

# Enhanced Multistage Vector Quantization by Joint Codebook Design

Wai-Yip Chan, Smita Gupta, and Allen Gersho

**Abstract**—Multistage vector quantization (MSVQ) can achieve very low encoding and storage complexity in comparison to unstructured vector quantization. However, the conventional stage-by-stage design of the codebooks in MSVQ is suboptimal with respect to the overall performance measure. We introduce an algorithm for the joint design of the stage codebooks to optimize the overall performance. The performance improvement, although modest, is achieved with no effect on encoding or storage complexity and only a slight increase in design effort.

## I. INTRODUCTION

IN vector quantization (VQ), an input vector of signal samples or parameters is quantized by selecting the best matching *representation* or *reproduction* vector from a codebook of  $2^{kr}$  stored code vectors of dimension  $k$ . VQ is an optimal coding technique in the sense that all other methods of coding a random vector in  $k$  dimensions with a specific number  $b = kr$  of bits are equivalent to special cases of VQ with generally suboptimal codebooks. However, optimal VQ assumes a single and possibly very large codebook with no imposed constraints on its structure. The resulting encoding and storage complexity, of the order of  $2^{kr}$ , may be prohibitive for many applications. To circumvent the complexity problem, constraints are frequently imposed on the VQ codebook structure and/or on the encoding algorithm.

A structured VQ scheme which can achieve very low encoding and storage complexity is multistage VQ (MSVQ) [1]. In MSVQ, the  $kr$  bits are divided between  $L$  stages, with  $b_i$  bits for stage  $i$ ,  $i \in \{1, \dots, L\}$ , and  $\sum_{i=1}^L b_i = kr$ . Each stage consists of a *stage codebook* with  $2^{b_i}$  code vectors of dimension  $k$ . A representation  $\hat{X}$  of an input random vector  $X$  is formed by selecting from each stage codebook a stage code vector,  $\hat{y}_i$  for stage  $i$ , and forming their sum, i.e.,  $\hat{X} = \sum_{i=1}^L \hat{y}_i$ . Thus, the storage complexity of MSVQ is  $\sum_{i=1}^L 2^{b_i}$  vectors, which can be much less than the complexity of  $\prod_{i=1}^L 2^{b_i} = 2^{kr}$  vectors for unstructured VQ.

In MSVQ, as originally proposed by Juang and Gray [1], the stage codebooks are searched *sequentially*. The input vector  $X$  is quantized with the first stage codebook producing the selected first-stage code vector  $\hat{y}_1 = Q_1(X)$  where  $Q_i(\cdot)$  denotes the  $i$ th stage quantization operation. A *residual* vector  $E_1$  is formed by subtracting  $\hat{y}_1$  from  $X$ . Then  $E_1$  is quantized

using the second stage codebook, with exactly the same procedure as in the first stage, but now with  $E_1$  instead of  $X$  as the input to be quantized. Thus, in each stage except the last ( $L$ th) stage, a residual vector is generated and passed to the next stage to be quantized independently of the other stages. The decoder receives for each stage an index identifying the stage code vector selected and forms the reproduction  $\hat{X}$  by summing the identified vectors. The overall quantization error  $X - \hat{X}$  is equal to the quantization residual from the last stage,  $E_L$ . Sequential searching of the stage codebooks renders the encoding complexity similar to the storage complexity,  $\sum_{i=1}^L 2^{b_i}$ .

While MSVQ incurs some performance penalty over unstructured VQ, it is nevertheless of great value because of the substantially reduced encoding and storage complexity. Its suboptimality is due to three characteristics of MSVQ:

- the use of multiple stage codebooks to generate the reproduction,
- the convenient, but suboptimal, sequential search performed by the encoder, and
- the use of a sequential design method for generating the stage codebooks.

Abandoning a) implies returning to unstructured VQ and losing the complexity advantages of MSVQ. Characteristic b), sequential search, can be replaced with a joint search where the set of stage code vectors is selected by minimizing an *overall distortion* over all of the  $2^{kr}$  combinations of possible selections. This was studied in [2]. However, the encoding complexity of joint search is similar to that of unstructured VQ so that much of the appeal of MSVQ is lost. While a) and b) are essential to retain the complexity advantages of MSVQ, characteristic c), the conventional design method, is indeed suboptimal and an improved design algorithm can indeed enhance performance.

The conventional way of designing the stage codebooks for sequential search MSVQ with a mean-squared error (MSE) overall distortion criterion is to apply the generalized Lloyd algorithm (GLA) [3] stage-by-stage for a given training set of input vectors. The GLA is first applied to the training set to design the first stage codebook. A training set for the second stage is then constructed by pooling together the residual vectors resulting from the quantization of the original training set with the first stage codebook. The second stage codebook is then designed with the GLA applied to the derived residual training set. This process is repeated to design subsequent stages. Each of the stage codebooks is designed to minimize the average distortion incurred in quantizing the vectors applied to that stage. Clearly, a better scheme would be to design the stage codebooks *jointly* to minimize the

Paper approved by the Past Editor-in-Chief of the IEEE Communications Society. Manuscript received July 16, 1991. This work was supported in part by the University of California MICRO program, Rockwell International Corporation, Compression Labs. Inc., and Eastman Kodak Company.

A. Gersho and S. Gupta are with the Center for Information Processing Research, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

W.-Y. Chan is with the Department of Electrical Engineering, McGill University, Montreal, P.Q. H3A 2A7, Canada.

IEEE Log Number 9203580.

0090-6778/92\$03.00 © 1992 IEEE

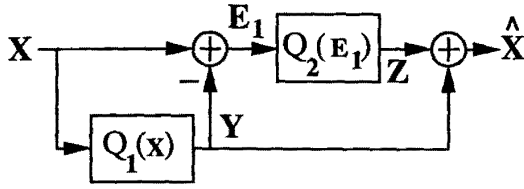


Fig. 1. Multistage VQ with two stages.

overall distortion. In order to obtain a joint codebook design algorithm, it is helpful to find optimality conditions analogous to the classical ones of unstructured VQ [4]. We first review the basic operation of MSVQ.

## II. MSVQ OPERATION

For simplicity, the development in this paper is restricted specifically to MSVQ with only two stages, since the ideas can be trivially extended to more stages. The basic structure for 2-stage MSVQ is depicted in Fig. 1. The quantizer  $Q_1$  in stage 1 uses a codebook  $Y = \{y_1, y_2, \dots, y_N\}$  and the stage 2 quantizer  $Q_2$  uses a codebook  $Z = \{z_1, z_2, \dots, z_M\}$ . The performance objective to be minimized for the quantizer is the overall MSE given by

$$D = E\{d(X, \hat{X})\} = E\{\|X - \hat{X}\|^2\} \quad (1)$$

where the expectation  $E\{\cdot\}$  is taken over the probability distribution function (pdf) of  $X$ . The MSVQ encoder generates indexes  $I$  and  $J$  respectively for the first and second stages and sends the indexes to the decoder. The encoder chooses the code vectors  $y_I$  and  $z_J$  for the quantization of an input vector  $X$  as follows. First the code vector  $y_I$  in codebook  $Y$  nearest in Euclidean distance to the input  $X$  is found such that

$$\|X - y_I\|^2 \leq \|X - y_i\|^2, \quad i = 1, 2, \dots, N \quad (2)$$

A residual vector  $E_1$  is then formed

$$E_1 = X - y_I \quad (3)$$

The residual  $E_1$  is quantized with the code vector  $z_J$  which minimizes the Euclidean distance over all other code vectors in codebook  $Z$

$$\|E_1 - z_J\|^2 \leq \|E_1 - z_j\|^2, \quad j = 1, 2, \dots, M \quad (4)$$

In this way, after  $N + M$  distance computation and comparison operations, the input vector  $X$  is quantized to  $\hat{X}$  where

$$\hat{X} = y_I + z_J. \quad (5)$$

Note that the quantization error in the last (second) stage is equal to the overall error between the original and reproduction vector

$$X - \hat{X} = X - y_I - z_J = E_1 - z_J = E_1 - \hat{E}_1. \quad (6)$$

In general, in every stage except the last stage of a sequential search MSVQ, the selected stage code vector does not minimize the overall distortion. However, the code vector selected from the last stage does minimize the overall distortion given the selections from all prior stages.

The encoding procedure determines a partition of the space of input vectors into a set of disjoint regions  $W_{ij}$ . The region  $W_{ij}$  is the set of all input vectors that are encoded with the index pair  $(i, j)$  and which will be quantized with the reproduction value  $y_i + z_j$ . This region is given by

$$W_{ij} = \{x : \|x - y_i\|^2 \leq \|x - y_n\|^2, n = 1, \dots, N; \\ \|x - y_i - z_j\|^2 \leq \|x - y_i - z_m\|^2, m = 1, \dots, M\}. \quad (7)$$

It should be noted that for a given set of stage codebooks, the encoder's sequential search operation is suboptimal and the partition it induces is suboptimal. Nevertheless, it can be viewed as an approximation to the optimal partition resulting from the complexity constraint that the search be performed sequentially.

An optimal partition would be obtained if the encoder performed a *joint search* over all possible index pairs for the pair  $I$  and  $J$  that minimizes the overall distortion over all possible reproduction vectors  $y_i + z_j$ .

$$\|X - y_I - z_J\|^2 \leq \|X - y_i - z_j\|^2 \quad \text{for} \\ i = 1, 2, \dots, N, \quad j = 1, 2, \dots, M. \quad (8)$$

Thus, in this case the input vector  $X$  would be compared with each of the  $NM$  possible reproduction vectors. Even though joint search MSVQ and unstructured VQ have similar encoding complexity, the constraint imposed by the multistage structures results in substantial performance degradation; this is illustrated below with numerical results. Unless memory is severely limited and computational power is abundant, joint search MSVQ is not likely to be a favorable alternative to unstructured VQ. Nevertheless, the performance of joint search MSVQ provides a useful bound on the performance achievable with sequential search MSVQ and serves as a useful reference.

## III. MSVQ OPTIMALITY CONDITIONS

Consider first how the decoder can be optimized for a given MSVQ encoder. The standard MSVQ decoder is simply a table lookup. From the encoder, the decoder receives indexes  $i$  and  $j$ , identifying respectively stage-1 code vector  $y_i$  and stage-2 code vector  $z_j$ . Although it is straightforward to find  $NM$  optimal reproductions, one for each partition region  $W_{ij}$ , this would give us a single large codebook for the decoder. There is no expeditious way to translate these optimal reproduction values into separate codebooks for each stage, yet we wish to preserve the reduced storage feature of MSVQ. We circumvent this difficulty by introducing new partitions of the input space, each associated with a particular stage codebook and considering the constrained optimization of a particular stage codebook when other stage codebooks are fixed.

We define the sets

$$R_i = \bigcup_{m=1}^M W_{im}, \quad S_j = \bigcup_{n=1}^N W_{nj}. \quad (9)$$

Thus,  $R_i$  is the set of all input vectors that will lead to first-stage index  $i$  being transmitted to the receiver and  $S_j$

is the set of all input vectors that will lead to the transmission of second-stage index  $j$  by the encoder. The set of regions  $\{R_1, R_2, \dots, R_M\}$  forms a partition of the input space associated with stage 1. Similarly, the set of regions  $\{S_1, S_2, \dots, S_N\}$  forms a different partition of the input space associated with stage 2.

For a fixed stage 2 codebook and for a given stage 1 partition of the input space, the stage 1 codebook is optimal if for each  $i$ ,  $y_i$  minimizes the overall MSE in the region  $R_i$ . Thus, the optimal value of  $y_i$  is the value of  $y$  that minimizes the conditional distortion

$$D_i = E\{\|X - y - Q_2(X - y)\|^2 | X \in R_i\}. \quad (10)$$

With  $y$  as an argument of the stage 2 quantization function  $Q_2$ , an explicit solution for this minimization appears to be intractable. Although (10) may be minimized by an iterative procedure, we take an alternative approach by constraining the encoder to be fixed while optimizing the first stage decoder codebook. Then we seek  $y$  to minimize the average distortion:

$$D'_i = E\{\|X - y - U\|^2 | X \in R_i\} \quad (11)$$

where  $U = Q_2(X - y_i)$ . Introducing a vector parameter  $v$ , (11) can be expanded as follows:

$$\begin{aligned} D'_i &= E\{\|X - U - v + v - y\|^2 | X \in R_i\} \\ &= E\{\|X - U - v\|^2 | X \in R_i\} + \|v - y\|^2 \\ &\quad + 2E\{(X - U - v)^T(v - y) | X \in R_i\}. \end{aligned} \quad (12)$$

Since  $v$  is an arbitrary, nonrandom, parameter, we may set  $v = E\{X - U | X \in R_i\}$ . Then, the third term vanishes regardless of the value of  $y$ . Since the second term is always nonnegative, we have the condition

$$E\{\|X - U - y\|^2 | X \in R_i\} \geq E\{\|X - U - v\|^2 | X \in R_i\} \quad (13)$$

where equality is achieved if and only if  $y = v$ . Thus, (13) shows that for any given stage 1 encoder code vector  $y_i$ , we can generate an optimal  $i$ th code vector for the first stage decoder according to

$$y'_i = E\{X - Q_2(X - y_i) | X \in R_i\} \quad (14)$$

We call this value a *conditional centroid* for the first stage.

For a fixed encoder and fixed first stage decoder codebook, the stage 2 decoder codebook is optimal if for each  $j$ ,  $z_j$  minimizes the overall MSE in the region  $S_j$ . The optimal value of  $z_j$  is the value of  $z$  that minimizes the conditional distortion

$$D_j = E\{\|X - Q_1(X) - z\|^2 | X \in S_j\}. \quad (15)$$

The minimizing  $z$ , which we call a conditional centroid for the second stage is very simply

$$z'_j = E\{X - Q_1(X) | X \in S_j\} = \{E_1 | X \in S_j\}. \quad (16)$$

Equation (14) provides the key design condition to enhance the first-stage decoder codebook for a given encoder and a given second stage decoder codebook. Similarly, (16)

provides the key design equation for enhancing the second-stage decoder given the encoder and the first stage decoder codebook.

For a given decoder, we need to find an optimal sequential search encoder. If the decoder and the first stage encoder are fixed, the best second-stage encoder codebook is clearly the decoder's second-stage codebook. If the decoder and the second-stage encoder are fixed, a reasonable but not necessarily optimal choice for the first-stage encoder codebook is the current first-stage decoder codebook. These results provide the basis for the design introduced below.

#### IV. MSVQ DESIGN

In this section, we show that the conventional stage-by-stage design for sequential search MSVQ can be improved by joint design where the stage codebooks are reoptimized iteratively and alternately to minimize the overall distortion objective. The iterative algorithm is somewhat similar to the GLA. However, unlike the GLA, it does not necessarily produce a non-increasing sequence of distortion values.

Suppose we have a training set  $T$ , consisting of  $T$  training vectors  $x_t, t = 1, 2, \dots, T$ , that are statistically representative of the random vector source  $X$ . The use of a training set is equivalent to replacing the pdf of the source with a probability mass function where each training vector has equal probability. The overall MSE distortion measure can then be approximated by the MSE incurred in quantizing the training set

$$D\{d(X, \hat{X})\} = \frac{1}{T} \sum_{t=1}^T \|x_t - \hat{x}_t\|^2 \quad (17)$$

where  $\hat{x}_t$  are the quantized training vectors.

We will use the parenthesized superscript  $(n)$  to indicate variables in the  $n$ th iteration of the design procedure with initial value of  $n = 0$ . Suppose we have initial codebooks  $Y^{(0)}$  and  $Z^{(0)}$  for the first and second stages, respectively. These initial codebooks can readily be obtained from the traditional stage-by-stage MSVQ design. For monitoring the performance associated with the codebooks  $Y^{(n)}$  and  $Z^{(n)}$  at the  $n$ th iteration, let  $d_{n+1}$  denote the MSE value in (17) computed using these codebooks. A very large value is assigned to  $d_0$ .

The initial codebooks are optimized by the following algorithm.

##### A. Enhanced Design Algorithm

1) *Encode and Partition Training Set*: Encode each vector  $x_t$  in the training set  $T$  with the current ( $n$ th) set of stage codebooks. Let  $i^*(t)$  and  $j^*(t)$  denote the indexes obtained in encoding  $x_t$ . For  $i = 1, \dots, N$ , let  $R_i^{(n+1)}$  be the set of training vectors assigned indexes  $i^* = i$ . Compute  $d_{n+1}$ , the MSE incurred in encoding  $T$  according to (17).

2) *Test*: If the quotient  $|d_n - d_{n+1}|/d_n$  is less than a predetermined small positive threshold  $\delta$  or if  $n$  has reached a predetermined upper limit, the algorithm is terminated. The final codebooks are then taken as  $Y^{(n^*)}$  and  $Z^{(n^*)}$  where  $n^*$  is such that  $d(n^*) \leq d(n)$  for all  $n$ .

3) *Update Codebook 1*: Each code vector in stage 1 is replaced by the conditional centroid according to (14). The

new code vectors  $\mathbf{y}_i^{(n+1)}$ , for  $i = 1, \dots, N$ , constitute the new codebook  $\mathbf{Y}^{(n+1)}$  for stage 1.

4) *Encode and Repartition Training Set*: Encoding is performed again as in step 1, but using  $\mathbf{Y}^{(n+1)}$  instead of  $\mathbf{Y}^{(n)}$ , to produce a new set of index labels  $i^*(t)$  and  $j^*(t)$  for the training vectors. The vectors are thereby partitioned into  $M$  sets,  $\mathcal{S}_j^{(n+1)}$ ,  $j = 1, \dots, M$ , with all vectors having the same  $j^*$  index assigned to the set  $\mathcal{S}_j^{(n+1)}$ .

5) *Update Codebook 2*: A new second stage codebook  $\mathbf{Z}^{(n+1)}$  is obtained by replacing the  $j$ th code vector,  $j = 1, \dots, M$ , with the conditional centroid according to (16).

6) *Increment*: Increment  $n$  by 1 and go to Step 1.

The enhanced MSVQ design algorithm above applies the necessary optimality conditions, (14) in step 3 and (16) in step 5, to improve the stage codebooks. In this sense the algorithm is somewhat similar to the popular GLA. However, the GLA is a strict descent algorithm; each step of the GLA either decreases or leaves unchanged the overall MSE. In the enhanced MSVQ design algorithm, the sequential encoding in steps 1 and 4 leads to suboptimal partitions; only by jointly searching the codebooks would the encoder generate optimal partitions. The consequence of a suboptimal encoder is that even though step 3 is optimal in the sense of improving the stage 1 decoder codebook, the improved codebook might not be optimal for the encoder. If the overall MSE were also computed in step 4, it is possible for the value to be larger than that computed in preceding step 1. For a two-stage MSVQ, step 3 is the only operation in the algorithm that could increase the MSE; this property is corroborated by our simulations.

The possibility of a nonmonotonic MSE raises the issue of how to effectively terminate the iterative process. Many *ad hoc* termination schemes can be devised, and we have suggested one scheme in step 2. This step may, for instance, be modified so that the algorithm will terminate only when the relative change in  $d_n$  is less than  $\delta$  for several consecutive steps, or when the total number of algorithm steps has reached a given limit.

Another consequence of the nonmonotonicity in  $d_n$  is that the final stage codebooks at termination may not be the best ones. This problem is easily solved by choosing the stage codebooks from an intermediate iteration with the lowest  $d_n$ . In our simulation,  $d_n$  is decreasing most of the time, though it might fluctuate when its relative change has diminished after several steps of monotonic descent. The simplicity of sequential search makes it feasible to run the algorithm for many iterations (30 was usually adequate in our simulations) and then choose the best overall codebook set.

We investigated whether better performance could be achieved by modifying the algorithm so that the evolution of the overall distortion is guaranteed to be non-increasing. We note that the encoder's selection of code vectors from the last stage minimizes the overall distortion given the selections made from prior stages. Thus, we altered the algorithm to disable the update of stage 1 encoder codebook while still allowing the stage 1 decoder codebook to change according to step 3; this entailed keeping two copies of the stage 1 codebook and modifying step 3 so that only the decoder's copy was updated. Though this modification led to a nonincreasing MSE,

TABLE I  
SNR FOR MSVQ OF AUDIO DCT COEFFICIENTS 2.5 b/samp, 4-dim,  
10 bit CODEBOOKS NOTE: SNR FOR UNSTRUCTURED VQ = 13.22 dB

Stage 1/Stage2 Bit Allocation	SNR (dB)		
	Design Methods		
	Standard	Enhanced	Joint-Search
5/5	11.62	11.93	12.44
6/4	11.61	12.00	12.45
4/6	11.81	11.99	12.57
7/3	11.79	12.11	12.54
3/7	11.93	12.06	12.65

the resultant performance never exceeded that of the enhanced algorithm in the form given above. However, this does justify step 3 despite it being a source of suboptimality. An alternate strategy for tackling the nonmonotonicity involves skipping step 3 whenever the step would lead to degraded performance. We did not pursue this because the current algorithm converged in virtually all cases we tested.

## V. SIMULATION RESULTS AND DISCUSSION

The enhanced MSVQ design algorithm has been applied to the design of stage codebooks for normalized DCT coefficient vectors extracted from wide-band audio signals sampled at 32 kHz (Table I). We have also designed codebooks for vectors consisting of samples of speech waveforms sampled at 8 kHz as well as for blocks of pixels from gray-scale images. However, Table I adequately captures the essence of our results. The training set for the results in Table I was formed by grouping four adjacent DCT coefficients into a vector and normalizing each vector by a quantized "gain" [5]. The table shows the overall signal-to-noise ratio (SNR) values for various configurations of two-stage, four-dimensional, 10 bit MSVQ. Each configuration represents a different allocation of bits between the two stages. The SNR performance results for sequential search MSVQ are shown for two different design methods: a) conventionally designed MSVQ, b) our enhanced MSVQ design algorithm. Also, for comparison, the SNR performance of joint search MSVQ using the design of Barnes [2] is also shown. Note that joint search MSVQ requires a high complexity encoder rather than the standard MSVQ encoding structure shown in Fig. 1. The conventionally designed codebooks also served as the initial codebooks for the enhanced design algorithm. Of primary interest here are the improvements in SNR obtained from the enhanced algorithm rather than the absolute values, which are highly dependent on the statistics of the underlying signals. From the table, it is evident that in all cases the enhanced design algorithm results in a performance improvement over the conventional MSVQ design; however, the improvements are fairly modest and do not exceed 0.4 dB in all MSVQ configurations for the audio application.

When each configuration with asymmetrical bit allocations between the two stages is compared with its counterpart configuration with swapped bit allocation (such pair of configurations have the same complexity), the configuration with more bits in the first stage always benefits more from the enhanced design. Note that the difference in the performance

of conventional MSVQ between the two asymmetric configurations with the same complexity is quite small. With the enhanced design, the difference in performance between the two configurations is nearly removed. Also, the difference in performance between the traditional MSVQ and the joint search MSVQ is between 0.72–0.85 dB. Relative to this performance difference, the modest improvements achieved with the enhanced design seem worthwhile, especially considering the order of magnitude lower encoding complexity of sequential search MSVQ.

It should be noted that since the conventional design always minimizes the first stage MSE without regard to the effect on overall performance, then it follows that the enhanced design will inevitably increase the MSE of the first stage even though the overall MSE has been lowered. This means that the enhanced design algorithm is not suited to applications where the stage codebooks are employed as embedded or multiresolution quantizers.

## VI. CONCLUSION

We have presented an iterative design algorithm for the standard MSVQ coding system. The proposed algorithm has consistently but moderately improved performance over the conventional design. This is achieved without affecting a very appealing property of MSVQ, its low encoding and storage complexity. Though the enhanced design algorithm does not guarantee a monotonic evolution in the overall

MSE, the algorithm did furnish improved codebooks for all cases tested and with minimal incremental design effort. The similarity of the algorithm to the GLA raises the possibility of additional performance enhancement by applying an emerging class of stochastic algorithms (see, e.g., [6],[7]). Moreover, the optimization procedure can be readily applied to MSVQ with more than two stages and to hybrids between tree structured VQ and MSVQ [5]. Another application is the joint optimization of codebooks in "partitioned VQ" [4], which can be regarded as a constrained form of MSVQ. Partitioned VQ, also called "split VQ", is an efficient approach for quantizing linear predictive coding parameters in speech coders [8].

## REFERENCES

- [1] B. H. Juang and A. H. Gray, "Multiple stage vector quantization for speech coding," in *Proc. ICASSP*, Paris, France, Apr. 1982, pp. 597–600.
- [2] C. F. Barnes, "Residual quantizers," Ph.D. dissertation, Brigham Young Univ., Dec. 1989.
- [3] 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.
- [4] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. New York: Kluwer Academic, 1992.
- [5] W. Y. Chan and A. Gersho, "Constrained-storage vector quantization in high fidelity audio transform coding," in *Proc. ICASSP*, May 1991, pp. 1109–1112.
- [6] J. Vaisey and A. Gersho, "Simulated annealing and codebook design," in *Proc. ICASSP*, Apr. 1988, pp. 1176–1179.
- [7] K. Zeger, J. Vaisey, and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 40, pp. 310–322, Feb. 1992.
- [8] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," in *Proc. ICASSP*, May 1991, pp. 661–664.