# FINE-COARSE SPLIT VECTOR QUANTIZATION: AN EFFICIENT METHOD FOR SPECTRAL CODING

H.R. Sadegh Mohammadi and W.H. Holmes

School of Electrical Engineering
University of New South Wales

ABSTRACT - Line Spectral Frequencies (LSFs) are the most popular parameters for spectrum quantization in speech coders using linear prediction. We propose a new method for the quantization of the LSFs, namely Fine-Coarse Split Vector quantization (FCSVQ). The paper explains the principles of this method, including training and optimization of the associated codebooks. It is shown that this quantizer can be implemented efficiently with negligible computational overhead compared to simple scalar quantization. Satisfactory performance of the new method is verified through experimental simulations.

## INTRODUCTION

Short-term spectrum quantization is one of the major issues in low rate speech coders. To improve the quality of reconstructed speech in these coders, it is recommended to decrease the spectral coding distortion. This is not an easy task specially with few bits available. In linear-prediction (LP) based speech coders, such as Code Excited Linear Prediction (CELP) described in federal standard 1016 (1991), the short-term spectrum is represented by a set of parameters extracted from a tenth order LP polynomial. Among different parameters which could be used, the Line Spectral Frequencies (LSFs) are most common. The ordered pairs of LSFs are also called line spectral pairs (LSPs). Many different methods have been reported for quantization of the LSFs in the past, but the main interest is for vector quantization (VQ) methods which have low bit rate.

It has been assumed that with the current methods, more than 20 bits/frame will be necessary to quantize the entire tenth order LSF parameters by a single stage unstructured vector quantizer, if an acceptable distortion and reasonable quality is required. Since neither the computational complexity nor the memory requirement of such a quantizer is feasible for a real time speech coder, it is vital to use decomposition methods such as split VQ and multi-stage VQ to reduce the complexity and storage costs of VQ to an affordable amount.

It has been shown that if the LSFs are divided into three subvectors (3-split VQ) about 24 bits/frame are needed for vector quantization. The computational complexity is still around 0.32 MIPS (for 2-split VQ the computation cost is about eight-fold of this), which is a considerable overhead for any low rate speech encoder (Paliwal & Atal, 1991).

In general, two approaches have been suggested for decreasing the computational cost. The first proposes the fast search methods for searching the same codebooks and some of them have more storage requirements. The other reported techniques use structured codebooks for a more efficient codebook search. Obviously these need entirely different and highly structured codebooks. These methods also increase the storage cost. The proposed fine-coarse split vector quantization (FCSVQ) is an example of the second approach.

This paper is organized as follows. The next section deals with the principles of proposed method. Then the design of the related codebooks will be discussed. An experimental simulation based on a CELP speech coder is also described. Finally, conclusions are presented.

## FINE-COARSE SPLIT VECTOR QUANTIZATION

Fine-coarse split vector quantization combines the low computational complexity of a structured codebook with simplicity of the training stage of the unstructured ones. This generally results in a proper codevector selection, which most of the time almost equals to the optimum choice. The FCSVQ reduces the complexity of the quantization process significantly. Here, the concepts of this method are reviewed and its application to spectral coding is explained.

Principles

Fine-coarse vector quantization (FCVQ) was first introduced by Moayeri et al. (1991). Normally, it contains two stages of quantization with two different but related codebooks. The *fine* codebook has more codevectors than the *coarse* codebook. The input vector is quantized with the fine codebook, which is usually a highly structured codebook. Scalar quantization (SQ) and lattice vector quantization (LVQ) are two possible schemes for fine quantization. Then the selected codevector is mapped to another codevector from the coarse codebook. The latter transformation can be performed by a simple look-up table with little computational overhead.

Figure 1 shows an example of a two-dimensional FCVQ, which gives a general idea of this quantizer and the dependency between the related codevectors from the fine and coarse codebooks. In this example the FCVQ method has been applied to the quantization of a two-dimensional vector. The fine codebook is created by scalar quantization of each element of the input vector. The scalar quantizer uses a uniform quantization table with 13 levels, so the fine codebook has 169 codevectors. The coarse codebook has only 5 codewords.

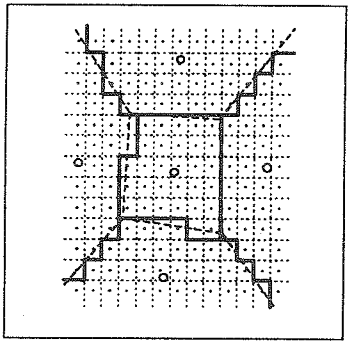

Figure 1. An example of FCVQ Voronoi cells for a two-dimensional input vector

It is quite clear that the effective shape of the Voronoi cell related to each coarse codevector in the FCVQ is the union of associated cells from the fine codebook. In other words, the shape of each Voronoi cell approximates the related cell of a similar unstructured codebook by covering that with the cells associated with the fine codewords. Obviously, as the size of the fine cells decreases (high resolution assumption), the performance of this method converges to that of the unstructured VQ.

LSF quantization

Fine-coarse split vector quantization combines the idea of FCVQ with split vector quantization. This method can be applied for efficient quantization of LSFs. The procedure is as follows. First, the LSF parameters are derived from a tenth order linear prediction. This LSF vector is divided into three subvectors with 3, 3 and 4 elements. Then each LSF is scalar quantized according to some quantization table. In fact, this stage prepares the fine subvector. This subvector is transformed to a codeword of the coarse codebook The transformation is accomplished through a look-up table mapping.

For the implementation of the FCSVQ algorithm, in the fine quantization stage, we have used the same number of quantization levels for each LSF as the FS1016 standard. This is virtually equivalent to 34 bits/frame. For the coarse quantization stage a codebook with the size of 256 is used for every subvector. This scheme ends up with 24 bits/frame for the entire LSF vector. To generate the entry for each look-up table, the indices (in binary format) of the scalar quantization of the elements in each subvector are combined. This results in a single binary word address for each subvector, which includes the indices of the quantized elements as sub-words. This address refers to the index of the coarse codeword in the look-up table. This coarse codevector is the final quantized value of the related LSF subvector. The latter index is also transmitted as data to the decoder side, where it is used directly as the entry to the coarse codebook for reconstruction of the final quantized LSFs.

In this study, a differential scalar quantization scheme has been used in the fine quantization stage. The first element of each subvector is quantized independently by the ordinary scalar quantization method, but for the other elements in the subvector the difference between the LSF and the quantized value of the previous LSF is scalar quantized. It has been shown that independent scalar quantization method is not efficient for LSF quantization, the reason for this is the monotonic increasing of the LSFs (Sadegh Mohammadi & Holmes, 1994). Despite independent scalar quantization, the differential quantization method concentrates most of the codewords in the area where the LSF vector space is dense.

## CODEBOOK GENERATION

As the previous explanation suggests, the current implementation of fine-coarse split vector quantization needs one quantization table for the fine stage, three codebooks for the coarse stage, and three look-up tables for the transformation. The training and optimization procedures for these codebooks and tables will now be described.

### Training

In this implementation of FCSVQ, the fine and coarse codebooks are trained separately, and then the look-up tables, which link the codebooks, are prepared. The scalar quantization table is trained sequentially. First the quantization levels for the first element of each sub-vector are found using the algorithm suggested by Lloyd (1982). Then the same algorithm, which is also known as the Lloyd-Max algorithm, is used to train the quantization levels for the difference between every LSF and the previously quantized LSF in each subvector. One could also use other algorithms that provide a global optimum quantization table.

In the next step, an independent unstructured coarse codebook is trained by the generalized Lloyd algorithm described in Makhoul et al. (1985). Therefore, three coarse codebooks are prepared, one for each of the fine codebooks. Of course, for the best results it is recommended that a global optimum training algorithm is used to prepare the codebooks.

Then all possible combinations of scalar quantized subvectors in the same LSF partition are classified by the codewords of the related coarse codebook. This extensive computation is performed once for the three parts of LSF vector, and it is done off-line. The results are organized as look-up tables. The combined indices of the scalar quantized LSFs in each subvector are used as the entry or address of a cell of the table, which contains the index of the related codevector from the coarse codebook.

### Optimization

Even if the optimum training procedures are used for independent training of each codebook, the resulted FCSVQ will not be optimum. Nevertheless, several enhancement methods can be used for further improvement of the FCSVQ performance. The reasons for this sub-optimality will be addressed and the solutions will be explained.

Firstly, since the scalar quantization levels are trained sequentially, the resulted quantization table causes more distortion compared to the situation in which a single step training algorithm is used. The differential scheme inherently makes the sequential procedure inevitable, but hopefully one can use an optimization procedure that provides better quantization table. This procedure is outlined here. The proposed differential scalar quantization can be seen as a multi-stage vector quantization, in which the codebooks have a particular structure, and are searched with a weighted distortion measure. In this study, an improved version of the algorithm suggested by Chan et al. (1992) for multi-stage VQ has been adapted for differential SQ and applied for optimization the scalar quantization table.

The coarse codebooks and look-up tables can be optimized jointly by the method offered by Moayeri et al. (1991). In this iterative method first the look-up tables are considered to be optimal and the coarse codebooks are recalculated. Then the resulting codebooks are assumed to be optimal and the classification is repeated for finding the look-up tables. By using this technique, the distortion gradually converges to a local minimum which provides less distortion. This method, after some improvement for the current application, which will be discussed later, is used in our study.

Since neither the fine codebooks nor the coarse codebooks have Voronoi cells with fixed size, the numbers of fine codewords which are associated with each coarse codevector are different. In fact, the non-uniform distribution of the codewords in the vector space results in some coarse codevectors having too many related fine codebooks and some too few. It has been observed in our simulations that for the allocated number of quantization levels, in some parts of the vector space the density of fine codewords is similar to that of the coarse codevectors or even less. As a result of this, some of the coarse codewords will never be used and the effective size of the codebook will be slightly less than its actual size.

To alleviate this problem two approaches can be followed. The first approach is to increase the number of scalar quantization levels, which will also increase the size of the look-up table and the storage cost. In the second approach, one improves of the optimization algorithm that jointly optimizes the coarse codebook and look-up table. All we need is a minor change in the iterative optimization algorithm. That is, in each iteration of the algorithm, after the classification of fine codewords by the coarse codebook, each unselected coarse codeword is perturbed to the nearest fine codevector which is not chosen by classification of the coarse codewords by the fine codebook. The latter approach is applied in our simulations.

## SIMULATION EXPERIMENTS

A computer simulation has been developed to evaluate the performance of the proposed quantization method. This section deals with the preparation, implementation and assessment of the experiments.

### Data preparation and training

The procedure used for extraction of the LSFs from the training and test databases is the same as the one described in the FS1016 standard. The LSFs are computed from the tenth order LP coefficients. Training of the codebooks has been done on a speech database of 10240 frames (almost 5 minutes of speech signals) spoken by different male and female speakers. Two thirds of this database is taken from the TIMIT database and the rest from the other speech samples. The test database includes 25 utterances (69 seconds of speech) with the same ratio of TIMIT/non-TIMIT data as the training database. Neither the speakers nor the sentences are common to the two databases.

The necessary codebooks and look-up tables have been prepared for the fine-coarse split vector quantization of the LSFs as explained previously. Both training and optimization stages have been used for this purpose. Also, the coarse codebooks which are derived in the training stage (before optimization for FCSVQ) are used for an unstructured vector quantization. Moreover, for the same subvector partitions and codebook size a uniform binary tree-searched vector quantization codebook is trained for each subvector. Hereafter, this method, which is well known for its good performance and affordable costs, will be called tree-searched split vector quantization (TSSVQ).

### Simulation

The standard CELP algorithm was implemented with different LSF quantization schemes, including FS1016 scalar quantization, unstructured full search vector quantization, tree-searched split vector quantization and no quantization, as well as the proposed method. The results of these experiments have been evaluated by different objective measures.

Furthermore, the computational complexity and memory requirements of the different vector quantization methods have been calculated. In this computation one instruction represents multiply-add, comparison or data format conversion. In the storage cost estimation, floating point precision is considered for scalar quantization levels, while the format of coarse codeword elements is assumed to be double precision. Finally, single precision format is allocated to the cells of look-up tables. Table 1 shows a comparison between the storage cost and computational complexity of the proposed method, TSSVQ and unstructured vector quantization.

| Quantization Method | Storage Cost [kbyte] | Computation Cost [No. of Instr.] |
|---|---|---|
| Unstructured VQ | 5.00 | 7,680 |
| TSSVQ | 9.96 | 480 |
| FCSVQ | 13.44 | 48 |

Table 1. Cost comparison

Distortion measures

Three objective distortion measures are considered for evaluation of the results. The first measure is spectral distortion (SD), which is a popular measure in the spectral coding field and is defined as

$$SD = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{\pi}\int_{0}^{\pi}\left(10\log S_n(\omega) - 10\log \hat{S}_n(\omega)\right)^2\right]^{\frac{1}{2}} \qquad [\text{dB}] \quad ,$$

where $S_n(\omega)$ and $\hat{S}_n(\omega)$ denote the original and quantized LPC spectra for the $n$-th frame of speech and $N$ is the total number of frames in the simulation.

In this study, we propose another distortion measure which is called *Synthesized Spectral Distortion* (SSD) as a second objective measure. This distortion measure is similar to the conventional spectral distortion, but it calculates the spectral distortion between the LPC spectra of the original signal and the one extracted from the synthesized speech. The use of the synthesized spectral distortion is motivated by several facts. In conventional SD it is considered that the residue resulting from passing the speech signal through the inverse filter of vocal-tract (LP filter) is later passed through the quantized version of vocal tract filter. Also, the same implementation is used for any subjective test.

There are several oversimplifications in this modelling in the use of SD. All the standard speech coders based on linear prediction calculate the target excitation by passing speech through the quantized inverse filter and not the unquantized one. Secondly, almost any frame oriented speech coder based on linear prediction uses an interpolation scheme between the quantized LP filters of the adjacent frames, and again this is discarded in the modeling. Finally, the model implicitly assumes that either the excitation is exactly reconstructed by the excitation codebooks or that the error has a totally white spectrum, neither of which is true, particularly for low rate speech coders.

The synthesized spectral distortion has none of these defects, simply because it is an assessment on the results of a real speech coder with no modelling simplification. Moreover, the SSD is not only affected by the static performance of the quantized spectrum, but it also reflects the dynamic behavior of the quantization (inter-frame effects). Of course, one should remember that the SSD is affected by both the spectral coding distortion and the distortion resulting from other parts of the coding process (e.g. excitation quantization). Nevertheless, as long as different spectral coding methods are evaluated on a single speech coder with different spectral coding schemes, such as the simulations in this paper, the results of SSD more truly reflect the performance of spectrum quantization than SD.

As a final comment about SSD, it noteworthy that possibly this measure should have a strong correlation with a true subjective test, as long as the short-term spectral envelope estimated by the linear prediction model is considered to be a proper model for formant analysis in the human auditory system. The reason is that the SSD is calculated from the short-term spectral envelopes of the original and synthesized speech.

The third objective measure is segmental signal to noise ratio, which evaluates the errors of the synthesized speech produced by decoder.

## Results and discussion

As the Table 1 indicates, the complexity of FCSVQ is less than 1% of that of unstructured VQ, and only to 10% of TSSVQ. The memory requirement of FCSVQ is almost 30% more than TSSVQ and 2.6 times that of unstructured VQ. Table 2 depicts the results of different distortion measures for the simulation of the test speech database. The results show that the performance of FCSVQ is superior to that of TSSVQ.

| Quantization Method | No. of Bits per Frame | SD [dB] | Percent. of Frame with SD>2 dB | Percent. of Frame with SD>4 dB | SSD [dB] | Seg_SNR [dB] |
|---|---|---|---|---|---|---|
| Unquantized | --- | --- | --- | --- | 2.12 | 9.17 |
| FS1016 Std. | 34 | 1.49 | 10.6 | 0.4 | 2.31 | 8.73 |
| Unstructured VQ | 24 | 1.46 | 12.9 | 0.0 | 2.35 | 8.93 |
| TSSVQ | 24 | 1.65 | 21.5 | 0.3 | 2.43 | 8.82 |
| FCSVQ | 24 | 1.61 | 19.2 | 0.2 | 2.41 | 8.84 |

Table 2. Results of simulation experiments

## CONCLUSION

The fine-coarse split vector quantization method is proposed for spectral coding in low rate speech coders. This method has been outlined and the related training and optimization issues are addressed. Also a new spectral distortion measure has been suggested. Experimental simulation verifies that the proposed method is superior to the tree-searched vector quantization scheme, with both better performance and lower computational complexity.

## ACKNOWLEDGMENTS

The first author wishes to thank Jahad Daneshgahi, Islamic Republic of Iran for supporting him during this research. We gratefully acknowledge the assistance of Mr D. Sen and Ms S. Yim for providing a simulation program for the FS1016 standard speech coder. This research was also assisted by GIRD Grant Number 17016.

## REFERENCES

Federal standard 1016 (1991) *Telecommunications: analog to digital conversion of radio voice by 4,800 bit/second code excited linear prediction (CELP)*, National Communications System, Office of Technology and Standards, Washington, DC20305-2010.

Chan, W.Y., Gupta, S. & Gersho, A. (1992) *Enhanced multi-stage vector quantization by joint codebook design*, IEEE Trans. on Commun. 40, 1693-1697.

Lloyd, S.P. (1982) *Least squares quantization in PCM*, IEEE Trans. Inform. Theory 28, 129-137.

Makoul, J., Roucos S. & Gish, H. (1985) *Vector quantization in speech coding*, Proc. IEEE 73, 1551-1558.

Moayeri, N., Neuhoff D.L. & Stark, W.E. (1991) *Fine-coarse vector quantization*, IEEE Trans. on Signal Proc. 39, 1503-1515.

Paliwal, K.K. & Atal, B.S. (1991) *Efficient vector quantization of LPC parameters at 24 bits/frame*, IEEE Trans. on Speech and Audio Proc. 1, 3-14.

Sadegh Mohammadi, H.R. & Holmes, W.H. (1994) *Efficient two-step spectrum quantization methods for low rate speech coding*, to be presented at the Intern. Symp. on Inform. Theory, & Its Appl., Sydney.