FAST AND LOW-COMPLEXITY LSF QUANTIZATION USING ALGEBRAIC VECTOR QUANTIZER

Minjie Xie and Jean-Pierre Adoul

Electrical and Computer Engineering Department
University of Sherbrooke
Sherbrooke, Quebec, Canada J1K 2R1

ABSTRACT

This paper presents an algebraic vector-quantization scheme for encoding the LSF parameters used in describing the time-varying short-term spectrum of speech in many modern vocoders. The quantizer achieves an average spectral distortion of 1 dB at 28 bits/frame for the telephone bandwidth. The scheme is based on low-dimensionality regular-point lattices. Properties of lattices are taken advantage of in both the design and the search of the quantizer codebook. Namely, this algebraic codebook need not be stored in memory and the optimum vector is found through simple rounding of the input variables instead of the usual exhaustive search. Thus, the scheme results in significant savings of memory and reduced computational complexity when compared to traditional vector-quantizer solutions.

1. INTRODUCTION

The linear predictive coding (LPC) method is one of the most popular approaches for describing the time-varying short-term spectrum of the speech signal. In many speech coding systems, LPC coefficients are transformed to the line spectrum frequency (LSF) parameters which are a very effective representation for quantization of the LPC information [1][2]. The LSF's are related to the poles of the LPC filter (or the zeros of the inverse filter) in the z-plane. For a 10th order LPC analysis, the z-transform of the LPC inverse filter is denoted by

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_{10} z^{-10} . \quad (1)$$

From (1), two new polynomials are defined :

$$P(z) = A(z) + z^{-11} A(z^{-1}) \quad (2)$$

and

$$Q(z) = A(z) - z^{-11} A(z^{-1}) . \quad (3)$$

The roots of these polynomials are usually called the line spectrum pairs or the line spectral frequencies (LSF's). Some important properties are described in detail in [1] - [4].

Recently, some vector-quantization schemes of LSF parameters have been developed and "transparent quality" quantization [3], defined by a 1 dB spectral distortion, was achieved by these schemes [3] - [6]. A drawback of these techniques is the large amount of memory required to store the codebook and the high complexity of computation used in comparing the input vector to each codevector. The problem can usually be resolved at the cost of reduced performance of the quantizer.

An algebraic vector-quantization algorithm which is based on regular-point lattices is proposed in this paper. From a geometric standpoint, a lattice is a regular arrangement of points in n-dimensional Euclidean space $\boldsymbol{R}^n$. From an algebraic standpoint, an n-dimensional lattice is a collection of vectors which form a group under ordinary vector addition in $\boldsymbol{R}^n$. The simplest n-dimensional lattice is the integer lattice $\boldsymbol{Z}^n$ which consists of all vectors with integer coordinates [7].

In this work, low-dimensionality regular-point lattices are used to design vector quantizers that can be used in real-time speech-coding algorithms. Some of the properties of lattices are taken advantage of for both efficient designing and efficient searching of the quantizer codebook. Firstly, the codebook is not stored in memory. The codevectors are a subset of the points of an integer lattice. They are indexed by an algebraic method. Secondly, the search for the nearest neighbor in the codebook to some input vector can be done very efficiently because of the lattice regular structure.

2. ALGEBRAIC ALGORITHM FOR VECTOR QUANTIZATION OF LSF PARAMETERS

In this section, we describe the algebraic vector quantization of LSF parameters. The speech database used in the design is based on speech spoken in seven languages. It consists of 93,500 LSF parameter vectors resulting from a 10th order LPC analysis performed every 24 ms speech frame.

The ten LSF parameters are ordered points on the [0, π] interval (Fig. 1). Four independent sub-quantizers are used: one absolute vector quantizer (VQ) in two dimensions, one relative differential VQ in two dimensions and finally two relative differential VQs in three dimensions. The absolute quantizer is used for quantizing jointly the 3rd and 7th LSF parameters. This is the only non-algebraic vector quantizer (NAVQ) with a stored codebook of size 64 which is designed by using the LBG or K-means algorithm [8]. Let us call, $\hat{\omega}_3$ and $\hat{\omega}_7$, the corresponding transmitted values.

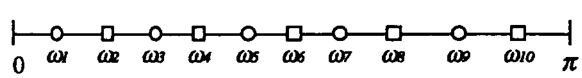

Fig.1. LSF parameters in the angular frequency domain.

0-7803-2431-5/95 $4.00 © 1995 IEEE

The eight remaining LSF parameters are divided into three groups. The pair ω_1 and ω_2, the triplet ω_4, ω_5 and ω_6 and the triplet ω_8, ω_9 and ω_{10}. Each group is quantized using one of the three above-mentioned differential vector quantizers. The three vectors to quantize, $x_1=(x_{11}, x_{12})$, $x_2=(x_{21}, x_{22}, x_{23})$ and $x_3=(x_{31}, x_{32}, x_{33})$, are defined as follows

$$\begin{cases} x_{11} = \dfrac{\omega_1}{\hat{\omega}_3} \\ x_{12} = \dfrac{\hat{\omega}_3-\omega_2}{\hat{\omega}_3} \end{cases}, \tag{4}$$

$$\begin{cases} x_{21} = \dfrac{\omega_4-\hat{\omega}_3}{\hat{\omega}_7-\hat{\omega}_3} \\ x_{22} = \dfrac{\omega_5-\hat{\omega}_4}{\hat{\omega}_7-\hat{\omega}_3} \quad ; \textit{ where } \hat{\omega}_4 = \hat{x}_{21}(\hat{\omega}_7-\hat{\omega}_3)+\hat{\omega}_3 \\ x_{23} = \dfrac{\hat{\omega}_7-\omega_6}{\hat{\omega}_7-\hat{\omega}_3} \end{cases} \tag{5}$$

and

$$\begin{cases} x_{31} = \dfrac{\omega_8-\hat{\omega}_7}{\pi-\hat{\omega}_7} \\ x_{32} = \dfrac{\omega_9-\hat{\omega}_8}{\pi-\hat{\omega}_7} \quad ; \textit{ where } \hat{\omega}_8 = \hat{x}_{31}(\pi-\hat{\omega}_7)+\hat{\omega}_7 \; . \\ x_{33} = \dfrac{\pi-\omega_{10}}{\pi-\hat{\omega}_7} \end{cases} \tag{6}$$

The components of these vectors represent LSF differences normalized in such manner that the sum of the components add to one. The statistical joint distribution of the two components of x_1 is confined to a rectangular isosceles triangle (Fig. 2). For this reason we shall call the quantizer a *"triangular VQ"*. Similarly, the joint distributions of components of the three-dimensional vectors x_2 and x_3 are confined to the shape of a simplex (i.e.: a pyramidal region) (Fig. 3) and consequently we shall call the corresponding quantizer a *"simplex VQ"*. An approach is now presented for designing both the triangular vector quantizer and the simplex vector quantizers based on lattices.

The triangular vector quantizer (TVQ) codebook consists of points of a regular-point lattice which form a rectangular isosceles triangle in the plane. The sum of the components of each codevector is a positive even (/ odd) integer no greater than some fixed quantity, N. For instance, a codebook of size 64 is obtained in this fashion by considering the set of integer-component vectors with an even sum no greater than $N = 14$. This codebook is shown in Fig. 4.

The TVQ codebook is not stored in memory. The codevectors are indexed according to an algebraic rule which we shall soon discuss. The index of the selected codeword is transmitted through the channel. At the receiver, the codevector is found from the received index. In order to use this codebook with integer-component codewords we need to scale the input vector x_1.

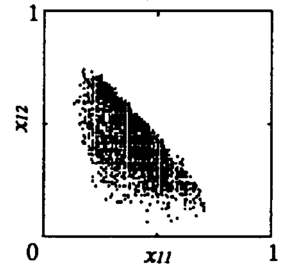

Fig.2. Statistical joint distribution of components of x_1.

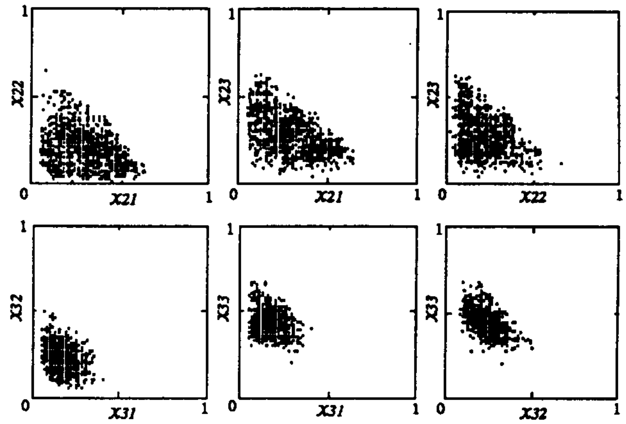

Fig.3. Statistical joint distributions of components of x_2 and x_3.

For the example of Fig. 4, the codevectors are indexed from 0 to 63. The TVQ codebook is split into several *"levels"* according to the sum of the codevector components. It can be seen from the figure that levels 0, 1, 2, ..., 7 contain 1, 3, 5, ..., 15 codevectors, respectively. Evidently, the number of codevectors constitutes an odd arithmetic series. The index k_0 of the first codevector at a certain level is set equal to the number of codevectors lying at lower levels. Hence, the index of a any codeword at this level is obtained as the sum of the "offset", k_0, and the codeword rank within the level. It is easy to verify that the nth partial sum of this arithmetic series is $1 + 3 + 5 + 7 + \cdots + (2n - 1) = n^2$. Therefore, the index, k, of a codevector $c = (c_1, c_2)$ is given by

$$k = k_0 + c_1 = \left(\frac{c_1+c_2}{2}\right)^2 + c_1 \; . \tag{7}$$

In Fig. 4 the codevectors have an even component sums. It is also possible to consider codevectors with odd component sums. In this case the indexing scheme is slightly different. The numbers of codevectors at successive levels form an even arithmetic series so that the index k of any codevector in this case is given by

$$k = k_0 + c_1 = \frac{(c_1+c_2)^2-1}{4} + c_1 \; . \tag{8}$$

At the receiver, we first extract k_0 from the received index k and then deduce c_1 and c_2.

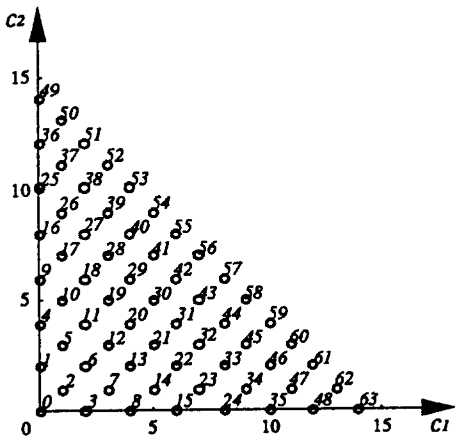

Fig.4. TVQ codebook of size 64 codewords with even component sums of no greater than $N = 14$.

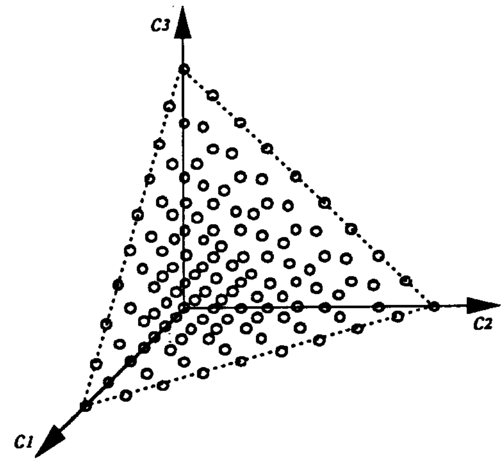

Fig.5. SVQ codebook of size 125 codewords with odd component sums of no greater than $N = 9$.

Let us now discuss a fast procedure to find the closest codevector, in the MSE sense, to an arbitrary vector z. The following transform matrix,

$$T = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \tag{9}$$

is used to rotate the vector z through a counterclockwise angle of $\frac{\pi}{4}$ radians. We then round off each component of the rotated vector to its nearest even integer to find the closest "rotated codevector". By inverse transform we readily find the closest codeword meeting the even parity condition, namely, that the sum of its integer components be even (/ odd). Finding the closest codeword can be done in an alternate way described in [9] by recognizing that the problem amounts to finding the nearest neighbor in the so-called D_2 lattice.

Just as in the TVQ case, the simplex vector quantizer (SVQ) codebook is also a finite region of the regular-point lattice in three dimensions and is not stored in memory. For example, Fig. 5 shows a codebook of size 125 with an odd sum of components no greater than $N = 9$.

The encoding and decoding of SVQ are similar, in many ways, to the procedure used in the TVQ case. Again we need to scale the three-dimensional input vector to match to the simplex codebook. For the example illustrated in Fig. 5, the codebook comprises five levels which have 3, 10, 21, 36 and 55 codevectors, respectively. The series corresponding to the number of codevectors is $1 \cdot 3 + 2 \cdot 5 + \cdots + n(2n+1)$ and the nth partial sum of this series is given by $1 \cdot 3 + 2 \cdot 5 + \cdots + n(2n+1) = \frac{1}{6}n(n+1)(4n+5)$.

Now let $c = (c_1, c_2, c_3)$ be any codevector and $s = c_1 + c_2 + c_3$ the sum of the components of c. The index k_0 of the first codevector at this level can be expressed as

$$k_0 = \frac{1}{6}m(m+1)(4m+5) \tag{10}$$

where $m = \frac{s-1}{2}$. Thus, the index k of the vector c is given by

$$k = k_0 + \frac{(s-c_1)(s-c_1+1)}{2} + c_2 . \tag{11}$$

For the case of codebooks designed with vectors having an even sum of components, we have

$$k_0 = \frac{1}{6}m(m+1)(4m-1) \tag{12}$$

where $m = \frac{s}{2}$, and the index k is also given by the expression (11).

The codebook search in SVQ is somewhat more complicated than in TVQ. We see from definitions (4) through (6) that, whereas the components of the two-dimensional vector x_1 are independent, and the first two components of the three-dimensional vector x_2 (or x_3) are correlated in the sense that the difference, x_{22} (or x_{32}), depends on the quantized value $\hat{x}_{21}$ (or $\hat{x}_{31}$). Recall that a codevector has integer components such that the component sum is even. This three-term-parity condition entails that either (only) two components are odd or none are. The component, x_{21}, can be rounded to either the nearest even integer or the nearest odd integer. We will consider both cases and retain the best choice from an overall-MSE-performance standpoint. First, we consider the nearest even integer of x_{21} and compute the corresponding $\hat{x}_{21}$. We then jointly quantize the pair, x_{22}, x_{23}, by rounding both variables to integers values such that the three-term-parity condition is met. Note that in this first case the last operation amounts to finding the nearest neighbor in the D_2 lattice [9]. Second, we consider the nearest odd integer of x_{21}, and repeat the same steps. Finally, the closest codevector is found by comparing the squared error distortions between the original vector and each of the reproduction vectors resulting from the two cases.

In order to obtain a better quantization performance, a non uniform bit allocation scheme is devised for any given bit rate. Since the LSF parameter numbers corresponding to each differential vector are different, the two-dimensional vector x_1 is quantized with less bits and the three-dimensional vectors x_2 and x_3 with more bits. However, the non-algebraic vector quantizer always has 64 codevectors (6 bits) in its stored codebook. Table I shows the bit allocation scheme for different bit rates (in the range 26-29 bits/frame).

3. QUANTIZATION PERFORMANCE

The large database mentioned earlier was used in evaluating the performances. The distortion measured is the spectral distortion (SD) between the original and quantized LSF parameters. The LSF parameter quantization performance of the algebraic vector quantizer is shown in Table II for different bit rates. It appears from this table that the 28 bits/frame algebraic vector quantizer can achieve an average spectral distortion of about 1 dB, less than 2% outliers in the range 2-4 dB, and no outlier having spectral distortion greater than 4 dB. Fig. 6 shows the histogram of the spectral distortion for this result.

4. CONCLUSION

In this paper, an algebraic algorithm for vector quantization of the LSF parameters of speech has been presented. Both the triangular vector quantizer (TVQ) and the simplex vector quantizer (SVQ) based on regular-point lattice have been designed to jointly quantize vectors LSF differences, two or three such differences at a time. The 28 bits/frame algebraic vector quantizer can achieve the commonly accepted conditions for transparent quality quantization of LPC information: an average spectral distortion of about 1 dB, less than 2% outlier frames having spectral distortion in the range 2-4 dB, and no outlier frame with spectral distortion greater than 4 dB. When contrasted with traditional VQ solutions, the algebraic-vector-quantizer approach just described results in significant savings in terms of memory requirement and the codebook search is reduced to a few rounding operations.

TABLE I
BIT ALLOCATION FOR DIFFERENT BIT RATES

Rate (bits/frame)	Bit Allocation (bits)			
	NAVQ	TVQ	SVQ1	SVQ2
26	6	5	8	7
27	6	5	9	7
28	6	5	9	8
29	6	6	9	8

TABLE II
SPECTRAL DISTORTION (SD) PERFORMANCE OF THE ALGEBRAIC VECTOR QUANTIZER

Rate (bits/frame)	SD (dB)	Outliers (%)	
		2-4 dB	>4 dB
26	1.17	3.06	0
27	1.10	2.18	0
28	1.04	1.80	0
29	0.99	1.52	0

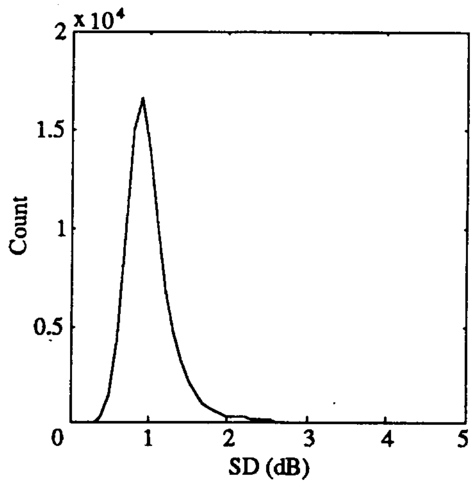

Fig.6. Histogram of the spectral distortion (SD) for the 28 bits/frame algebraic vector quantizer.

ACKNOWLEDGMENT

The authors wish to thank Dr. Redwan Salami for his assistance during the course of this work.

REFERENCES

[1] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals", *J. Acoust. Soc. Amer.*, vol. 57, p. S35(A), Apr. 1975.
[2] F. K. Soong and B. H. Juang, "Line spectrum pair (LSP) and speech data compression", in *Proc. ICASSP-84*, pp. 1.10.1-1.10.4, Mar. 1984.
[3] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame", *IEEE Trans. Speech and Audio Processing*, vol. SAP-1, No. 1, pp. 3-14, Jan. 1993.
[4] F. K. Soong and B. H. Juang, "Optimal quantization of LSP parameters", *IEEE Trans. Speech and Audio Processing*, vol. SAP-1, No. 1, pp. 15-24, Jan. 1993.
[5] R. Laroia, N. Phamdo, and N. Farvardin, "Robust and efficient quantization of speech LSP parameters using structured vector quantizers", in *Proc. ICASSP-91*, pp. 641-644, May 1991.
[6] M. Yong and V. Eyuboglu, "Low complexity LPC quantization using the structured vector quantizer", in *Proc. IEEE Workshop on Speech Coding for Telecommunications*, pp. 59-60, Oct. 1993.
[7] J. H. Conway and N. J. A. Sloane, "Sphere Packings, Lattices and Groups", Springer-Verlag, New York, 1988.
[8] Y. Linde, A. Buzo, and R. M. Gary, "An algorithm for vector quantizer design", *IEEE Trans. Commun.*, vol. COM-28, No. 1, pp. 84-95, Jan. 1980.
[9] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes", *IEEE Trans. Inform. Theory*, vol. IT-28, No.2, pp. 227-232, Mar. 1982.