TWO NOVEL LOSSLESS ALGORITHMS TO EXPLOIT INDEX REDUNDANCY IN VQ SPEECH COMPRESSION

Sridha Sridharan† *and John Leis*‡

†Speech Research Laboratory
Signal Processing Research Centre
Queensland University of Technology
Brisbane, Queensland, AUSTRALIA

‡Faculty of Engineering
University of Southern Queensland
Toowoomba, Queensland, AUSTRALIA
leis@usq.edu.au

ABSTRACT

We address the problem of speech compression at very low rates, with the short-term spectrum compressed to less than 20 bits per frame. Current techniques apply structured vector quantization (VQ) to the short-term synthesis filter coefficients to achieve rates of the order of 24 to 26 bits per frame. In this paper we show that temporal correlations in the VQ index stream can be introduced by dynamic codebook ordering, and that these correlations can be exploited by lossless coding approaches to reduce the number of bits per frame of the VQ scheme. The use of lossless coding ensures that no additional distortion is introduced, unlike other interframe techniques. We then detail two constructive algorithms which are able to exploit this redundancy. The first method is a delayed-decision approach, which dynamically adapts the VQ codebook to allow for efficient entropy coding of the index stream. The second is based on a vector sub-codebook approach, and does not incur any additional delay. Experimental results are presented for both methods to validate the approach.

1. PROBLEM FORMULATION

Most speech coding algorithms operating at low rates depend upon the notion of short-term prediction of the sampled waveform, using a prediction $\hat{s}(n)$ which is related to the true samples $s(n)$. The linear-prediction (LP) paradigm is well-known [4, 5]. Direct quantization of the resulting LP coefficients is feasible, though not without problems (especially at low rates). The line spectrum frequency (LSF), or line spectrum pair (LSP) representation has been found useful in this regard [6, 8].

Several researchers have recently attempted to estimate the required rate for LPC quantization, with a limit of 20 bits per frame [3, 7] generally accepted. However, a VQ at this level requires not only an impractical codebook size (2^{20} entries), but a prohibitively large amount of training data. Given these constraints, a rate of 24 to 26 bits per frame is considered feasible in the current literature [1].

Much work has been carried out on vector quantization (VQ) for a variety of signal compression problems since the landmark publication of the Linde-Buzo-Gray (LBG) VQ design algorithm [2]. In the simplest case of unstructured VQ, the codebook is a finite list of vectors, $\mathbf{c}_i \,:\, i = 1, \ldots, N$. The codebook vectors are preselected through a clustering or training process to represent the training data. Vector quantization of the LSF's was demonstrated in [1], where the notion of 1dB log-spectral distortion was used extensively. For so-called “transparent” quantization, rates of 24-26 bits per frame were found to be necessary.

In this contribution, we examine the split-vector class of vector quantizers, which has been extensively studied in recent times [1]. Adjacent analysis frames are highly correlated, and is is reasonable to attempt to exploit this correlation for coding gain. The fundamental problem is to exploit the interframe redundancy without introducing additional distortion.

The results reported in this paper were obtained using the TIMIT speech corpus, downsampled to 8kHz and analyzed using a 10-th order autocorrelation at a rate of 50 frames per second. LSF analysis was then performed on the resulting coefficients. The speech used for training the VQ consisted of 100,000 frames, whilst the speech used for

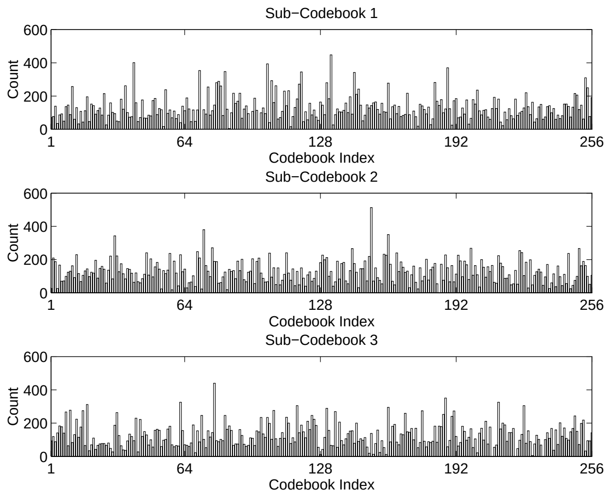

Figure 1: Codevector index distribution for the non-adaptive split vector quantizer (3×256 codebook).

testing consisted of 32,768 distinct frames. The split VQ method of [1] was utilized, with a 3-3-4 subvector split using 256-entry codebooks, for an effective rate of 24 bits per frame using a direct VQ-index encoding.

2. ALGORITHM I: DELAYED-DECISION CODING USING ADAPTIVE CODEBOOK

We first consider an adaptation algorithm applied to the split vector quantizer. This configuration gives good distortion performance combined with modest computational and storage requirements.

Figure 1 shows the histogram of allocation of codebook indices over the 32,768 test vector set. A relatively even distribution is shown over each sub-codebook, indicating relatively little gain could be extracted in the way of further entropy coding.

The codebook adaptation algorithm works as follows. After every spectral vector is quantized, we re-arrange the codebook *in order of increasing distortion, with respect to the quantized version of the input vector*. The decoder performs a similar operation, in order to ensure that the codebooks are synchronized. This scheme is evidently backwards adaptive, in that the decoder requires no additional information for the adaptation. Thus, no additional information need be transmitted: the operation has zero channel overhead.

The purpose of this dynamic codebook re-ordering is to produce a codebook whose index distribution is significantly biased towards lower-numbered indices. Elementary information theory tells us that the redundancy of the resulting sequence is reduced.

The transition from one codevector to the next is now

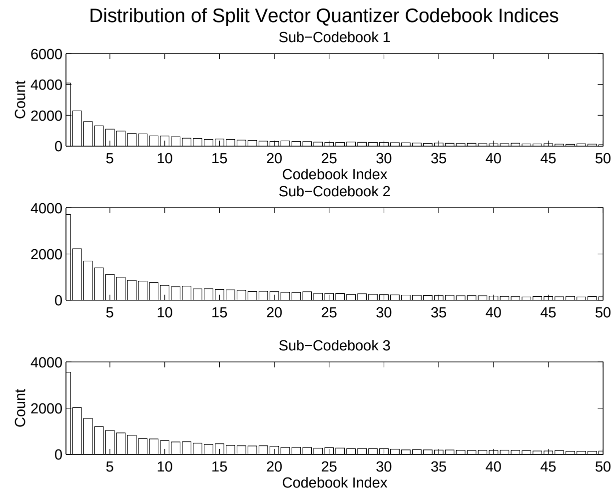

Figure 2: Codevector index distribution for the adaptive split vector quantizer (3×256 codebook).

markedly skewed. Before adaptation (Figure 3), the only index correlation observed is that from each codevector to itself. After application of the distortion-ordering algorithm, the successive-index assignment is evidently highly correlated (Figure 4).

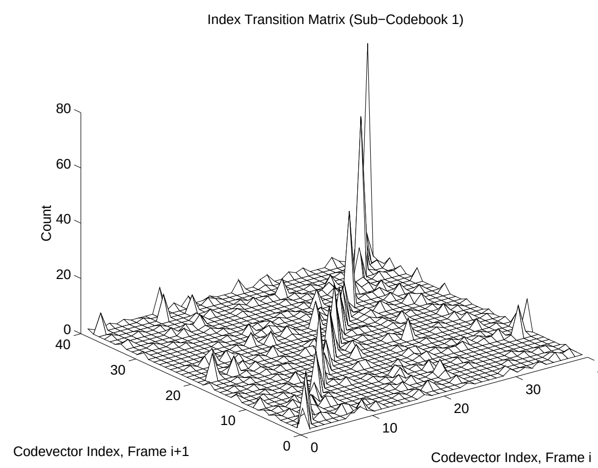

Figure 3: Transition histogram for memoryless split VQ: subvector 1, fixed codebook (first 40 vectors of 256 shown).

Using a variable-wordlength code, we are able to exploit this redundancy. Table 1 shows both the theoretical entropy and the achieved bit rate using an adaptive arithmetic encoder. Details of the arithmetic encoder may be found in [10, 9].

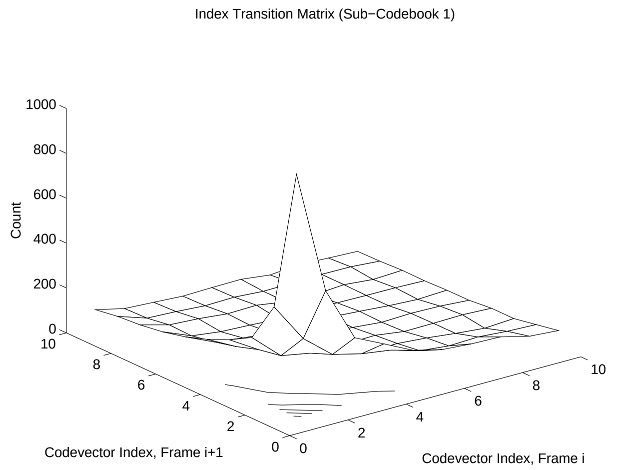

Figure 4: Transition transition histogram for split VQ subvector 1, adaptive codebook (first 10 vectors of 256 shown).

Table 1: Achievable rate for adaptive split vector quantizer using adaptive arithmetic coding.

	Subvector Split			Total
	1	2	3	
Theoretical	6.38	6.28	6.55	**19.21**
Achieved	6.41	6.32	6.58	**19.31**

3. ALGORITHM II: INSTANTANEOUS CODING USING A VECTOR SUB-CODEBOOK

Algorithm II does not update the codebook dynamically, as does Algorithm I. Instead, we examine the transition histogram of the unstructured codebook, and observe that the transition from one index to another is highly dependent upon the current state of the coder, as defined by the just-coded index. We begin by training the codebook in the conventional manner, using a standard training algorithm such as the Linde-Buzo-Gray method. Then, using the training sequence, we generate a conditional count over the training sequence for each codebook vector, specifying the count of the previous codevector index. Thus we generate the conditional histogram matrix of size $L \times L$, where the codebook size $L = 256$. We retain, in order, the highest C codebook indices for each of the L codevectors. This is illustrated diagrammatically in Figure 5. The memory size is increased by $L \times C$ additional indirection indices for each vector codebook (one per subvector codebook, or four in the case shown in Figure 5).

The vector search procedure is then modified as follows. For the previous codevector k at time $(n-1)$, we search the corresponding row of the sub-codebook, containing C entries. The full codevector is found by traversing the codebook double indirection indices to the codebook vectors themselves. An exhaustive search is then performed over the full codebook. Note that it is not necessary to test those codevectors which were nominated in the sub-codebook search stage. The effective encoding is then comprised of:

(i) A one-bit flag to indicate if the lowest-distortion vector is found in the sub-codebook;

(ii) If the lowest distortion codevector was found in the sub-codebook, an index of $\log_2 C$ bits;

(iii) If the lowest distortion codevector was not found in the sub-codebook, an index of $\log_2 L$ bits into the full codebook.

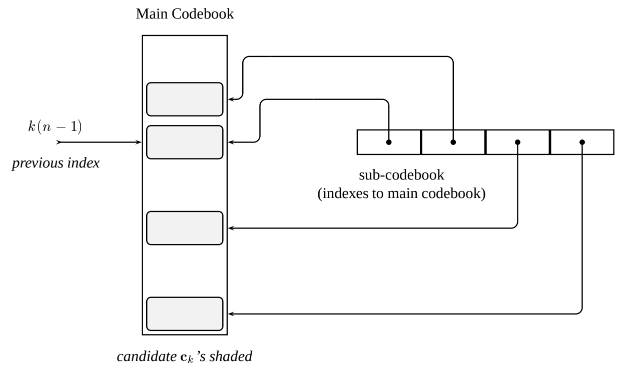

Figure 5: Structured sub-codebook search/encoding algorithm.

Obviously the success of the method depends on the number of times the lowest-distortion codevector is specified by the sub-codebook. We must experimentally determine the likely the hit rate for various sub-codebook sizes. For a 32,768 test vector sequence (outside the 100,000 training vector sequence) we have found relatively high hit rates, as shown in Figure 6.

We now combine the hit rate and sub-codebook size parameters to determine the overall performance. This is shown in Figure 7, where we see that clearly there exists an optimal solution of four index bits per codevector. This gives us a measured rate of approximately 7 bits per subvector, as compared to a full-rate of 8 bits. Thus, the method is effective at exploiting the inter-block index correlation.

4. CONCLUSIONS

We have described two methods for exploiting the interframe redundancy which exists in vector-quantized speech. The first is based on an adaptive VQ codebook, which requires a distortion-ordered sort after each frame is encoded. It was demonstrated to be suitable for split-vector quantization, reducing the rate from 24 bits per frame to 20 bits when

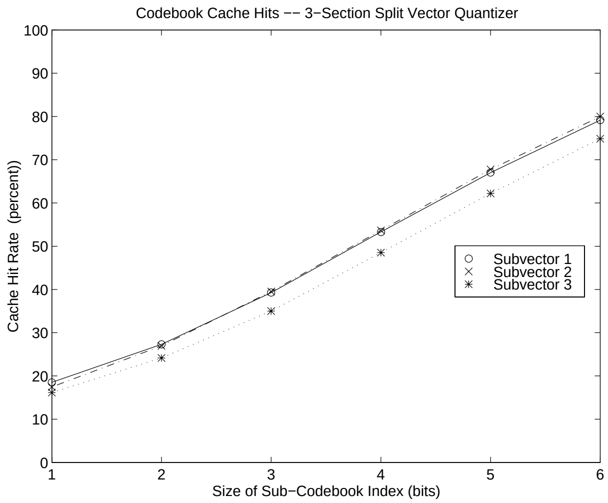

Figure 6: Hit rate for the fixed-cache split vector quantizer.

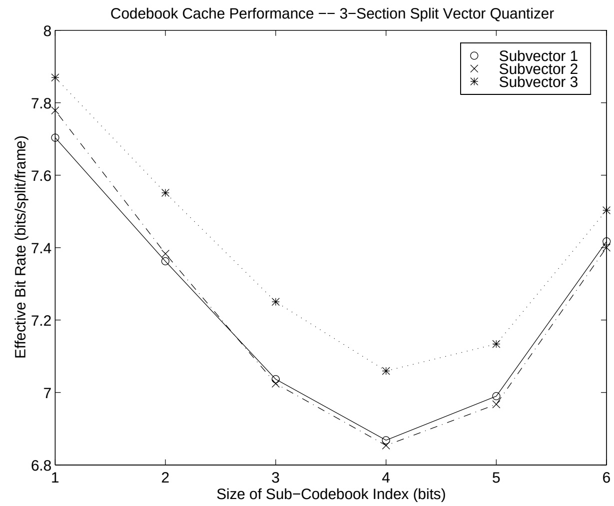

Figure 7: Address-codebook method: bit rate results for the split vector quantizer.

combined with delayed-decision adaptive arithmetic codeword assignment. The second method requires a pre-trained vector sub-codebook. As such, there is little additional computational requirement during encoding (although the off-line training procedure is more complex). Additionally, no special variable-wordlength codeword assignment is needed. The rate in this case was also demonstrated to be reduced to approximately 20 bits per frame.

5. REFERENCES

[1] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 3–14, Jan. 1993.

[2] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, vol. COM-28, no. 1, pp. 84–94, Jan. 1980.

[3] Per Hedelin, "Single Stage Spectral Quantization of 20 Bits", *Proc. ICASSP'94*, pp. I525–I528, 1994.

[4] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, chapter 5 - Digital Representations of the Speech Waveform, Prentice-Hall, 1978.

[5] B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", *J. Acoust. Soc. Amer.*, vol. 50, no. 2, pp. 637–655, 1971.

[6] F. Itakura, "Line Spectral Representation of Linear Predictive Coefficients of Speech Signals", *J. Acoust. Soc. Amer.*, vol. 57, no. S35, 1975.

[7] K. K. Paliwal and W. B. Kleijn, "Vector Quantization of LPC Parameters", in *Speech Coding and Synthesis*. Elsevier, 1995.

[8] F. K. Soong and B.-H. Juang, "Optimal Quantization of LSP Parameters", *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 15–24, Jan. 1993.

[9] I. H. Witten, R. N. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression", *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, June 1987.

[10] J. Rissanen and G. G. Langdon, "Arithmetic Coding", *IBM Journal of Research and Development*, vol. 23, no. 2, pp. 149–162, Sept. 1979.