and and the simpler method based on incremental distribution of bits result in the same solution based on numerical results for a Gaussian source.

IV. Numerical Results

In the following, we present the numerical results for the performance and the complexity of the proposed methods for an i.i.d. Gaussian source. The quantization is measured in terms of the mean square distance. In all comparisons, the memory size is in byte (8 bits) per N dimensions and the computational complexity is the number of additions/comparisons per dimension.

Table (I) shows the numerical results of the proposed quantizers at different bit rates. The first method is linear programming with a constraint on total self information (LP-H). This method has the best performance. After applying Huffman code and imposing the constraint on length instead of self information, the performance drops about by 0.9 dB (LP-L,1D). This degradation will be smaller for larger values of bits per dimension. This degradation in SNR can be reduced by increasing the dimension or by merging two or more dimensions and applying a Huffman code to the corresponding subspaces (refer to Table I). It is observed that the SNR for the IP approach is very close to LP-L. This means that the degradation in the quality of solution caused by dropping the zero-one constraint is negligible.

Tables (II) provides the numerical results corresponding to fixed-rate entropy-coded trellis coded quantizer. First, we assigned Huffman code to one dimensional codewords (FETCQ,1D). The results show about 1 dB improvement in comparison with (LP-L). We merge two dimension together and apply the Huffman code for them (FETCQ,2D). The results show about 0.3 dB improvement in comparison with (FETCQ,1D).

In Table III we have a comparison between proposed methods and the method presented in [12,17]. We have compared the quantizer under equal conditions i.e., the length of codeword for the scalar quantizer codebook is derived from Huffman code. The proposed methods offer the same performance while the new approaches show a substantial reduction in complexity. In term of comparison of proposed methods in this paper, although the IP has a smaller complexity in comparison with LP, but the LP has the advantage that it can be applied to trellis encoding and achieve some packing gain.

Rate=2.5 bits/dimension, M=8				
Dimension	LP-H	IP,1D	LP-L,1D	LP-L,2D
32	13.02	11.93	11.91	12.51
64	13.20	12.26	12.25	12.83
128	13.29	12.46	12.45	13.03
256	13.36	12.58	12.57	13.24
512	13.39	12.64	12.64	13.30

TABLE I

SNR (in dB) vs. dimension of quantizer for a Gaussian source, 1D and 2D refer to using the Huffman code over 1 and 2 dimensional subspaces respectively.

Rate=2.5 bits/dimension M=8		
Dimension	FETCQ-1D	FETCQ-2D
32	13.34	13.63
64	13.56	13.93
128	13.73	14.13
256	13.80	14.25
512	13.84	14.32

TABLE II

SNR (in dB) vs. dimension FETCQ using linear programming approach using a four state trellis for a Gaussian source, 1D and 2D refer to using the Huffman code over 1 and 2 dimensional subspaces respectively.

Method	Add	Multiplies	Memory	SNR
Rate=3.5 bits/dimension M=16, N=32				
IP	4.5	3	129 byte	18.06
LP	76	64	1.57 k-byte	18.06
SVQ [12]	736	16	3.9 k-byte	18.06
Rate=2.5 bits/dimension 2M=16, N=32				
FETCQ (LP)	96	64	1.75 k-byte	13.56
TB-SVQ [17]	5776	16	15.5 k-byte	13.56

TABLE III

Comparison of proposed methods and references [12, 17], the number of state is four.

References

[1] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory,* vol. IT-6, pp. 7-12, March 1960.

[2] S. P. Lloyd, "Least square quantization in PCM," *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 129–137, Jan. 1982.

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

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

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

[6] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 37, pp. 31–42, Jan. 1989.

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

[8] W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. Acoust., Speech, Signal Process.,* vol. 37, pp. 1568–1575, Oct. 1989.

[9] D. P. de Garrido, W. A. Pearlman, and W. A. Finamore, "A clustering algorithm for entropy-constrained vector quantizer design with application in coding images pyramids," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 5, pp. 83–95, Apr. 1995.

[10] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory,* vol. IT-32, pp. 468–583, Nov. 1986.

[11] A. C. Hung, E. K. Tsern, and T. H. Meng, "Error-resilient pyramid vector quantization for image compression," *IEEE Trans. on Image. Process.,* vol. 7, pp. 1373–1386, Oct. 1998.

[12] R. Laroia and N. Farvardin, "A structured fixed-rate vector quantizer derived from variable-length scalar quantizer—Part I: Memoryless sources," *IEEE Trans. Inform. Theory,* vol. IT-39, pp. 851–867, May 1993.

[13] A. K. Khandani, "A Hierarchical Dynamic Programming Approach to Fixed-rate, Entropy-Coded Quantization," *IEEE Trans. Inform. Theory,* vol. IT-42, pp. 1298–1303, July 1996.

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

[15] T. R. Fischer and M. Wang, "Entropy-constrained trellis-coded quantization," *IEEE Trans. Inform. Theory,* vol. IT-38, pp. 415–426, March 1992.

[16] M. W. Marcellin, "On entropy constrained TCQ," *IEEE Trans. Commun.,* vol. 42, pp. 14–16, Jan. 1994.

[17] R. Laroia and N. Farvardin, "Trellis-based scalar-vector quantizer for memoryless sources," *IEEE Trans. Inform. Theory,* vol. IT-40, pp. 860–870, May 1994.

[18] A. K. Khandani, "A Linear (Zero-one) Programming Approach to Fixed-rate, Entropy-Coded Vector Quantization," *IEE Proceeding Communication,* vol. 146, pp. 275–282, October 1999.

[19] B. Fox, "Discrete optimization via marginal analysis," *Management science,* vol. 13, no. 3, pp. 210–216, Nov. 1966.