A Step-Wise Search Approach to Fixed-rate Entropy-coded Quantization

S. Nikneshan and A. K. Khandani

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Technical Report UW-E&CE#2001-10

September 2, 2001

Abstract: This report describes a new fixed-rate entropy-constrained quantization scheme for stationary memoryless sources where the structure of code-words is derived from a variable-length scalar quantizer. This method is a sequential encoding algorithm which looks for the quantizer partition in a number of steps. It starts from the quantization points which result in the minimum distance among the available candidates, and then in each step tries to reduce a given cost value[1] by changing the quantizer point along an appropriately selected dimension. The algorithm continues til the total cost of the encoded codeword becomes smaller than its maximum allowable value. We show that the step-wise algorithm results in a substantial reduction in the complexity, while the associated degradation in the performance is negligible.

1 Introduction

Optimum fixed-rate scalar quantizers, introduced by Max [2] and Lloyd [3] (LMQ), minimize the average distortion for a given number of threshold points. To increase the resolution of quantization in high probability regions, the threshold points are closely spaced in those regions, and widely spaced where the probability values are small. In spite of the gain of LMQ in comparison with the uniform quantization, there is a big gap between the LMQ performance and the rate distortion bound.

An extension of LMQ to vector quantizers introduced by Linde, Buzo, and Gray in [4] performs arbitrarily close to the rate-distortion bound as the quantizer dimension N becomes large. However, the implementation complexity of [4] (both computational and storage) is exponential in NR (where N is the dimension and R is the per sample rate), making it impractical even for modest values of rate and dimension. Suboptimal tree-searched vector quantizers [5] solve the computational complexity problem, but only at the cost of added storage complexity.

To improve the performance of scalar quantizers, one could use variable-length encoding of the quantizer output. Goblick and Holsinger [6] showed by numerical experiment that uniform scalar quantizers with variable rate coding perform within about 1.5 dB of the rate distortion bound for an i.i.d. Gaussian source. Optimal entropy-constrained scalar quantizers (ECSQ) minimize the average distortion for a given output entropy and are known to asymptotically (at high rates) perform within 1.53 dB of the rate-distortion bound for a large class of memoryless

[1]Cost is defined as an integer number proportional to the additive self information [1].

sources [7, 8, 9, 10, 11]. Wood [8] provided a numerical descent algorithm for designing an entropy-constrained scalar quantizer, and showed that the result was only slightly superior to a uniform scalar quantizer followed by a lossless code. Berger [9, 10] described a condition for optimality of an entropy-constrained scalar quantizer for squared-error distortion measure. He formulated the optimization as unconstrained Lagrangian minimization and developed an iterative algorithm for the design of entropy-constrained scalar quantizers.

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. This class of convex optimization problems in information theory was first presented by Blahut [12]. The Blahut algorithm, for finding the rate-distortion function, is based on minimizing a Lagrangian where the Lagrangian multiplier is interpreted as the slope of the hyper-plane supporting the convex optimization region. Chou and *et al.,* [13] present an algorithm for the entropy-constrained vector quantization (ECVQ) using Lagrangian formulation which uses a form of generalized Lloyd algorithm.

The improvement due to entropy coding comes at the cost of a variable-rate output with its concomitant difficulties. To take advantage of entropy coding, while avoiding the disadvantages 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) [14], is an example of FEVQ in which the code-vectors are located on the intersection of a cubic lattice and a pyramid in an N-dimensional space. For Laplacian sources, this quantizer is asymptotically optimal and achieves the performance of ECSQ.

One class of FEVQ schemes are based on using a subset of points from a lattice (quantization lattice) bounded within the Voronoi region around the origin of another lattice (shaping lattice) [15]. This approach has been extended in [15] to the case of using the trellis diagram of a convolutional code to construct the shaping lattice. In both cases, the selected subset forms a group under vector addition modulo the shaping lattice. This group property is used to facilitate the complexity of the underlying operations.

Another class of FEVQ schemes are based on selecting the N-fold symbols with the lowest additive self-information (typical set). In this case, the selected subset has a high degree of

structure which can be used to substantially reduce the complexity. This scalar-vector quantizer (named as SVQ [1]) is a vector quantizer derived from a variable-length scalar quantizer and can usually achieve a large portion of the boundary gain by placing the code-vectors on and inside the typical region. However, no granular gain is realized by SVQ as the underlying grid is rectangular. A method for exploiting this structure based on using a dynamic programming approach with the states corresponding to integer numbers proportional to the additive self information (cost) of the code-words is used by Laroia and Farvardin in [1]. The core idea in the schemes of [1] 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. The states s and $s+c$ in two successive stages are connected by a link corresponding to the one-D symbol(s) of cost c. Consequently, the states in the nth stage, $n=0,\ldots,N-1$, represent the accumulative cost over the set of the first n dimensions. The links connecting two successive stages are labeled by the corresponding one-D distortions. Then, the Viterbi algorithm is used to find the path of the minimum overall additive distortion through the trellis.

Reference [16] uses a different approach to dynamic programming showing improvement with respect to the schemes of [1]. The key point in [16] is to decompose the underlying operations into the lower dimensional subspaces. This decomposition avoids the exponential growth of the complexity. The core of the scheme, as in any problem of dynamic programming, is a recursive relationship which is formed in a hierarchy of levels (where each level involves the Cartesian product of the lower dimensional subspaces).

In this report, we introduce a reduced-complexity method for fixed-rate entropy-constrained quantization. The approach, a very low complexity algorithm, starts from an initial point and improves the quantization SNR in a number of subsequent steps. It is shown (through numerical simulations) that for an important class of sources with a monotone probability density function, the proposed quantizer offers a performance very close to an optimum search procedure (based on dynamic programming) with a substantial reduction in the complexity.

The rest of report is organized as follows: Section 2 explains the step-wise algorithm to fixed-rate entropy constrained quantization. The section includes the basic definition about the algorithm and a discussion about the cases where the proposed method results in the optimum solution. Finally, in Section 3, we conclude the report by presenting some numerical results and a comparison between proposed methods with some other quantization schemes.

2 Step-Wise Algorithm (SWA) to Fixed-rate Entropy Constrained Quantization

The methods known for solving a constrained optimization problem fall into two general categories. One class of procedures are based on starting from a point which satisfies the optimality condition, but is not feasible. In this case, the optimization routine gradually moves towards the feasible region, while optimizing the changes in the objective function value. A second class of procedures start from a feasible point which is not optimum, and gradually move towards a better solution, while maintaining the feasibility. Both of these procedures are applicable to our following discussion, however, we focus on the first approach as it turns out to have a smaller complexity for the cases which are of interest to us.

Consider an initial solution corresponding to the quantizer partitions with smallest distortion along each dimension. If this initial point satisfies the rate constraint, then the solution is complete, otherwise, we plan to move in a sequence of steps towards a feasible solution which satisfies certain optimality conditions. To formulate the procedure, we consider a subset of quantizer points along each dimension which can potentially become a component in the final solution. We refer to these subsets as the *candidate sets* corresponding to each dimension, denoted by $\mathcal{C}_i$, $i = 0, \ldots, N-1$. We assume that the candidate set $\mathcal{C}_i$ is composed of $c_i = |\mathcal{C}_i|$ elements ($c_i \leq M$, M is the number of threshold points along each dimension).

To formalize the definition, we assign a second set of indices to the quantizer partitions along each dimension to index the elements within each candidate set. We assume that the reconstruction level and the cost corresponding to the jth element of the candidate set for the ith dimension are equal to r_i^j and l_i^j, $i = 0, \ldots, N-1$, $j = 0, \ldots, c_i - 1$, respectively, and $d_i^j = (r_i^j - a_i)^2$, where a_i is the input sample along the ith co-ordinate. We also assume that the elements of each candidate set are ordered according to their distance from the corresponding input component with the nearest point indexed by zero. Using these notations, the candidate set for the ith dimension is defined as the collection of quantizer partitions satisfying,

$$d_i^0 \leq d_i^1 \ldots \leq d_i^{c_i}$$
$$l_i^0 \geq l_i^1 \ldots \geq l_i^{c_i}$$

This definition is justified noting that if two points α and β satisfy, $d_i^\alpha \leq d_i^\beta$ and $l_i^\alpha \leq l_i^\beta$, then α is always a better choice as compared to β, and consequently, $\beta \notin C_i$. We refer to $\Delta_i^j = l_i^j - l_i^{j+1}$,

$j = 0, \ldots, c_i - 1$, $i = 0, \ldots, N-1$, as the cost increments. Note that $\Delta_i^j \geq 0$.

We also assign a *shadow price* to the element of each candidate set, defined as,

$$s_i^j = \frac{d_i^{j+1} - d_i^j}{\Delta_i^j}, \ j = 0, \ldots, c_i - 2$$

We define the optimization problem to be concave if the shadow prices s_i^j, $i = 0, \ldots, N-1$, are non-increasing for $j = 1, \ldots, c_i - 1$. Using these notations, the step-wise optimization procedure is formulated as follows,

- Start from an initial solution for which the partition selected along each dimension is the element of the corresponding candidate set with the minimum distance. This is the element indexed by zero in the corresponding candidate set.
- Compute the overall cost of the current solution, namely $\hat{L}_k$, where k is the iteration index.
- If $\hat{L}_k \leq L_{\max}$, quit, otherwise find the dimension for which the current selected point has the smallest shadow price among all the current selected points of all dimensions, change the current selected point along this dimension to the next element in the corresponding candidate set, and go to step 2.

Reference [17] discusses a special case of this optimization procedure for which $\Delta_i^j = 1, \ \forall i, j$, in which case they show that the step-wise procedure results in the optimum solution for a concave function.

An important special case of our analysis is for situations that the quantizer points are labeled by a dyadic Huffman tree[2] and the cost of the quantizer partitions is defined as the binary length of their associated code-word.

Theorem 1: The step-wise optimization procedure results in the optimum solution for a concave quantizer with unity cost increments (for example a concave quantizer based on a dyadic tree labeling).

Proof: The proof follows by a direct interpretation of theorems presented in [17].

Theorem 2: Assuming a concave quantizer, the kth iteration of the step-wise optimization procedure outlined above results in the optimum solution for a problem with $L_{\max} = \hat{L}_k$.

[2]A dyadic tree has the lengths $1, 2, 3, \ldots, l_{\max} - 1, l_{\max}, l_{\max}$, where $l_{\max}$ is the maximum length.

Proof: Consider two subsequent elements of a given candidate set, say $r_i^j, r_i^{j+1} \in \mathcal{C}_i$ with $\Delta_i^j = l_i^j - l_i^{j+1} = m$ where $m > 1$. To fill the gap between l_i^j and l_i^{j+1}, we consider a set of hypothetical quantizer partitions with code-word costs $l_i^j - 1, l_i^j - 2, \ldots, l_i^j - m + 1$, all with the same shadow price of $(d_i^{j+1} - d_i^j)/m$. The resulting hypothetical quantizer will be concave (if the original quantizer is concave), and will satisfy the conditions of Theorem 1 as applied to a quantizer with unity length increments. This means that the step-wise optimization procedure results in the optimum solution for the resulting hypothetical concave quantizer. On the other hand, as the shadow prices corresponding to the hypothetical quantization partitions used to fill a given gap are all the same, the step-wise optimization procedure will select these hypothetical partitions subsequent to each other. In this case, the solution to the original problem (obtained by step-wise algorithm) will coincide with that of the modified problem for similar values of $L_{\max}$, and consequently, is optimal.

3 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. A quantizer with a search mechanism based on a Dynamic Programming Algorithm (DPA) is used as the benchmark for comparison.

Table 1 and 2 shows the numerical results of the proposed quantizers at two different bit rates. For Table 1, the set of codeword lengths (costs) is $\{4, 4, 3, 2, 2, 3, 4, 4\}$, and for Table 2 the set of codeword lengths is $\{7, 7, 6, 5, 4, 3, 3, 3, 3, 3, 3, 4, 5, 6, 7, 7\}$ which are the lengths associated with the Huffman code designed in the last iteration of the employed iterative (LBG type) design algorithm. It was observed that for a Gaussian source and this set of codeword lengths, the quantizer is concave, and consequently satisfies the optimality conditions of Theorem 1 (concave with unity cost increments).

In order to evaluate the performance of SWA for a concave quantizer but not with unity incremental cost values, we present the numerical results for a quantizer with $M = 10$ and different set of codeword lengths as given in Table 3. In Table 3, the codeword lengths of half of the points are shown. Due to the symmetry of the Gaussian source with respect to the

origin, the same lengths apply to the other half. The SNR values presented in Table 3 show a negligible degradation in performance for the case that the quantizer does not have unity cost increments.

In Table 4, we have a comparison in terms of SNR and complexity between SWA and DPA for a quantizer with unity cost increments. As Table 4 shows, the proposed algorithm offers a substantial reduction in the complexity.

For the further evaluation of the proposed algorithm, we have tested the SWA with the lexicographic indexing. The lexicographic indexing and using DPA for codebook search has been presented in [1]. In this method, the cost associated with each threshold point is proportional to its self information, namely, $\lfloor -B \log_2(p) \rceil$ where p is the corresponding probability and B is a scaling factor employed to reduce the effects of round-off error. Larger values of B improve the quantizer performance at the price of an increase in the complexity. In this case, the quantizer does not have unity length increments, however, it was observed that it satisfies the condition of concavity for the cases studied. The numerical results shown in Table 5 and 6 indicates that even for the case that the quantizer does not have unity length increments, the SWA algorithm performs well. The gap between SWA and DPA in this case is at most 0.15 dB while the complexity of SWA is much lower than DPA.

In summary, the SWA performs close to optimum performance for a concave quantizer and in the case of a quantizer with unity length increments, it achieves the optimum performance. Its negligible complexity makes this method very attractive in comparison with dynamic programming approach while the corresponding degradation in performance is quite negligible.

References

[1] 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.

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

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

[4] 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.

[5] R. M. Gray and Y. Lindo, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Communication,* vol. COM-30, pp. 381–385, Feb. 1982.

[6] 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, Sept. 1967.

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

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

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

[10] T. Berger, "Minimum entropy quantizers and permutation codes," *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 149–157, March 1982.

[11] 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.

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

[13] 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.

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

[15] M. V. Eyuboglu and G. D. Forney, "Lattice and trellis quantization with lattice- and trellis- bounded codebooks—high-rate theory for memoryless sources," *IEEE Trans. Inform. Theory,* vol. IT-39, pp. 46–59, Jan 1993.

[16] 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.

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

Dimension	SWA/DPA (dB)
32	11.93
64	12.26
128	12.46
256	12.58
512	12.64
1024	12.71

Table 1: SNR of SWA/DPA (in dB) vs. dimension for a rate of 2.5 bits/dimension, $M = 8$ (using Dyadic Huffman tree labeling).

Dimension	SWA/DPA (dB)
32	18.06
64	18.50
128	18.79
256	18.99
512	19.10
1024	19.17

Table 2: SNR of SWA/DPA (in dB) vs. dimension for a rate of 3.5 bits/dimension, $M = 16$ (using Dyadic Huffman tree labeling).

Set of lengths	Rate	M	N	SWA (dB)	DPA (dB)
$\{1, 2, 3, 4, 4\}$	2.5	10	32	11.74	11.74
$\{1, 2, 3, 5, 5\}$	2.5	10	32	11.64	11.66
$\{1, 2, 4, 6, 7\}$	2.5	10	32	11.03	11.06
$\{1, 3, 4, 6, 6\}$	2.5	10	32	9.23	9.36
$\{1, 3, 4, 6, 6\}$	3.0	10	32	12.20	12.29

Table 3: Performance comparison of SWA vs. DPA.

Rate=3.5 bits/dimension $M = 16$, $N = 32$				
Method	Add/dimension	Multiplies/dimension	Memory	SNR (dB)
SWA	3	3	96 byte	18.06
DPA	688	16	3.6 k-byte	18.06

Table 4: Performance/Complexity comparison of SWA vs. DPA.

Rate=2.5 bits/dimension, $N = 32$				
	$M = 8$		$M = 10$	
B	SWA (dB)	DPA (dB)	SWA (dB)	DPA (dB)
4	12.06	12.11	12.08	12.13
8	12.50	12.60	12.51	12.60
16	12.71	12.83	12.78	12.85
32	12.81	12.96	12.87	—-
64	12.91	13.05	12.92	—-
128	12.92	13.07	12.95	—-

Table 5: SNR vs. dimension of SWA in comparison with DPA for $N = 32$, cost of the codewords is computed as $\lfloor -B \log_2(p) \rceil$.

Rate=2.5 bits/dimension, $N = 64$				
	$M = 8$		$M = 10$	
B	SWA (dB)	DPA (dB)	SWA (dB)	DPA (dB)
4	12.08	12.16	12.08	12.19
8	12.59	12.64	12.58	12.64
16	12.89	12.91	12.81	12.93
32	12.96	13.02	13.02	13.10
64	13.11	13.14	13.12	—-
128	13.13	13.17	13.17	—-

Table 6: SNR vs. dimension of SWA in comparison with DPA for $N = 64$, cost of the codewords is computed as $\lfloor -B \log_2(p) \rceil$.