A Generalized Multistage VQ Approach for Spectral Magnitude Quantization

Çağrı Özenç Etemoğlu and Vladimir Cuperman

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
E-mail:[cagri, vladimir]@ece.ucsb.edu

Abstract
This paper presents a novel vector quantization (VQ) technique in which the quantized vector is formed by adding the transformed outputs of a multistage codebook rather than just adding the outputs of the stages as in regular multistage vector quantization (MSVQ). The transformations are selected from a family of linear transformations represented by a codebook of matrices. This technique can be viewed as a generalized form of MSVQ. If the transformations are constrained to be the identity transformation, this technique becomes identical to the regular MSVQ. The design algorithm is based on joint optimization of the linear transformations and the stage codebooks. It is shown that the proposed technique yields high quality spectral magnitude quantization with performance exceeding that of multistage vector quantization (MSVQ) of similar complexity and bit rate.

1. Introduction
Efficient spectral magnitude quantization is needed in many low bit rate speech coders employing a sinusoidal model [1] [2] [3] [4] [5]. The spectral magnitudes are obtained by sampling the spectrum of either the speech or the LP residual at frequencies corresponding to pitch harmonics. This procedure generates a variable dimension vector since the number of pitch harmonics changes in time.

In this work, the variable dimension spectral vector is first transformed into a fixed dimension vector, and then the fixed dimension vector is quantized efficiently using the proposed VQ technique. The fixed dimension is chosen such that there is no modeling distortion caused by transformation. The best way to quantize the spectral magnitudes is to use a single stage optimal VQ. However the variable dimension spectral magnitudes typically require 10-20 bits for each 10 ms, which results in excessive requirements on the complexity and training set size needed to design the optimal VQ.

2. Problem formulation
Let \( \mathbf{x} \) be an \( M \)-dimensional input vector. According to the proposed approach, the quantized vector \( \hat{\mathbf{x}} \) is given by

\[
\hat{\mathbf{x}} = \sum_{i=1}^{L} \hat{T}_{i-1} \hat{\mathbf{c}}_i
\]

where \( \hat{\mathbf{c}}_i \) represents a codevector which is a member of the \( i \)th stage codebook \( \mathbf{C}_i \). \( \hat{T}_1 \) represents the linear transformation matrix determined by the selection of \( \hat{\mathbf{c}}_1 \). \( \hat{T}_1 \) is chosen from the matrix codebook \( \mathbf{C}_T \) for \( i = 1, 2, \cdots, L - 1 \). \( \hat{T}_0 \) is the identity matrix and \( L \) denotes the number of stages.

If the transformations \( \hat{T}_i \) for \( i = 1, 2, \cdots, L - 1 \) are constrained to be identity matrices, then the quantized vector \( \hat{\mathbf{x}} \) is

\[
\hat{\mathbf{x}} = \sum_{i=1}^{L} \hat{\mathbf{c}}_i
\]
as in regular MSVQ.

For the clarity of the presentation, from hereon we will concentrate on the two stage \((L = 2)\) case. The equations and rules derived for the two stage case can easily be applied to cases with more than two stages. Using two stages, the quantized vector \(\hat{x}\) is given by

\[
\hat{x} = \hat{c}_1 + \hat{T}_1 \hat{c}_2
\]

where \(\hat{c}_i\) represents a codevector which is a member of the \(i\)th stage codebook \(C_i\) for \(i = 1, 2\). \(\hat{T}_1\) represents the linear transformation matrix determined by the selection of \(\hat{c}_1\). \(\hat{T}_1\) is chosen from the matrix codebook \(C_{T1}\).

The quantization distortion criterion is WMSE where the weights are given by a diagonal weighting matrix \(\mathbf{W}\). The average distortion \(D\) on a set of \(N\) vectors \(\{x_k\}\) with weighting matrices \(\mathbf{W}_k\) is

\[
D = \frac{1}{N} \sum_{k=0}^{N-1} (x_k - \hat{x}_k)^T \mathbf{W}_k (x_k - \hat{x}_k)
\]

or, based on (3):

\[
D = \frac{1}{N} \sum_{k=0}^{N-1} ||x_k - \hat{x}_{1,k} - \hat{T}_{1,k} \hat{c}_{2,k}||^2 \mathbf{W}_k
\]

where \(\hat{T}_{1,k}\) is the transformation matrix and \(\hat{c}_{1,k}, \hat{c}_{2,k}\) are the stage codevectors corresponding to the input vector \(x_k\).

The objective here is to design the codebooks \(C_{T1}\), \(C_1\), \(C_2\) that minimize (5) and to develop an efficient coding rule for this VQ technique.

### 2.1. Encoding/Decoding

Given the linear transformation codebook \(C_{T1}\) and the first, second stage codebooks \(C_1\), \(C_2\) the optimal triplet \((\hat{T}_1, \hat{c}_1, \hat{c}_2)\) for encoding the vector \(x\) is given by

\[
(\hat{T}_1, \hat{c}_1, \hat{c}_2) = \arg \min_{T_1 \in C_{T1}, c_1 \in C_1, c_2 \in C_2} ||x - \hat{c}_1 - \hat{T}_1 \hat{c}_2||^2 \mathbf{W}
\]

The minimization required in (6) is computationally intensive if an exhaustive search is employed. To avoid high search complexity, a sequential search is employed as in MSVQ where \(\hat{c}_1\) is determined first. Furthermore, the first stage codebook \(C_1\), and the linear transformation codebook \(C_{T1}\) are related such that

\[
\hat{c}_1 = c_1, j, \hat{T}_1 = T_{1,j} \Leftrightarrow c_{1,j} = \arg \min_{c_{1,j} \in C_1} ||x - c_{1,j}||^2 \mathbf{W}
\]

where \(c_{1,j}\) and \(T_{1,j}\) are the \(i\)th entries of their respective codebooks.

Note that the search in (7) has the same computational complexity as the usual VQ search, and it enables the reconstruction of the quantized vector with two indices (one index per stage as in regular MSVQ). However, the use of transforms allows us to trade-off a larger memory (required for storing the transforms) for improved performance.

Once the vector \(c_{1,j}\) is determined, the associated linear transformation \(T_{1,j}\) is employed to search the second stage by choosing \(c_{2,j}\) to minimize

\[
c_{2,j} = \arg \min_{c_{2,m} \in C_2} ||x - c_{1,j} - T_{1,j} c_{2,m}||^2 \mathbf{W}
\]

The quantized vector is given by \(\hat{x} = c_{1,j} + T_{1,j} c_{2,j}\). Depending on the memory and complexity requirements the search in (8) can be done by either generating the transformed vectors using matrix multiplication at the time of search, or storing pre-computed transformed vectors. In the former case, the complexity is larger than MSVQ, while in the latter case the computational complexity is practically the same as in MSVQ.

### 3. Joint codebook optimization

In order to jointly optimize the codebooks, we use an iterative sequential optimization. The algorithm iterates between optimizing the first stage codebook \(C_1\) and the associated linear transformation codebook \(C_{T1}\) for a given second stage codebook \(C_2\), and optimizing the second stage codebook for the given first stage and linear transformation codebooks.

In order to sequentially optimize the codebooks, the input vector space is partitioned with respect to the codebook whose entries are being optimized. Let \(R_{i,j}\) denote the set of input vectors whose assigned indices are \(i\) for the codebook \(C_1(C_{T1})\), and \(j\) for the codebook \(C_2\). Given \(R_{i,j}\), the set of input vectors assigned to the \(i\)th entry of the codebook \(C_1(C_{T1})\) is given by

\[
U_i = \bigcup_{j=1}^{N_2} R_{i,j}
\]

and the set of vectors assigned to the \(j\)th entry of residual codebook \(C_2\) is

\[
V_j = \bigcup_{i=1}^{N_1} R_{i,j}
\]

where \(N_2\) is the size of \(C_2\) and \(N_1\) is the size of both \(C_1\) and \(C_{T1}\).
3.1. Design of the first stage and linear transformation codebooks for a given second stage codebook

Given the fixed second stage codebook and the partition \( U_i \), our objective is to compute \( C_{1,i} \) and \( T_{1,i} \) for \( i = 1, \ldots, N_1 \) to minimize (5). In other words, \( C_{1,i} \) and \( T_{1,i} \) are obtained as the solution of the optimization problem

\[
(C_{1,i}, T_{1,i}) = \arg \min_{C_1, T_1} \sum_{k: x_k \in U_i} \|x_k - \hat{c}_1 - \hat{T}_1 \hat{c}_{2,k} \|^2 W_k
\]  

(11)

The minimization defined above can be changed into a minimization over only \( \hat{T}_1 \), using the fact that the optimal \( \hat{c}_1 \) for a given \( \hat{T}_1 \) is

\[
\hat{c}_1 = \left( \sum_{k: x_k \in U_i} W_k \right)^{-1} \sum_{k: x_k \in U_i} W_k (x_k - \hat{T}_1 \hat{c}_{2,k})
\]  

(12)

Substituting (12) into (11) we obtain

\[
T_{1,i} = \arg \min_{T_1} \sum_{k: x_k \in U_i} \|x_k - \left( \sum_{n: x_n \in U_i} W_n \right)^{-1} \sum_{n: x_n \in U_i} W_n (x_n - \hat{T}_1 \hat{c}_{2,n}) - \hat{T}_1 \hat{c}_{2,k} \|^2 W_k
\]  

(13)

The solution of the minimization problem defined by (13) may not be unique. The \( j \)th row of \( T_{1,i} \), \( r_{ij} \) will be chosen as the solution with the minimum norm and is given by

\[
r_{ij}^* = Z_i^T Y_{ij}^+ \quad j = 1, \ldots, M
\]  

(14)

where

\[
Z_i^T = \begin{bmatrix} [u_{1,i}^{1/2} x_{1,i}] & \cdots & [u_{i,i}^{1/2} x_{i,i}] \\ \vdots & \ddots & \vdots \\ [u_{M,i}^{1/2} x_{M,i}] \end{bmatrix} - \\
( \sum_{k: x_k \in U_i} w_{k} \hat{c}_1 - \hat{T}_1 \hat{c}_{2,k} ) \left( \sum_{k: x_k \in U_i} w_{k,i} \hat{c}_1 \right)^{-1} \left( \sum_{k: x_k \in U_i} w_{k} \hat{c}_1 x_{k,i} \right) \left[ u_{1,i}^{1/2} \right] \\
\vdots \\
\left[ u_{M,i}^{1/2} \right] \nonumber
\]

\[
Y_{ij} = \begin{bmatrix} [u_{1,i}^{1/2} \hat{c}_{2,1}] & \cdots & [u_{i,i}^{1/2} \hat{c}_{2,i}] \\ \vdots & \ddots & \vdots \\ [u_{M,i}^{1/2} \hat{c}_{2,M}] \end{bmatrix} - \\
( \sum_{k: x_k \in U_i} w_{k} \hat{c}_{2,k} ) \left( \sum_{k: x_k \in U_i} w_{k,i} \hat{c}_{2,k} \right)^{-1} \left( \sum_{k: x_k \in U_i} w_{k} \hat{c}_{2,k} x_{k,i} \right) \left[ u_{1,i}^{1/2} \right] \\
\vdots \\
\left[ u_{M,i}^{1/2} \right] \nonumber
\]

(15)

\( Y_{ij}^+ \) denotes the pseudoinverse of \( Y_{ij} \), \( w_{k,j} \) denotes \( j \)th diagonal element of the \( k \)th weight matrix, and \( |U_i| \) denotes the cardinality of \( U_i \). Once \( T_{1,i} \) is calculated, \( C_{1,i} \) can simply be computed using (12) with \( \hat{T}_1 = T_{1,i} \).

3.2. Design of the second stage codebook for a given first stage and linear transformation codebooks

Given the fixed first stage, linear transformation codebooks and the partition \( V_j \), we will compute \( C_{2,j} \) for \( j = 1, \ldots, N_2 \) to minimize (5). Hence \( C_{2,j} \) will be given by

\[
c_{2,j} = \arg \min_{\hat{c}_2} \sum_{k: x_k \in V_j} \|x_k - \hat{c}_{1,k} - \hat{T}_1 \hat{c}_{2,k} \|^2 W_k
\]  

(16)

The solution of the above minimization may not be unique. The minimum norm centroid for the \( j \)th cluster is computed as

\[
c_{2,j} = A^+ b
\]  

(17)

where

\[
A = \begin{bmatrix} W_{1/2} \hat{T}_{1,1} \\
\vdots \\
W_{1/2} |V_j| \hat{T}_{1,|V_j|} \end{bmatrix}, \quad b = \begin{bmatrix} W_{1/2} \left( x_{1,1} - \hat{c}_{1,1} \right) \\
\vdots \\
W_{1/2} |V_j| \left( x_{|V_j|} - \hat{c}_{1,|V_j|} \right) \end{bmatrix}
\]  

(18)

3.3. Joint codebook design

The main design algorithm can now be stated by using the centroid computations and the sequential encoding rule described in earlier sections.

Once the codebooks are initialized, the main design algorithm performs the following steps:

1. Partition the training set to obtain \( R_{i,j} \).
2. Compute the overall distortion, if termination criterion is satisfied then stop else continue.
3. Compute the optimal codebook \( C_{1,j} \) using (14), and the optimal codebook \( C_{1} \) using (12).
4. Partition the training set to obtain a new \( R_{i,j} \).
5. Compute the optimal codebook \( C_{2} \) using (17).
6. Go to 1.

While steps 3 and 5 of this algorithm always decrease the overall distortion, the partitioning steps 1 and 4 may increase the distortion due to the suboptimal sequential encoding rule. Hence, the algorithm does not guarantee strict descent, however, in practice the distortion generally decreases. The termination criterion adopted in this algorithm is to stop when the relative change in the distortion is less than a given threshold.
4. Experimental results

In order to evaluate the performance of the proposed VQ technique for spectral magnitude quantization, we compared this technique with a two stage MSVQ. The speech material used is sampled at 8 kHz and consists of sentences spoken by male and female speakers. The spectral vectors are extracted in linear prediction (LP) residual domain every 10 ms and normalized by a gain factor. The variable dimension spectral vectors are converted into fixed dimension of \( M = 48 \) by zero padding. The vectors used in training and testing have dimension 48 or smaller. Therefore the distortion incurred consists of quantization distortion only, since there is no modeling distortion.

The objective performance is measured by using weighted signal to noise ratio (WSNR), which is defined as

\[
WSNR = \frac{1}{N} \sum_{k=0}^{N-1} \frac{\|S_k\|_W^2}{\|S_k - \hat{S}_k\|_W^2}
\]

(19)

where \( S_k \) denotes the spectral magnitude vector.

The weighting function used in the experiments is obtained by combining the spectral magnitude of the LP synthesis filter with a perceptual weighting filter. The weighting matrix \( W \) is diagonal and \( i \)th diagonal element corresponding to the \( i \)th spectral sample at the frequency \( \omega_i \) is given by

\[
w_i = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} \left| z = \exp(j\omega_i) \right|, \quad 0 \leq \gamma_2 < \gamma_1 \leq 1
\]

(20)

where \( A(z) \) is the LP filter, and \( \gamma_1 = 0.92, \gamma_2 = 0.7 \).

In our experiments, a set of 257425 vectors are used for training, and another set of 85808 vectors are used for testing. The performance of the proposed and the multistage VQ over the training set measured in terms of WSNR (dB) is shown in Table 1. Table 2 shows the performance over the test set. The design is done for various typical bit rates used in low bit rate speech coders. As shown, the proposed technique has better performance than the two stage MSVQ.

5. Summary

In this paper we introduced a new VQ technique in which the quantized vector is obtained by adding the transformed outputs of a multistage codebook. A design algorithm for the proposed vector quantizer is given and an efficient sequential encoding method is described. Although the VQ technique presented can be applied to any vector quantization problem, in this work its application to spectral magnitude vector quantization is studied. Experimental results show that this technique outperforms MSVQ of similar complexity and bit rate, by allowing us to trade-off a larger memory for improved performance.

6. References


