S4.11

Pitch Sharpening for Perceptually Improved CELP, and the Sparse-Delta Codebook for Reduced Computation

Tomohiko Taniguchi, Mark Johnson*, and Yasuji Ohta

Speech Signal Processing, Fujitsu Laboratories Ltd.
1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan

ABSTRACT

This paper will introduce two techniques for improving low bit rate CELP speech coders. The sparse-delta stochastic codebook is a recursive codebook design which can be searched using roughly 5% of the computational load required to search a full Gaussian codebook. Pitch sharpening is a class of algorithms which attempt to improve the perceptual quality of CELP by limiting the feedback of low-amplitude noise-like information to the adaptive codebook. We will present simulation results for sparse-delta, ternary sparse-delta, and simplified-search sparse-delta coders, and for reduced-gain and sparse-adaptive-codebook pitch sharpening algorithms.

1. INTRODUCTION

Previous authors have shown that the stochastic codebook in a CELP coder can be searched efficiently, using an FIR perceptual weighting filter, if the codevectors are sparse. It has also been shown that if the codevectors are very sparse, the weighting filter can be discarded altogether, and an even more efficient algorithm known as the modified sparse-vector fast-search (modified SVFS) can be employed [1].

This paper will introduce two algorithms for improving low bit rate CELP. In section 2, we will present the "sparse-delta" stochastic codebook structure, a pseudo-random codebook which can be efficiently searched using a recursive algorithm based on the modified SVFS. In section 3, we will give two sample "pitch sharpening" algorithms, which will improve the perceptual quality of CELP by limiting the feedback of stochastic noise into the adaptive codebook: one by altering the feedback synthesis gains, and the other by simply center-clipping the adaptive codebook. In section 4, we will review the conclusions drawn in each of the previous sections.

2. THE SPARSE-DELTA CODEBOOK

2.1 The Stochastic Codebook Search

Figure 1 shows the block diagram of a standard CELP speech coder. As shown, CELP quantizes the LPC residual in an input speech signal using a pair of excitation vectors, the pitch and code vectors, which are chosen sequentially to minimize the perceptually weighted MSE. If A is the perceptual weighting matrix, except where it appears in the target vector $A\mathbf{X}$, then the weighted MSE can be written as $|A\mathbf{X} - bA\mathbf{P} - gA\mathbf{C}|^2$. If we ignore the need to jointly optimize $\mathbf{C}$ with $\mathbf{P}$ [2], then the vector $\mathbf{C}$ which minimizes the weighted MSE will also maximize the function:

$$F(\mathbf{C}) = (\mathbf{C}^T A^T A \mathbf{X})^2 / \mathbf{C}^T A^T A \mathbf{C} \qquad (1)$$

F(C) is normally computed by perceptually weighting to find AC, and taking two vector products to find the numerator and denominator shown. Davidson and Gersho have shown in [1], however, that if the codevectors C are very sparse (more than 90% zero-valued), then equation (1) can be calculated most cheaply by using the modified Sparse-Vector Fast Search (SVFS). The modified SVFS finds the quantities $A^T A\mathbf{X}$ and $A^T A$ in advance, and then multiplies these quantities by the unweighted code vectors, taking advantage of the zero elements in each C-vector in order to reduce the codebook search complexity.

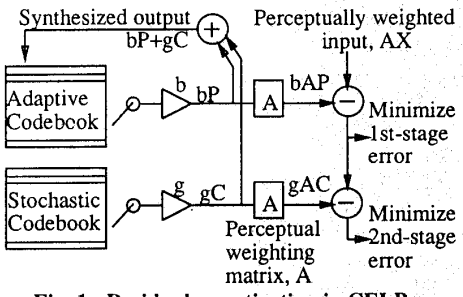

Fig. 1: Residual quantization in CELP

The overlapped codebook proposed by Lin [3] is a good example of a recursive codebook. In an overlapped codebook, each codevector is designed by shifting the previous codevector a given number of samples, and filling in the empty locations with new random sample values. It turns out that an overlapped codebook can be efficiently searched by calculating the weighted codevectors AC recursively, one after another, and that this computation is even further reduced if the codebook is sparse (mostly zero-valued). A very sparse codebook, however, is no less easy to search if it happens to be overlapped, since the directness of the modified SVFS leaves no room for the shifted redundancy of an overlapped codebook.

*currently studying in the Department of Electrical Engineering and Computer Science, MIT, Cambridge, Massachusetts, USA

CH2977-7/91/0000-0241 $1.00 © 1991 IEEE

2.2 The Sparse-Delta Codebook

The sparse-delta stochastic codebook is a recursive codebook design which, rather than shifting each codevector to form the next, simply changes a given number of samples in-place. This allows us to use a recursive implementation of equation (1), which we will call the Recursive Sparse-Delta Fast-Search (RSDFS), to search the stochastic codebook without perceptual weighting, regardless of the actual sparsity of the codebook.

In order to understand the RSDFS, we can write the in-place sample replacement by which the sparse-delta codebook is designed as the addition of a very sparse "delta vector," as shown:

$$\mathbf{C}_i = \mathbf{C}_{i-1} + \Delta\mathbf{C}_i \tag{2}$$

where i is the codebook index, and $\mathbf{C}_0$ is set to some reasonable initial vector. The RSDFS, then, calculates the numerator and denominator in equation (1) as follows:

$$R_{XC(i)} = \mathbf{C}_i^T A^T A\mathbf{X} = R_{XC(i-1)} + \Delta\mathbf{C}_i^T A^T A\mathbf{X} \tag{3}$$

$$R_{CC(i)} = \mathbf{C}_i^T A^T A\mathbf{C}_i \tag{4}$$

$$= R_{CC(i-1)} + 2\Delta\mathbf{C}_i^T A^T A\mathbf{C}_{i-1} + \Delta\mathbf{C}_i^T A^T A\Delta\mathbf{C}_i$$

If the number of non-zero samples in $\Delta\mathbf{C}$ is low, then a sparse-delta codebook can be efficiently searched, regardless of the sparsity of the actual code vectors $\mathbf{C}_i$.

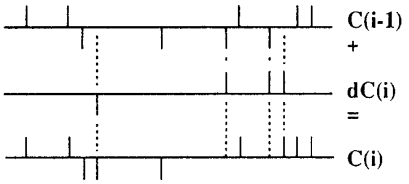

Fig. 2: Unity-variance ternary sparse-delta

2.3 The Unity-Variance Search Criterion

In order to make our codebook even more computationally efficient, we can make the codevectors themselves sparse, and even ternary-valued, so that their amplitudes are restricted to the set {+1,0, and -1} [4]. The code vectors shown in Fig. 2 are restricted in this fashion, and have, in addition, one further restriction: the number of non-zero samples in any given vector is constant. This means that the delta-vector itself is ternary-valued, making the RSDFS extremely efficient. Not satisfied with this efficiency, however, we will add one more simplifying approximation.

If the number of non-zero samples per vector in Fig. 2 is constant, then the unweighted power of each vector, $\mathbf{C}^T\mathbf{C}$, is constant. Unfortunately, the weighted power used in the standard search criterion, $\mathbf{C}^T A^T A\mathbf{C}$, is still variable. In order to make use of the constant $\mathbf{C}^T\mathbf{C}$, however, we can try simplifying the codebook search criterion. We propose choosing a code vector which maximizes the function:

$$F(\mathbf{C}) = R_{XC}^2/\mathbf{C}^T\mathbf{C} = kR_{XC}^2 \tag{5}$$

If we can actually use this search criterion without significantly degrading the synthesized speech, then we can test each code vector for optimality using only the computation required for equation (3).

Table 1 gives segmental SNR values and computational complexity estimates at 4.8 Kbps for several of the codebooks and codebook search algorithms discussed so far. As can be seen, the table represents several possible trade-offs between computational complexity and reconstructed speech quality.

Codebook Type	Segmental SNR	Estimated Complexity
Full Gaussian	13.2 dB	99 Mflops
Sparse Gaussian, 75% zero-valued	13.4 dB	72 Mflops
Overlapped Sparse, 1 new sample/vector	13.3 dB	15 Mflops
Sparse-Delta, 2 new samples/vector	13.1 dB	10 Mflops
Ternary Sparse-Delta, 4 new samples/vector	13.1 dB	21 Mflops
Above, with Simplified Search Criterion	12.0 dB	1.0 Mflops

Table 1: Complexity and SNR of CELP coders with several kinds of stochastic codebooks

3. PITCH SHARPENING

3.1 Noise in the Adaptive Codebook

In theory, the adaptive codebook in CELP is supposed to represent the periodic component in the LPC residual. Since the adaptive codebook is fed back from the synthesized excitation, however, instead of being fed forward from the input residual, this purpose may be thwarted by additive noise in the codebook. Fig. 3 shows how this can happen.

The LPC residual in a CELP coder is formed by adding the pitch vector to a stochastic code vector, which is designed to represent an uncorrelated noise process. Naturally, neither of these vectors is a perfect match with the input signal, and there will be a certain amount of quantization noise. Since the stochastic code vector is designed to look like uncorrelated noise, the stochastic codebook quantization noise, in particular, will look like a low-amplitude uncorrelated noise signal added to the output.

In the excitation for periodic speech, however, the actual signal events will tend, in theory at least, to happen only in those parts of the signal during which air is flowing through the glottis, as shown in Fig. 3. This will leave large portions of the synthesized LPC residual which are zero, except for the low-level uncorrelated quantization noise noted above.

The adaptive codebook will duly read in these noisy synthesized waveforms, and attempt to match them with the input signal one or two pitch periods later. Naturally, the quantization noise will not be matched, but since the squared-error

criterion tends to ignore small, evenly-spread errors, the coder will generally choose the correct pitch period regardless of the noise. Now, however, both the adaptive and the stochastic codebooks will be adding low-amplitude uncorrelated noise to the synthesized output.

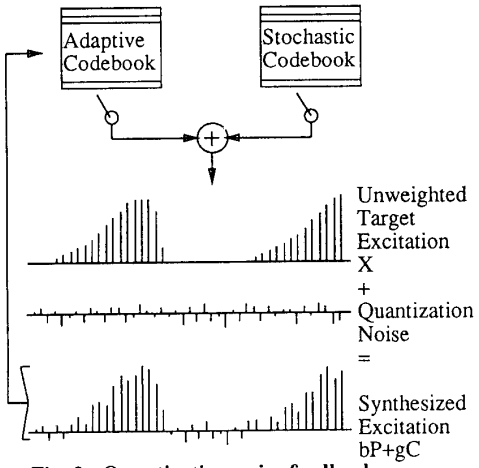

Fig. 3: Quantization noise feedback to the adaptive codebook

3.2 Pitch Sharpening

Several techniques have been proposed to help eliminate this noise in the adaptive codebook. Kroon and Atal [5] proposed low-pass filtering the adaptive codebook, since the actual speech events in periodic excitation tend to be slowly-moving, so that the high-frequency part of the quantization noise is most noticeable. Similarly, Wang and Gersho [6] have proposed comb filtering both the adaptive codebook and the synthesized speech, on the assumption that any energy between the pitch harmonics is quantization noise, and should be discarded.

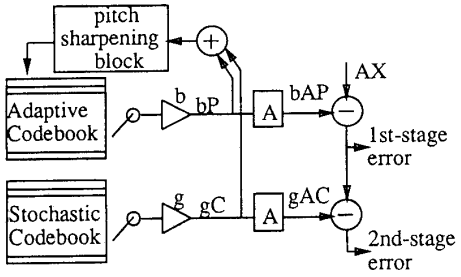

Fig. 4: CELP with pitch sharpening

This section will propose a generalized block structure for systems such as these which tweak the long-term predictor to reduce the stochastic quantization noise in the adaptive codebook. We can draw this generalized structure as a non-linear feedback component, as shown in Fig. 4, which takes the noisy feedback from the coder output, and "cleans up" all of the non-periodic components, leaving only the signal component which is expected to be periodic. This "sharpened" image of the pitch component is then fed into the adaptive codebook, and used to model the next frame of the input speech.

Such a system should have two goals. First, if the adaptive codebook contains a good model of the input speech periodicity, the pitch sharpening algorithm should protect the codebook from unwanted noise feedback. Second, however, if the model is poor, then we want the adaptive codebook to change rapidly, in order to lock on to the new pitch period as quickly as possible.

The next two sections will discuss computationally efficient pitch sharpening algorithms which fit the two goals stated above.

3.3 Reduced-Gain Pitch Sharpening

One way to reduce the effect of feedback from the stochastic codebook is to simply scale down the feedback. Fig. 5 shows how this can be done.

Instead of making the adaptive codebook out of delayed output samples, the coder in Fig. 5 divides the synthesis into two roughly parallel blocks. In one block, the stochastic and adaptive code vectors are optimally scaled, as normal, and added together, and used to excite the LPC output synthesis filter. The other block scales the code vectors using an MSE-sub-optimal gain set, and uses the result to replenish the adaptive codebook, as shown.

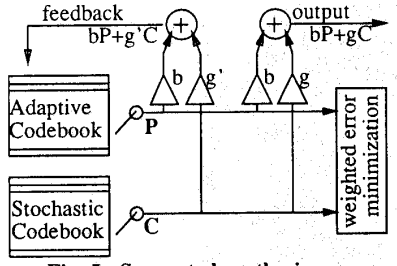

Fig. 5: Separated synthesis for reduced-gain pitch sharpening

Separating the synthesis blocks, as shown in Fig. 5, allows us to modify the stochastic codebook gain to fit the two goals of pitch sharpening. First, in order to protect the adaptive codebook from quantization noise, we can reduce the feedback gain of the stochastic codebook whenever the pitch is a good model of the input. Second, in order to allow rapid adjustment of the adaptive codebook, we can leave the gain unreduced at all other times. If the input signal were available at the decoder, we could calculate the pitch SNR, as in Shoham's constrained excitation algorithm [7], and scale the feedback gain proportionately. Since the pitch SNR isn't available to the decoder, however, we will choose a related metric. One easy-to-calculate possibility is the quantized fractional code energy:

$$g'\left(g_{feedback}\right) = \frac{|gC|}{|bP + gC|}\, g_{optimum} \qquad (6)$$

In computer simulation, the SNR of CELP with pitch sharpening tends to be slightly less than that of normal CELP, but

the perceptual quality is consistently and noticeably improved. Fig. 6 shows why this might be so. By reducing the noise-like component of the adaptive codebook, pitch sharpening makes it more difficult for the coder to match the high-frequency input information, so that the full-band SNR is degraded. In the low frequency bands which carry periodic information, and to which the human ear is most sensitive, however, the proposed algorithm provides a much better match to the input than does conventional CELP.

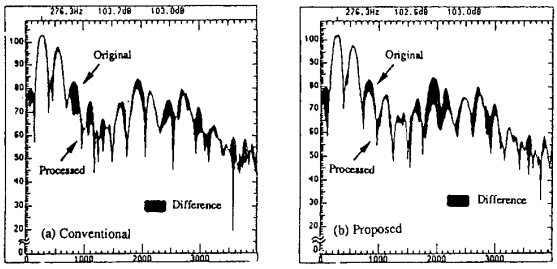

Fig. 6: Spectrograms of coded speech
a) conventional CELP
b) CELP with reduced gain pitch sharpening

3.4 Sparse-Codebook Pitch Sharpening

A second way to reduce the effect of stochastic feedback is to center-clip the adaptive codebook. If, for example, the synthesized excitation vector has the shape of a glottal pulse added to uncorrelated noise, as shown in Fig. 3, then we can eliminate half the uncorrelated noise in the adaptive codebook by simply center-clipping the synthesized residual before feedback.

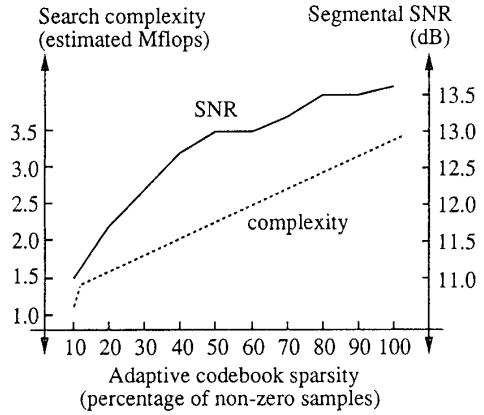

Fig. 7: Quality and complexity of CELP with center-clipped adaptive codebooks

As might be expected, the segmental SNR of a CELP coder is slightly worse when the coder uses a sparse adaptive codebook, but as shown, the decrease is not linear. Apparently, the adaptive codebook can be clipped to roughly 50% sparsity without any significant drop in the segmental SNR. This corresponds well with the theoretical excitation waveform shown in Fig. 3, since the lowest-amplitude 50% or so of the samples shown in Fig. 3 are mostly noise, and contain little of the important periodic signal information.

While the SNR has a noticeable knee at about 50%, however, the computational complexity of searching the adaptive codebook is almost a strict linear function of the sparsity. Thus, although sparse-codebook pitch sharpening doesn't provide nearly the perceptual improvement of reduced-gain pitch sharpening, it might still be desirable to clip the adaptive codebook to about 50%, simply to take advantage of the resulting complexity reduction.

4. CONCLUSIONS

In this paper, we have presented two new algorithms for the improvement of low-bit-rate CELP.

First, we showed how the sparse-delta stochastic codebook can be searched using a minimum of computation, while maintaining a reasonable output speech quality. In particular, we showed that it is possible to construct a ternary-valued sparse-delta codebook in which all of the unweighted code vectors have the same power, and to search this codebook, using a modified search criterion which we introduced, in a small constant number of additions per code vector.

Second, we showed that the perceptual quality of low bit rate CELP is sometimes degraded when quantization noise from the stochastic codebook is fed back into the adaptive codebook. We introduced a class of algorithms known as pitch sharpening which can be used to solve this problem. We then presented two sample pitch-sharpening algorithms. Reduced-gain pitch sharpening provided a clear perceptual improvement by directly reducing feedback from the stochastic codebook. Center-clipping the adaptive codebook, on the other hand, provided less perceptual improvement than hoped for, but had the added advantage of reducing the adaptive codebook search complexity.

REFERENCES

[1] G. Davidson and A. Gersho, "Real-Time Vector Excitation Coding of Speech at 4800 bps," *Pro. ICASSP*, pp. 2189-2192: April 1987.

[2] M. Johnson and T. Taniguchi, "Low-Complexity Multi-Mode VXC Using Multi-Stage Optimization and Mode Selection," *Proc. ICASSP*, to appear: May 1991.

[3] D. Lin, "Speech Coding Using Pseudo-Stochastic Block Codes," *Proc. ICASSP*, pp. 1354-1357: April 1987.

[4] J-P. Adoul, P. Mabilleau, M. Delprat, and S. Morisette, "Fast CELP Coding Based on Algebraic Codes," *Proc. ICASSP*, pp. 1957-1960: April 1987.

[5] P. Kroon and B.S. Atal, "Strategies for Improving the Performance of CELP Coders at Low Bit Rates," *Proc. ICASSP*, pp. 151-154: April 1988.

[6] S. Wang and A. Gersho, "Improved Excitation for Phonetically-Segmented VXC Speech Coding Below 4 Kb/s," *Proc. GLOBECOM*, pp. 946-950: Dec. 1990.

[7] Y. Shoham, "Constrained-Stochastic Excitation Coding of Speech at 4.8 kb/s," *International Conf. on Spoken Lang. Processing*, pp. 645-648: Nov. 1990.