2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

10.21437/ICSLP.1992-11

VECTOR QUANTIZATION OF SPEECH LSF PARAMETERS WITH GENERALIZED PRODUCT CODES

Erdal Paksoy†, Wai-Yip Chan††, and Allen Gersho†

†Center for Information Processing Research
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
erdal%kane@ece.ucsb.edu, gersho@ece.ucsb.edu

†† Department of Electrical Engineering
McGill University, 3480 University Street
Montreal, Quebec, Canada H3A 2A7
chan@pike.ee.mcgill.ca

ABSTRACT

Generalized product code (GPC) vector quantization (VQ) is applied to the coding of speech linear prediction filter parameters. We show that the performance of conventional product code VQ can be improved through the use of conditional feature codebooks and multiple-survivor encoding search. Two particular product code structures, split VQ (SVQ) and multistage VQ (MSVQ), are explored within the GPC framework for the quantization of prediction filter parameters in the line spectral frequency (LSF) domain. Our experiments show that with a suitable MSVQ scheme, 21 bits/frame suffice to encode the LSF parameters to furnish transparent-coding quality. Transparent coding with SVQ requires 22 bits/frame, but only half of the encoding complexity of the best MSVQ scheme.

1. INTRODUCTION

Speech synthesis filters based on linear prediction modeling have become an essential feature of the majority of low bit rate speech coders currently being studied, under development, or in commercial use. Examples of such coders are multipulse LPC (MP-LPC), code-excited LPC (CELP), vector excitation coding (VXC), regular pulse excited LPC (RPE-LPC), low delay CELP (LD-CELP), and vector sum excited linear prediction (VSELP). In such coders, the accurate reproduction of speech formants and their time evolution depends largely on how accurately the LPC parameter set is quantized for each analysis frame. This task becomes increasingly difficult as current efforts in speech coding strive for lower bit rates (i.e., 4 kb/s or lower) while maintaining high quality. Consequently, there is a great need to find extremely efficient ways to quantize LPC parameters with a minimum number of bits while maintaining a suitable measure of accuracy in the spectral representation specified by these parameters. While it is known that vector quantization (VQ) theoretically offers the optimal way to quantize a vector of parameters, complexity limitations generally require that ideal VQ (with no structural constraints) be replaced by suboptimal methods. This opens the door to a large variety of alternatives whose relative merits can usually be assessed only by experimental results. Here we consider generalized product codes (GPCs) a particularly attractive and powerful approach to structurally constrained VQ of LPC parameters. The results demonstrate that "transparent" quality LPC quantization is achievable with as few as 21 bits.

It is well-known that the quantization of LPC parameters can be performed with higher efficiency if the prediction coefficients are first transformed into a different representation such as log area ratios, arcsine reflection coefficients, or line spectral frequencies (LSFs). Of these, the LSFs appear to present some advantages for vector quantization over the other representations [1]. Therefore, we used the LSF representation in our experiments.

Various scalar quantization and vector quantization methods have been used in the past for coding the LPC spectral information. The scalar quantization methods benefit from low computational complexity, and perhaps also some extra degree of robustness, but require more bits than VQ methods. For instance, scalar quantization with no interframe coding consumes 34 bits per frame in the U.S. Federal Standard 4800 b/s speech coder [2]. Instead, with VQ, substantial savings in bits are possible. However, the quantization resolution necessary for transparent coding of an LPC vector whose dimension is in the range of 8 to 12 (8 kHz sampling) can not be viably furnished with unstructured VQ; the encoding, storage, and design complexity would be prohibitive. To circumvent the complexity problem, structured VQ, more specifically product code VQ, can be used. In this work we report our experiments for coding of the LSF parameter set based on two VQ structures, split vector quantization (SVQ) and multistage vector quantization (MSVQ), which we examine next in the context of a novel structured VQ framework. It should be noted that other recent studies on VQ of LPC parameters have also used structural constraints and have made important advances

This work was supported in part by the National Science Foundation, Fujitsu Labs, the UC Micro program, Rockwell International, Hughes Aircraft Company, Eastman Kodak, and the Natural Sciences and Engineering Research Council of Canada.

in performance, e.g., [1] , [3] , [4].

In this paper, we first review the concept of a product code and its recent generalization to a GPC. We then consider measures of distortion and the notion of *transparent* quality in coding LPC parameters. With this background, we then describe our methods, experiments, and results in quantization of LSF parameter vectors.

2. GENERALIZED PRODUCT CODES

In unstructured VQ, a source is represented with only a *single* codebook. To obtain best performance for a given codebook, a source vector $\mathbf{x}$ is encoded by finding a code vector in the codebook to minimize a distortion criterion; the distortion-minimizing code vector is emitted by the decoder as a reproduction $\hat{\mathbf{x}}$ of $\mathbf{x}$. In product code VQ, a source is represented by $s > 1$ codebooks, one for each of s *features* $\mathbf{f}_i$, for $i = 1, \cdots, s$, associated with the source. For a given source vector $\mathbf{x}$, the encoder selects one feature vector, $\hat{\mathbf{f}}_i$, for $i = 1, \cdots, s$, from each codebook. A *synthesis function* $\mathbf{g}$ maps the Cartesian product of these selected features into a reproduction $\hat{\mathbf{x}}$ in the source vector space. Thus, $\hat{\mathbf{x}} = \mathbf{g}(\hat{\mathbf{f}}_1, \cdots, \hat{\mathbf{f}}_s)$.

In a *sequential search product code* [5], the features are encoded sequentially in s stages, from $\mathbf{f}_1$ to $\mathbf{f}_s$. In stage i, feature $\mathbf{f}_i$ is extracted from $\mathbf{x}$ via a *feature extraction rule* which may be dependent on the features quantized earlier in the sequence, i.e., $\mathbf{f}_i = \mathbf{h}_i(\mathbf{x}, \hat{\mathbf{f}}_1, \cdots, \hat{\mathbf{f}}_{i-1})$. The quantization of feature $\mathbf{f}_i$ in stage i minimizes a *feature distortion measure* $d_i(\mathbf{f}_i, \mathbf{c}_{i,n})$ between $\mathbf{f}_i$ and all code vectors $\mathbf{c}_{i,n}$ in the single codebook of feature $\mathbf{f}_i$.

A *generalized product code* (GPC) [6] is a sequential search product code whose features $\mathbf{f}_i$, for $i > 1$, may each have $M_i \geq 1$ codebooks. M_i is called the *codebook fanout* for the i-th feature and M_1 is always unity. Associated with $\mathbf{c}_{ij,n}$, the nth code vector in the j-th codebook C_{ij} of feature $\mathbf{f}_i$, $i = 1,\ldots, s-1$, is a *codebook pointer* $\mu_{ij,n}$ whose value identifies one of the M_{i+1} codebooks for feature $\mathbf{f}_{i+1}$. Thus, if feature $\mathbf{f}_i$ is quantized to code vector $\mathbf{c}_{ij,n}$, then the codebook addressed by the pointer $\mu_{ij,n}$ of that code vector is used for the quantization of the next feature $\mathbf{f}_{i+1}$.

The GPC structure can be graphically visualized as a trellis diagram, where the nodes are codebooks, each vertical array of nodes corresponds to a particular feature, and the directed links are the pointers that lead to the next-stage codebooks. This is illustrated in Fig. 1, for the case of a two-stage GPC with a fanout of two.

The feature extraction rules and feature distortion measures enable the employment of dynamic-programming tree search algorithms (e.g., the M,L algorithm) to organize the search into making delayed decisions by keeping alive multiple evolving paths, i.e. the *survivors* [6]. Best performance is obtained for a given GPC when no path on the trellis is pruned until the last feature, and this corresponds to maximum encoding complexity. Multiple-survivor search for LPC parameter coding was also reported in [3].

The feature codebooks in a GPC can be designed jointly with the codebook pointers one stage at a time using the *constrained storage VQ* (CSVQ) algorithm [7]. In GPC design, the codebook fanout is treated as a storage complexity parameter, and the number of survivors kept

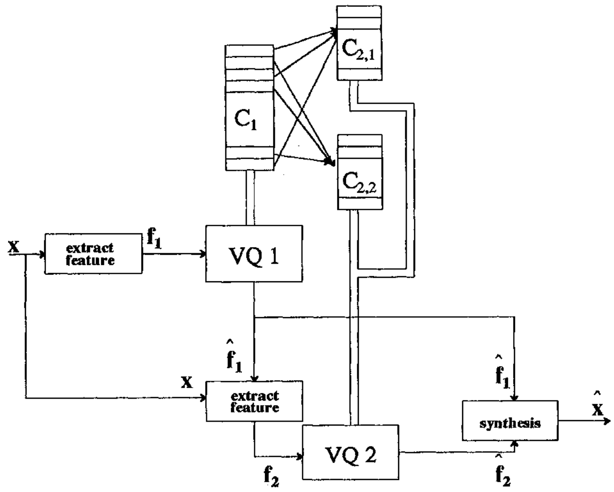

Fig.1 : Two-stage GPC VQ of LSFs with fanout of two.

in the delayed decision search is treated as an encoding complexity parameter. In general, the larger is either parameter, the better is the performance of the quantization scheme. Moreover, the parameters can be selected independently of each other.

The structured VQ we employ in this work belongs to the the family of GPCs called *summation product codes*, defined by the synthesis function $\mathbf{x} = \mathbf{f}_1 + \cdots + \mathbf{f}_s$. The feature vectors in this family are residual vectors, obtained by subtracting from the source vector the sum of the quantized features from all previous stages, $\mathbf{f}_i = \mathbf{x} - (\hat{\mathbf{f}}_1 + \cdots + \hat{\mathbf{f}}_{i-1})$. Both the conventional multistage VQ (MSVQ) [8] and tree structured VQ (TSVQ) [9] are in this family [5]. The difference between them is that MSVQ has unity fanout for each feature whereas TSVQ has maximum possible fanout for each feature. Split VQ (SVQ) is a restricted form of MSVQ where the feature vectors of SVQ have some components constrained to be zero in such a way that the summation in the synthesis function can be replaced by a concatenation and the feature extraction rule reduces to a partitioning of the source vector. Thus MSVQ is more general than SVQ in the sense that every SVQ can be realized as an MSVQ but the converse is not true. This generality implies that for a given assignment of bits, codebook fanout, and distortion measures to the features, MSVQ can outperform SVQ though at a cost of higher complexity. From the point of view of performance-complexity tradeoff, however, MSVQ is not necessarily superior to SVQ.

3. DISTORTION MEASURES FOR DESIGN AND ENCODING

We generally aim for *transparent* quantization of the LPC parameters so that there is no audible difference between coded speech signals synthesized using quantized and unquantized LPC coefficients. A widely accepted criterion for measuring the accuracy of LPC quantization is the Log Spectral Distortion (SD) measure given by (see, for example, [1].):

$$SD^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} [\ 10 \log S(\omega) - 10 \log \hat{S}(\omega)\ ]^2 d\omega$$

For transparent quantization, it is considered to be sufficient to quantize the LPC coefficients with an average

SD of up to approximately 1 dB, while holding below 2% the percentage of outlier frames having an SD value between 2 dB and 4 dB. No outlier frames should have an SD exceeding 4 dB.

Our main performance objective to be minimized is the average SD between quantized and unquantized LPC parameter vectors. However, due to the complex dependence between LSFs and the spectral envelope and the complexity of the SD computation, there is no easy way of incorporating the SD into the codebook design algorithm. For this reason a weighted mean squared error (WMSE) distortion measure is used to design the codebooks. Such a distance measure gives more weight to perceptually significant vector components. Various methods for calculating weights have been proposed by several authors [1] , [4] , [10]. In our experiments, we tested the performance of all these weighting functions and observed that they all result in similar quantization performance; hence we chose to compute the weights according the formulas in [1].

While SD is difficult to use in codebook design, it can be used in encoding. However, because of the computational cost of calculating SD, it is not realistic to perform the entire codebook search using that distortion function. On the other hand when a multiple survivor method is employed, SD can be used to select the best candidate from the final stage survivor set [3]. This was done in the case of SVQ and MSVQ, where L_i survivors were kept for each feature and the SD was used to select the best quantized output from the last survivor set.

One of the major advantages of the LSF representation is that as long as the ascending order of the LSF vector components is preserved in the quantization process, the synthesis filter is guaranteed to be stable. When codebooks are designed using the generalized Lloyd algorithm (modified for the WMSE distortion measure), the centroids consistent with the measure do not guarantee the conservation of the ascending order. Instead of the usual centroid computation, we therefore obtain the centroid for each quantization region by simply averaging all training vectors in that region, as would be done for the MSE criterion, and thereby stability is guaranteed. A slightly better approach is to use a centroid consistent with WMSE, check the ordering of the centroid components, and replace the centroid by one consistent with MSE if the ordering is incorrect. In practice, it was observed that both these methods yield virtually identical performance. A stability check is also performed in the encoding process, to ensure that the components of the reconstructed LSP vector are in ascending order.

4. LPC ANALYSIS

In order to obtain a training set of LPC vectors, LPC analysis was performed on a large database of approximately 24.5 minutes of speech signals that have been lowpass filtered at 3.4 kHz and sampled at 8 kHz. Frames with energy below a threshold were classified as "silence" and rejected from the training set. We performed 10th order LPC analysis using the modified covariance method with high frequency compensation. The analysis window size was 20 ms, and consecutive analysis frames were chosen to overlap by 10 ms, thereby doubling the LPC vector training set size from 72400 to 144800 vectors. Bandwidth expansion was also used, i.e., we multiplied the ith prediction coefficient by γ^i, where i = 1,..,10 and γ is a constant equal to 0.996. The performance of the trained quantizers was evaluated using a test set comprising 7700 frames (2.5 minutes) and generated independently from the training set.

5. RESULTS

We quantized the speech LSF parameter vector using MSVQ and SVQ. In the SVQ experiments, the first subvector (feature) consists of the first four LSFs and the second subvector contains the remaining six. We chose this configuration because it outperformed an even 5-5 split for all the bit allocations we have attempted for the two subvectors. We varied the fanout and the number of survivors for both MSVQ and SVQ, in order to explore the potential of GPC VQs. For each coding rate, the results exhibited below are for the best bit allocation configuration.

In the experiments with SVQ, we observed that for a given bit rate, the performance of the quantization method improved with the fanout and with the number of candidates retained in the multiple survivors search. Fig. 2 shows the variation of average SD as a function of the bit rate for SVQ designed using WMSE under different fanout and codebook search conditions. The advantage gained from using multiple codebooks essentially saturates after a fanout of four.

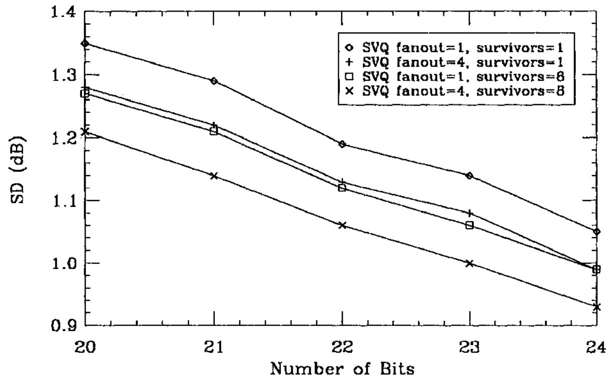

Fig. 2 : Performance of Split Vector Quantization of LSFs using Generalized Product Codes

The following conclusions can be drawn from Fig. 2. Using four second-stage codebooks for SVQ gives a savings of one bit/frame, i.e., we get the same average SD value for a fanout of 4 and 23 bits/frame as we get for unity fanout and one more bit in the second stage. It must be noted that the computational complexity of 23-bit SVQ with a fanout of four is less than that of the 24-bit SVQ proposed in [1] while the SD performance is the same; this is likely a worthwhile tradeoff between performance, encoding complexity, and storage complexity. Selecting the best candidate out of a set of 8 survivors also saves about one bit/frame. When a fanout greater than unity and multiple-survivor search is used, it is possible to transparently quantize LPC parameters using 22 bits/frame, saving 2 bits/frame over the results reported in [1]. The proportion of outlier frames having a spectral distortion value between 2 dB and 4 dB is also held under 2% for all quantizers having an average SD value below 1.15 dB. No frames with SD > 4 dB were found.

Table 1 shows the performance of MSVQ without fanout. The numbers in the second and third column correspond to the number of bits used in the first and

Table 1. Performance of MSVQ using WMSE.

Bit Allocation			L	Avg SD	Outliers
Total	f1	f2		(dB)	2-4 dB(%)
24	12	12	1	1.00	1.35
23	12	11	1	1.07	1.87
22	11	11	1	1.14	2.87
21	11	10	1	1.21	4.19
20	10	10	1	1.29	6.30
24	12	12	8	0.88	0.35
23	12	11	8	0.94	0.62
22	11	11	8	1.00	0.74
21	11	10	8	1.07	1.71
20	10	10	8	1.14	2.21

second stage codebooks respectively. The numbers in the fourth column, labelled L, represent the number of survivors kept before the final pruning of the survivor set using SD. We can see that MSVQ has an advantage of 1 bit/frame over SVQ. This gain is by virtue of the aforementioned generality of the MSVQ vis-a-vis SVQ. We also observe from Table 1 that using a multiple survivors search where the final decision is arbitrated by the SDs of 8 candidates leads to a savings of 2 bits per frame. Hence this method allows us to transparently quantize the LPC vectors at 21 bits/frame.

We found that increasing the fanout from 1 to 6 does not yield a tangible improvement in the SD performance of MSVQ; both the average SD values and the number of outliers remain the same. This can be explained by the fact that in the case of SVQ the second feature statistics benefit from the ordering property of the LSFs and hence are conveniently classified for quantization using the multiple codebooks. In the case of MSVQ, because of the relatively high resloution first stage quantization, the second feature statistics are more homogeneous compared to those of the second feature in SVQ and do not benefit as much from classification.

We also subjectively evaluated the quality of the various quantization schemes, by synthesizing speech from the original (not quantized) residual, using quantized LPC coefficients. In informal listening tests, we observed that the perceptual quality of the reconstructed speech correlates quite well with the SD values.

6. CONCLUSIONS

In this paper, we explored the vector quantization of speech LSF parameters by applying some elements of a generalized product code framework. In contrast with the prior observations of Paliwal and Atal [1] our results have shown that, for a given bit rate, MSVQ performs better than SVQ, even when SVQ has fanout but MSVQ does not. Nevertheless, SVQ is certa[illegible] because it has lower complexity and [illegible] the ordering property of the LSFs thr[illegible] multiple second feature codebooks. Also, [illegible] an effective vehicle to demonstrate the beneh[illegible] thereby confirming that under appropriate co[illegible]ons, GPC VQ can offer a valuable approach for improving structured VQ performance. Our experiments showed that with a suitable MSVQ scheme, it is possible to transparently quantize LSF parameters using 21 bits/frame. In contrast, SVQ furnishes transparent quality at 22 bits/frame while expending roughly half of the encoding complexity and two times the storage complexity of the best MSVQ scheme.

References

[1] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *Proc. IEEE Int. Conf. Acoust., Speech, Sign. Processing*, pp. 661-664, Toronto, Canada, May, 1991.

[2] J. P. Campbell, Jr., V. C. Welch, and T. Tremain, "An Expandable Error-Protected 4800 bps CELP Coder (U.S. Federal Standard 4800 bps Voice Coder)," *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Processing*, pp. 735-738, Glasgow, Scotland, May 1989.

[3] B. Bhattacharya, W. LeBlanc, S. Mahmoud, and V. Cuperman, "Tree Searched Multi-stage Vector Quantization of LPC Parameters For 4 Kb/s Speech Coding," *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Processing*, vol. 1, pp. 105-108, San Francisco, March, 1992.

[4] R. Laroia , N. Phamdo, and N. Farvardin, "Robust and Efficient Quantization of Speech LSF Parameters Using Structured Vector Quantizers," *Proc. IEEE Int. Conf. Acoust., Speech, Sign. Processing*, pp. 641-644, Toronto, Canada, May, 1991.

[5] W.-Y. Chan and A. Gersho, "Enhanced Multistage Vector Quantization with Constrained Storage," *Proc. 24th Asilomar Conf. Circuits, Systems, and Computers*, pp. 659-663, Pacific Grove, CA, November 1990.

[6] W.-Y. Chan, "The Design Of Generalized Product-Code Vector Quantizers," *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Processing*, vol. 3, pp. 389-392, San Francisco, March, 1992.

[7] W.-Y. Chan and A. Gersho, "Constrained-Storage Quantization of Multiple Vector Sources by Codebook Sharing," *IEEE Transactions on Communications*, vol. 39, pp. 11-13, January 1991.

[8] B.H. Juang and A.H. Gray, Jr., "Multiple Stage Vector Quantization for Speech Coding," *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Processing*, pp. 597-600, April 1982.

[9] A. Buzo, A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "Speech Coding Based Upon Vector Quantization," *IEEE Trans. Acoust., Speech, Sig. Processing*, pp. 562-574, Oct. 1980.

[10] G. S. Kang and L. J. Fransen, "Application of Line-Spectrum Pairs to Low-Bit-Rate Speech Encoders," *Proc. IEEE Int. Conf. Acoust.,Speech, Sign. Processing*, pp. 244-247, Tampa, March 1985.