A Sequential Search Approach to Fixed-rate Entropy-coded Quantization *

Sasan Nikneshan and Amir K. Khandani

Dept. of Elec. and Comp. Engineering

University of Waterloo

Waterloo, Ontario N2L 3G1

Tel. (519) 885-1211 x5324, Fax. (519) 888-4338

Abstract: This paper describes a new Fixed-rate Entropy-constrained Vector Quantization (FEVQ) scheme for stationary memoryless sources based on a sequential search procedure. It is shown that the proposed algorithm results in a substantial reduction in the complexity while the degradation in performance is negligible.

1 Introduction

To improve the performance of a scalar quantizer, one could use variable-length entropy coding of the quantizer output. To take advantage of entropy coding, while avoiding the dis-advantages associated with 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) [2], is an example of FEVQ in which the code-vectors are located on the intersection of a cubic lattice and a pyramid. Another class of FEVQ schemes are based on using a subset of points from a lattice bounded within the Voronoi region around the origin of another lattice [3]. The dominant technique for FEVQ is based on selecting the N-fold symbols with the lowest additive self-information. In this case, the selected subset has a high degree of structure which can be used to reduce the complexity.

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

A method for exploiting this structure based on using dynamic programming with the states corresponding to integer numbers proportional to the additive self information (cost) of the code-words is used in [1]. The core idea in [1] is to use a trellis where states s and $s+c$ in two successive stages are connected by a link corresponding to the one-D symbol(s) of cost c. Then, the Viterbi algorithm is used to find the path of the minimum overall distortion through the trellis.

In this paper, we introduce a reduced-complexity method for FEVQ. The algorithm starts from an initial point with lowest quantization noise and then moves towards a feasible point (satisfying the rate constraint) in a number of subsequent steps. Numerical simulations show that the proposed quantizer offers a performance very close to the optimum method (using dynamic programming) with a substantial reduction in the complexity.

2 Sequential Search Algorithm (SSA) to FEVQ

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*, denoted by $\mathcal{C}_i$, $i = 0, \ldots, N-1$. Note that the cardinality of $\mathcal{C}_i$ satisfies $c_i = |\mathcal{C}_i| \le M$ where M is the number of threshold points along each dimension. We assume that the elements of each candidate set are ordered according to their distance from the input with the nearest point indexed by zero. The reconstruction level and the cost corresponding to the jth element of the candidate set for the ith dimension are denoted as, r_i^j and l_i^j, $i = 0, \ldots, N-1$, $j = 0, \ldots, c_i - 1$, respectively. Define $d_i^j = (r_i^j - a_i)^2$, where a_i is the ith input sample. Note that if two points α and β satisfy, $d_i^\alpha \le d_i^\beta$ and $l_i^\alpha \le l_i^\beta$, then α can be always selected as a better or comparable choice as compared to β, and consequently, $\beta \notin \mathcal{C}_i$. As a result, for the ith candidate set, we have,

$$\begin{array}{c} d_i^0 < d_i^1 \ldots < d_i^{c_i} \\ l_i^0 > l_i^1 \ldots > l_i^{c_i} \end{array}$$

We refer to $\Delta_i^j = l_i^j - l_i^{j+1} > 0$, $j = 0, \ldots, c_i - 2$, $i = 0, \ldots, N-1$, as the cost increments. We assign a *shadow price* to the element of each candidate set, defined as,

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

The optimization problem is said to be concave if the shadow prices s_i^j, $i = 0, \ldots, N-1$, are non-increasing for $j = 0, \ldots, c_i - 2$ [5].

Using these notations, the SSA is formulated as follows,

1. Start from an initial solution for which the partition selected along each dimension is the element with the minimum distance.

2. Compute the overall cost of the current solution, namely $\hat{L}_k$, where k is the iteration index.

3. If $\hat{L}_k \leq L_{\max}$, quit, otherwise find the dimension for which the current selected point has the smallest shadow price, change the current selected point along this dimension to the next element in the corresponding candidate set, and go to step 2.

Theorem: Assuming a concave quantizer, the kth iteration of the SSA results in the optimum solution for a problem with $L_{\max} = \hat{L}_k$.

Proof: Refer to [4] for proof.

Reference [5] discusses a special case of this optimization procedure for which $\Delta_i^j = 1,\ \forall i, j$, in which case it is shown that the sequential search procedure results in the optimum solution for a concave function. An important special case of our analysis, satisfying the above optimality condition, occurs when the quantizer points are labeled by a Huffman tree where the code-word lengths are either equal, or change in unity increments (with the cost defined as the binary length of the code-words).

3 Numerical Results

Numerical results are generated 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 shows the results at two different bit rates. For $R = 2.5$, $M = 8$, the codeword lengths (costs) are $\{4, 4, 3, 2, 2, 3, 4, 4\}$, and for $R = 3.5$, $M = 16$, the codeword lengths are $\{7, 7, 6, 5, 4, 3, 3, 3, 3, 3, 3, 4, 5, 6, 7, 7\}$. These are the lengths associated with the Huffman code designed in the last iteration of the employed iterative (LBG type [6]) design algorithm. Note that both cases have unity cost increments. We have observed that the quantizer is also concave, and consequently, satisfies the optimality conditions.

In Table 2, we have a comparison in terms of SNR and complexity between SSA and DPA for a quantizer with unity cost increments. As Table 2 shows, the proposed algorithm offers a substantial reduction in the complexity.

The lexicographic indexing and using DPA for codebook search has been presented in [1]. We have tested the SSA with the lexicographic indexing where the cost associated with each point is defined as, $[-B \log_2(p)]$ where $[.]$ denotes rounding, p is the probability and B is a scaling factor used to reduce the effects of round-off error [1]. 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 cost increment, however, the numerical results shown in Table 3 indicates that the SSA algorithm performs very close to DPA.

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

[3] 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.

[4] S. Nikneshan, Lattice/Trellis based Fixed-rate Entropy-constrained Quantization, Ph.D. thesis, E&CE Dept., Univ. of Waterloo, Oct. 2001.

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

[6] 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.

N	$R = 2.5,\ M = 8$ SSA/DPA (dB)	$R = 3.5,\ M = 16$ SSA/DPA (dB)
32	11.93	18.06
64	12.26	18.50
128	12.46	18.79
256	12.58	18.99
512	12.64	19.10
1024	12.71	19.17

Table 1: SNR of SSA/DPA (in dB) vs. dimension.

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

Table 2: Performance/complexity comparison of SSA vs. DPA.

	$N = 32$		$N = 64$	
B	SSA (dB)	DPA (dB)	SSA (dB)	DPA (dB)
4	12.06	12.11	12.08	12.16
8	12.50	12.60	12.59	12.64
16	12.71	12.83	12.89	12.91
32	12.81	12.96	12.96	13.02
64	12.91	13.05	13.11	13.14
128	12.92	13.07	13.13	13.17

Table 3: SNR vs. dimension of SSA in comparison with DPA, cost is computed as $[-B\log_2(p)]$ where p is the probability.