Exhibit J

HIERARCHICAL VECTOR QUANTIZATION OF SPEECH WITH DYNAMIC CODEBOOK ALLOCATON

Allen Gersho and Yair Shoham

Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106

ABSTRACT

This paper introduces a Hierarchical Vector Quantization (HVQ) scheme that can operate on "supervectors" of dimensionality in the hundreds of samples. HVQ is based on a tree-structured decomposition of the original supervector into a large number of low dimensional vectors. The supervector is partitioned into subvectors, the subvectors into minivectors and so on. The "glue" that links subvectors at one level to the parent vector at the next higher level is a feature vector that characterizes the correlation pattern of the parent vector and controls the quantization of lower level feature vectors and ultimately of the final descendant data vectors. Each component of a feature vector is a scalar parameter that partially describes a corresponding subvector. The paper presents a three level HVQ for which the feature vectors are based on subvector energies. Gain normalization and dynamic codebook allocation are used in coding both feature vectors and the final data subvectors. Simulation results demonstrate the effectiveness of HVQ for speech waveform coding at 9.6 and 16 Kb/s.

1. INTRODUCTION

Speech waveforms may contain significant correlation over hundreds of samples, i.e., corresponding to basic sound units of duration 0.1 seconds or larger. Although vector quantization (VQ) is in principle the ideal way to encode a block or segment of this size, the associated computational and storage complexities are astronomically high. Existing suboptimal VQ schemes that trade performance for a reduced complexity are still limited to dimensionalities in the range of 10 to 20. For a review of current work in VQ, see [15] and [16].

The Hierarchical Vector Quantization (HVQ) scheme proposed in this paper can operate on vectors of dimensionality in the hundreds of samples. HVQ is particularly suitable for speech waveform coding because speech is characterized by large correlated segments, due to its quasi-periodicity and its slowly varying production mechanism. The correlation pattern within a speech segment, or some other feature of its internal structure, may be described by a properly defined feature vector of dimensionality much lower than that of the segment itself. This vector contains valuable side information about the segment, which can be used to efficiently code lower dimensional subsets of the segment.

This work was supported in part by the Air Force Office of Scientific Research under grant AFOSR 82-0008.

HVQ is based on a tree-structured decomposition of the given large dimensionality input vector, called the *supervector*. The supervector is split into subvectors, each subvector is then split into lower dimensional minivectors, and so on until the lowest level subvectors, called cells, are obtained. At each level (except the bottom level), a feature vector is extracted from each data vector at that level. The feature vectors are themselves vector quantized and used to control the bit allocation for quantizing lower level feature vectors and ultimately the cells themselves. Each quantized feature vector contains a partial description of the data vector at that level and provides a vital unit of side information about the supervector characteristics. The bottom level is composed of low dimensional vectors, the cells, which are efficiently quantized using the acquired side information.

The essence of this scheme lies in the definition of the various feature vectors and in the way they are used in the coding process. It is crucial that these vectors be highly correlated and easy to extract from the data vectors. Also, the way vectors from different levels are interrelated during the coding process must be simple and efficient.

This paper focuses on a three level HVQ using the Euclidean norm of a subvector as a basic feature. Normalization and codeword allocation are used in exploiting the side information during the coding process.

In Section 2, the system is introduced and explained in detail. Section 3 discusses the dynamic codeword allocation problem. Section 4 contains a discussion on the design of the various codebooks used in the system. Section 5 considers the capability of the system as a variable dimension VQ. Finally, Section 6 describes simulation results.

2. THREE LEVEL HVQ - GENERAL DESCRIPTION

The proposed scheme is shown in Figure 1. The high dimensional supervector, $X=(x_1,x_2, \cdots ,x_K)$, where K is the dimension of the input supervector, is partitioned into M subvectors: $X=(X_1,X_2,\ldots,X_M)$ where X_i denotes the i^{th} subvector. An M-dimensional feature vector, denoted by $S=(s_1,s_2, \cdots ,s_M)$, is defined and associated with the vector on the basis of one scalar feature per subvector. In the proposed coder, this feature is simply the norm of the corresponding subvector, although other definitions are possible. s_i is, therefore given by:

$$s_i = \|X_i\| = \left[\sum_{x \varepsilon X_i} x^2\right]^{\frac{1}{2}} \qquad (1)$$

Each subvector is further partitioned into L cellvectors, called cells, of dimension $k=K/(ML)$: $X_i=(X_{i1},X_{i2},\ldots,X_{iL})$ where $X_{i,j}$ is the j^{th} cell of the i^{th} subvector. An L-dimensional feature vector, denoted by

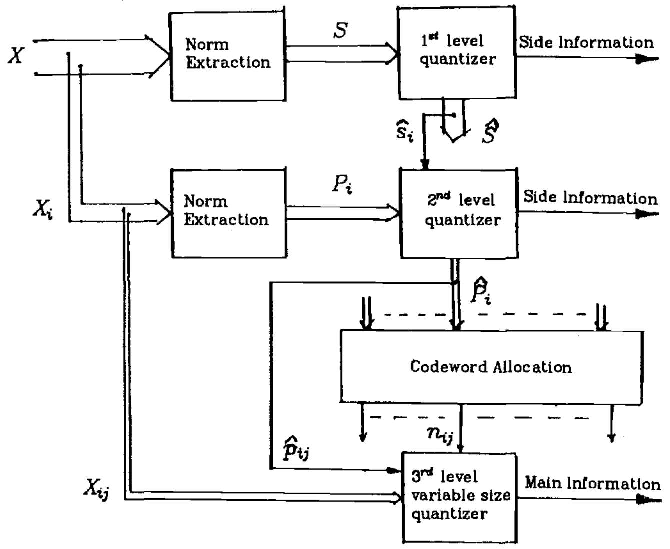

Figure 1. Three level HVQ scheme.

$P_i=(p_{i1},p_{i2},..,p_{iL})$, is associated with each subvector and consists of the corresponding cell norms. Thus, the norm of the j^{th} cell in the i^{th} subvector is given by:

$$p_{ij} = \|X_{ij}\| = \left[\sum_{x \varepsilon X_{ij}} x^2\right]^{\frac{1}{2}} \tag{2}$$

The quantization is carried out in a hierarchical fashion: In the first level, the M-dimensional vector S is quantized to $\hat{S}$ using one first level codebook. In the second level, each L-dimensional vector P_i is normalized using $\hat{S}_i$ and then quantized, using a second codebook. In the third level, each cell is normalized by the corresponding quantized norms from the second level and quantized by a set of third level variable size codebooks.

The main idea of this quantization structure is that some features of the entire supervector characteristics are passed all the way down to every minivector, thereby, influencing the way it is quantized. One way in which the quantization of an individual cell is influenced by the entire input speech segment is the normalization process mentioned above. The second way is through ***dynamic codebook allocation*** whereby the coder uses the quantized vectors of norms to optimally allocate a codebook of an appropriate size to each cell while maintaining a fixed number of bits for the overall specification of the supervector.

3. THE CODEBOOK ALLOCATION PROBLEM

The codebook allocation takes place in the third level where each cell is coded using a codebook of different size. Through this operation (and the normalization process - to be discussed later) coding of each individual cell is affected by the entire block pattern in such a way as to minimize the overall distortion.

The problem is to find the allocation rule, or, the mapping from the quantized version of the norms $\hat{p}_{ij}$ to the corresponding codebook sizes denoted by n_{ij}. The codebook sizes are, more conveniently, expressed by the required number of bits as: $r_{ij}=\log_2 n_{ij}$. One basic constraint is imposed on these allocation values, that is:

$$\sum_{i=1}^{M}\sum_{j=1}^{L} r_{ij} = R = K(b-b_s) \tag{3}$$

where R is the overall number of bits assigned to the main information (the quantized cells), b is the overall system bit rate and b_s is the bit rate of the side information (the quantized norms). Both b and b_s are given in bits/sample. Since there are practical limitations on the number and size of the third level codebooks, r_{ij} can only assume values from a given finite set:

$$r_{ij} \ \varepsilon \ \left\{B_1,B_2,....,B_q\right\} = \text{B} \tag{4}$$

Also, the allowed codebook sizes 2^{B_m}, $m=1,..,q$, must be non-negative integers. A reasonable, but not necessary, choice for the set B is $\{0,1,2,..,q\}$.

The optimal allocation rule aims at minimizing an average of a given distortion measure, subject to the above constraints. The distortion measure is defined over the entire supervector and denoted by $d(X,\hat{X})$, where $\hat{X}$ is the quantized supervector. For mathematical tractability we impose an additivity restriction on the distortion measure, to be decomposable in terms of subvectors and cells:

$$d(X,\hat{X}) = \sum_{i=1}^{M} d_i(X_i,\hat{X}_i) = \sum_{i=1}^{M}\sum_{j=1}^{L} d_{ij}(X_{ij},\hat{X}_{ij}) \tag{5}$$

This also decouples different cells with respect to their contribution to the overall distortion. This means that each cell can be assigned a measure d_{ij} independently of other cells. It is easy, in this case, to achieve noise shaping by assigning the cells different weights.

The definition of the features as norms of of subvectors is particularly suited to the following distortion measure:

$$d(X,\hat{X}) = (X-\hat{X})^t\, W(X-\hat{X}) = \sum_{i=1}^{M}\sum_{j=1}^{L} w_{ij}\, \|X_{ij}-\hat{X}_{ij}\|^2 \tag{6}$$

where W is a triangular non negative matrix whose elements on the main diagonal are w_{ij}. The constants w_{ij} may be chosen to achieve a desired shaping of the quantization noise. If no shaping is desired, we set $w_{ij}=1$ and get the usual Euclidean distance for the distortion measure. In the subsequent discussion we assume that $w_{ij}=1$ for simplicity. It can be shown that the only change required to account for the case where $w_{ij}\neq 1$ is to replace p_{ij} by $p'_{ij}=\sqrt{w_{ij}}p_{ij}$.

Using (6) the expected value of the distortion measure $D=E\left\{d(X,\hat{X})\right\}$. may be written in terms of conditional expectations as follows:

$$D = E\left\{\sum_{i=1}^{M}\sum_{j=1}^{L} D_c(\hat{p}_{ij},r_{ij})\right\} \tag{7}$$

where

$$D_c(\hat{p}_{ij},r_{ij}) = E\left\{\|X_{ij}-\hat{X}_{ij}\|^2 / \ \hat{p}_{ij},r_{ij}\right\} \tag{8}$$

This expression is conditioned on the scalars p_{ij} and r_{ij} which are randomly related to each other. Nevertheless, the allocation algorithm (to be determined) deterministically maps the *set* P=$\{p_{ij}\ ,\ i=1,M\ ,\ j=1,L\}$ to the *set* R=$\{r_{ij}\ ,\ i=1,M\ ,\ j=1,L\}$ To minimize (7), the double summation has to be minimized over all possible mappings, subject to the constraints imposed on r_{ij}.

The optimization problem can, now, be stated as follows: Given the set P, minimize $\sum_{i=1}^{M}\sum_{j=1}^{L} D_c(\hat{p}_{ij},r_{ij})$ over all possible sets R, subject to $\sum_{j=1}^{L}\sum_{i=1}^{M} r_{ij}=R$ and $r_{ij}\varepsilon$B

To solve the problem, the conditional expectations $D_c(p_{ij},r_{ij})$ should be explicitly expressed in terms of $\hat{p}_{ij}$ and r_{ij}. This functional dependency, which is crucial to the allocation algorithm, is not given. However, a good estimate of it can be found, based on the following arguments.

As will be explained in the next section, the cell quantizer is of gain-shape type which means that the quantized cell is given in the form

$$\hat{X}_{ij} = \hat{p}_{ij} A \tag{9}$$

where A denotes a codevector from a shape codebook of size n_{ij}. Using (9) we get:

$$D_c(\hat{p}_{ij}, r_{ij}) = p_{ij}^2 E\left\{ \| \frac{X_{ij}}{\hat{p}_{ij}} - A \|^2 / \hat{p}_{ij} \right\} \tag{10}$$

We, now, assume that conditional expectation in (10) is essentially independent of $\hat{p}_{ij}$. This assumption, which has been verified experimentally, is heuristically explained by the fact that due to the large long-term dynamic range of the speech waveform, the cell energy is essentially independent of the cell shape. Therefore, the performance of the shape codebook in coding the cell shape, is almost independent of the gain of the cell. As a result, (10) can be rewritten as:

$$D_c(\hat{p}_{ij}, r_{ij}) = \hat{p}_{ij}^2 G(r_{ij}) \tag{11}$$

$G(r)$ represents the distortion of a normalized cell quantizer as a function of the codebook size (in bits). Values of $G(r)$ versus r, for $r \varepsilon$B, are obtained during the codebook design. Experimental data shows that $G(r)$ can, quite accurately, be modeled as:

$$G(r) = 2^{-\alpha r} \tag{12}$$

The parameter α can be found from a best fit to the design data, and is approximately equal to $2/k$. Using (12) the problem reduces to that of minimizing:

$$\sum_{j=1}^{L} \sum_{i=1}^{M} \hat{p}_{ij}^2 2^{-\alpha r_{ij}} \tag{13}$$

with respect to the set R, for a given set P, subject to (3) and (4).

We briefly outline two methods for solving the problem. In the first method the problem is, initially solved without the constraint (4), i.e., r_{ij} are assumed to be continuous but non-negative values. This problem is solved using variational calculus techniques. See [1]. The resulting r_{ij}'s are, then, quantized to the nearest value in the set B, as required by (4). This last operation may violate the constraint (3), but, by properly constructing the set B, the effect of this violation can be made negligible.

The second method is based on the "Marginal Returns" approach [2] and provides an exact solution. It works in the case where the set B is uniform, i.e., B=$\{0, t, 2t, .., qt\}$. t is the bit allocation increment. The process goes as follows: r_{ij} are initialized to zero and the marginals $\hat{p}_{ij}^2(G(r_{ij}) - G(r_{ij}+t))$ are found for all i,j. r_{ij}, for which the marginal is maximum, is incremented by t bits. These steps are repeated until (9) is satisfied.

4. THE CODEBOOK DESIGN

Three codebooks must be designed. The first level codebook is used in coding the vector S. The second level codebook codes each of the vectors P_i, $i=1,..,M$ and provides the set P which determine the allocation set R. A set of q third level codebooks of sizes $B_1, B_2, .., B_q$ (in bits) is used to code the cells X_{ij}. This set constitutes the dynamically controlled, variable size third level codebook.

The first level codebook is designed as a standard VQ using the LBG algorithm [3] with the Euclidean distance for the distortion measure.

Both the second and the third level quantizers are based on a special form of a gain-shape quantizer [4] where the gain is provided externally and is not subject to optimization. Figure 2 depicts the structure of such a

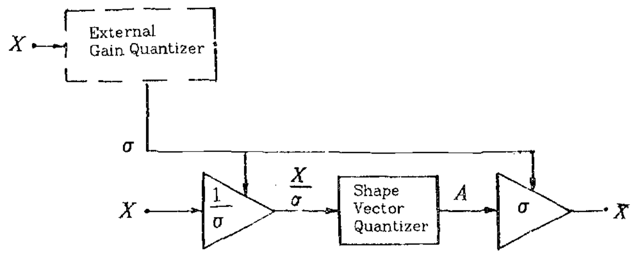

Figure 2. Gain normalized vector quantizer.

quantizer. As shown, the quantized version of the input vector X is given by:

$$\hat{X} = \sigma A \tag{14}$$

where σ, an input gain value, is supplied by another external quantizer. A represents a codevector from a shape codebook. The design objective is to optimize this shape codebook taking into account the statistical relationship of the supplied gain to the input vector. The basic idea behind this structure is the utilization of the information, carried by σ about the input vector, to improve the coder performance.

The design of this coder is carried out using an algorithm, similar to the LBG algorithm. The coding (partition) rule and the centroid calculation are modified to account for the special situation but the logical flow of the algorithm is exactly the same as described in [3].

The coder has, in fact, two inputs, X and σ, which may be combined into one augmented input vector (X,σ). The coding rule is defined over the space of the augmented input. Let us denote by A_j the j^{th} codevector in a codebook of size M. The optimal coding rule is, then, given by:

$$\| \frac{X}{\sigma} - A_{j^*} \| \leq \| \frac{X}{\sigma} - A_j \| \quad ; \quad j=1,..,M \tag{15}$$

where j^* is the index of the best codeword, given the input (X,σ). This coding rule implies a partition over the augmented space, whose individual subset is denoted by C_j and given by:

$$C_j = \left\{ (X,\sigma) \; ; A_j \text{ is chosen} \right\} \tag{16}$$

With the aid of (16) we define the following conditional expectation

$$E_{c_j}\left\{ \, . \, \right\} = E\left\{ \, . \, / \, (X,\sigma) \varepsilon C_j \right\} \tag{17}$$

Then, it can be shown that the centroid, or the optimal j^{th} codeword is given by:

$$A_j = \frac{E_{c_j}\left\{ \sigma X \right\}}{E_{c_j}\left\{ \sigma^2 \right\}} \tag{18}$$

Note that unlike [4] we do not impose the constraint that the shape vector has unit norm. Rather, an optimal normalization process determines the shape norms, allowing improved performance.

Expressions (15) and (16), which state necessary conditions for optimality, are used in the LBG design procedure to get the optimal shape codebooks of the second and third levels.

The codebook design is, usually, based on a training set which is assumed to adequately represent the random process X. The conditional expectations are estimated from the training set by:

$$E_{C_j}\left\{\sigma X\right\} = \frac{1}{|C_j|}\sum_{(X,\sigma)\epsilon C_j} \sigma X \tag{19}$$

and

$$E_{C_j}\left\{\sigma^2\right\} = \frac{1}{|C_j|}\sum_{(X,\sigma)\epsilon C_j} \sigma^2 \tag{20}$$

where $|C_j|$ is the number of training vectors in C_j.

To avoid excessive coding complexity, in cases where the required shape codebbook is too large, a suboptimal low complexity VQ method can be incorporated. In particular, it is convenient, in this case, to use the multi-stage VQ approach [5]. It can be shown that equations (10) through (12) still hold, so, the allocation algorithm does not change.

The training set used in the codebook design is a large set of *augmented vectors*. This means that a set of quantized gains has to be prepared before designing the second and the third codebooks The overall system design is therefore conducted in the following sequence:

(1) Given a training set of K-dimensional supervectors, prepare an M-dimensional training set of vectors of norms by calculating the norm of the corresponding subvectors.

(2) Design the first level codebook using the above training set and store the resulting training set of quantized M-dimensional norm vectors.

(3) Prepare a training set of L-dimensional norm vectors by splitting each subvector into L cells and calculating the L subnorms.

(4) Design the second level codebook using the training set of step (3) and the quantized norms (gains) of step (2). Store the resulting training set of quantized subnorm vectors.

(5) Prepare a training set of cells by splitting the subvectors into L k-dimensional cells.

(6) Design the third level codebooks using the training set of step (5) and the quantized gains of step (4). Note that q such codebooks have to be designed, one for each of the $B_1, B_2, .., B_q$ codebook sizes.

5. VARIABLE DIMENSION VQ

The HVQ scheme allows an easy way of implementing variable dimension vector quantization (VDVQ). This is because the major difficulty in VDVQ, maintaining a fixed bit rate, is readily solved here through the partitioning and codeword allocation processes. The parameters L,M and k are kept fixed to allows for a maximum main dimension of $K_{\max}=kLM$. If, however, the actual dimension K is less then $K_{\max}$, the missing data points are simply assigned zero values. Unlike the usual VQ schemes, bits are not wasted on coding zero valued components since they define cells with zero norm and, by means of the feature vectors, are assigned zero number of bits and are not transmitted. An additional small amount of side-information is transmitted to inform the receiver of the actual segment size K. To keep the bit rate fixed, the allocation algorithm is adjusted to the new dimension by changing only the parameter R.

VDVQ is important in the context of speech waveform coding. It enables an efficient waveform segmentation into the natural variable-length speech units.

6. SIMULATION RESULTS

To demonstrate the operation of the system, we constructed a special training set, with highly correlated vectors of norms. This was done by segmenting a speech waveform into variable size "steady state" segments and transforming each segment to the frequency domain using the discrete cosine transform. Since the smoothed short-term power spectrum of a speech segment belongs to a very limited family of possible shapes, the resulting vectors are highly correlated.

The speech material was composed of 8 male and 3 female voices. The full speech segment was 75 sec and was sampled at a rate of 8000 s/sec.

In the simulations, the parameters M,L and k were 16, 12 and 4 respectively. The segment length varied in the range from 256 to 768 data points. Three-stage codeooks [5] were used to reduce the coding complexity. About 30% of the total number of bits was allocated to the side information. The system was tested with two different bit rates: 2.0 and 1.2 bit/sample.

Preliminary simulations without any system optimization gave the following results in terms of MSE signal to quantization noise ratio: 20db for 16.0 kb/s and 15db for 9.6 kb/s Although the SNR is not necessarily indicative of speech perceptual quality, it enables quick comparison with other coding systems. The use of noise shaping allows an improved perceptual quality that is not indicated by the SNR values. Informal listening tests indicate that good communication quality is achieved at 9.6 kb/s and very nearly toll quality is achieved at 16 kb/s.

References [6]-[16] describe several representative speech waveform coding systems for which the SNR is reported. Specifically, [12]-[16] refer to VQ based systems. Based on the above preliminary results, the HVQ scheme outperforms all of these systems and, thus, appears to be very promising.

REFERENCES

[1] A. Segall, "Bit Allocation and Encoding for Vector Sources", IEEE Tr. on Inf. Th. IT-22 No. 2 pp.162-169 Mar 76.

[2] B. Fox, "Discrete Optimization via Marginal Analysis", Management Sci. Vol. 13, pp. 210-216, Nov 66.

[3] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Tr. on Comm. COM-28 pp. 84-95 Jan 80.

[4] M.J. Sabin and R.M. Gray, "Product Code Vector Quantization for Speech Waveform Coding", Conf. Rec. Globecom 82, pp. 1087-91, Dec 82.

[5] B.H. Juang, A.H. Gray Jr., "Multiple Stage Vector Quantization for Speech Coding", Proc. Int. Conf. ASSP, pp. 597-600, Paris, May 82.

[6] R.V. Cox, R.E. Crochiere, "Real Time Simulation of Adaptive Transform Coding", IEEE Tr. on Acoust., Sp, Sig.Proc. Vol. ASSP-29, No. 2, Apr 81.

[7] N.S. Jayant, "Pitch Adaptive DPCM Coding of Speech With Two Bit Quantization and Fixed Spectrum Prediction", BSTJ Journal, pp. 440-454, March 1977.

[8] C.S. Xydeas, C.C Evci, R. Steele, "Sequential Adaptive Predictors for ADPCM Speech Encoders", IEEE Tr. on Comm. Vol. COM-30, No. 8, pp. 1942-54, Aug 82.

[9] R.E. Crochiere, S.A. Webber, J.L. Flanagan, "Digital Coding of Speech in Sub-bands", BSTJ, Vol. 55, pp. 1069-85, Oct 76.

[10] L.C. Stewart, R.M. Gray, Y. Linde, "The Design of Trellis Waveform Coders", IEEE Tr. on Comm. Vol. COM-30, pp. 702-710, Apr 82

[11] R. Zelinski, P. Noll, "Adaptive Transform Coding of Speech Signals" IEEE Tr. on Acous. Spch. Sig. Proc. Vol. ASSP-25, No. 4, Aug 77.

[12] R.M. Gray, H. Abut, "Full Search and Tree Search Vector Quantization of Speech Waveforms", ICASSP-82 Proc., pp. 593-96, Paris, May 82.

[13] H. Abut, R.M. Gray, "Vector Quantization of Speech and Speech-Like Waveforms", IEEE Tr. on ASSP, Vol. ASSP-30, pp. 423-436, Jun 82.

[14] V.M. Cuperman and A. Gersho, "Adaptive Differential Vector Coding of Speech," Conf. Rec., IEEE Global Communications Conf., Miami, Dec. 1982, pp. 1092-1096.

[15] A. Gersho and V. Cuperman, "Vector Quantization: A Pattern-Matching Approach to Speech Coding," IEEE Communications Magazine, to appear, December 1983.

[16] R. M. Gray, "Vector Quantization," IEEE ASSP Magazine, to appear, April 1984.