TOLL-QUALITY 16 KB/S CELP SPEECH CODING WITH VERY LOW COMPLEXITY

Juin-Hwey Chen

Speech Coding Research Department
AT&T Bell Laboratories
Murray Hill, New Jersey, USA

ABSTRACT

In this paper, we present a 16 kb/s CELP coder with a complexity as low as 3 MIPS. The main thrust is to reduce the complexity as much as possible while maintaining toll-quality. This Low-Complexity CELP (LC-CELP) coder has the following features: (1) fast LPC quantization, (2) 3-tap pitch prediction with efficient open-loop pitch search and predictor tap quantization, (3) backward-adaptive excitation gain, and (4) a trained excitation codebook with a small vector dimension and a small codebook size. Most CELP coders require one full DSP or even two DSP chips to implement in real-time. In contrast, 3 to 6 full-duplex LC-CELP coders can fit into a single DSP chip, since each takes only around 3 MIPS to implement. This coder achieved slightly higher mean opinion scores (MOS) than the CCITT 32 kb/s ADPCM. It also exhibits good performance when tandemed with itself or transcoded with other coders.

1. INTRODUCTION

The Code-Excited Linear Prediction (CELP) coder was first proposed in 1984 [1] with a goal of achieving high speech quality around 4.8 kb/s. This initial CELP coder showed a significant performance gain over existing speech coders at that time, but it had an astronomical complexity — well over 400 million multiply-adds per second. Within two to three years, several researchers devised many complexity reduction techniques to bring the complexity of CELP or CELP-like coders to a more reasonable level. Since 1984, there have been numerous papers on complexity reduction methods for CELP. See [2] for a comprehensive review of these fast methods. It should be noted that even with such complexity reduction techniques, CELP coders are generally still regarded as high-complexity speech coders, since most CELP coders still require one full DSP or even two DSPs to implement in real-time.

Most CELP coders proposed so far have a bit-rate of 8 kb/s or below. One important reason is that digital cellular radio has been a major driving force for CELP coding research in the past several years. For digital cellular radio, a low encoding rate is important due to bandwidth limitation of radio channels. However, there are also many other speech coding applications where bandwidth is not so scarce a resource, but it is essential to have a coder complexity as low as possible while maintaining high speech quality. Examples in this category include voice mail, voice response, voice announcement, digital answering machines, multimedia, and other voice storage applications. These applications require a low-complexity coder either because they are cost sensitive due to fierce price competition, or because a very large number of voice channels have to fit into a given system chassis. The 16 kb/s Low-Complexity CELP (LC-CELP) coder was created to satisfy the needs of such applications or other applications where low complexity and high quality are both important.

In the following, we first describe the LC-CELP algorithm in Section 2, and then discuss the coder's performance and complexity in Sections 3 and 4, respectively.

2. CODER DESCRIPTION

Figure 1 shows a block diagram of the LC-CELP encoder. As usual, the decoder is simply the part of the encoder from the excitation VQ codebook to the short-term filter, so its block diagram is omitted here. In conventional CELP coders, usually everything except the LPC parameters is closed-loop quantized. In LC-CELP, only the excitation codevector is closed-loop quantized. Both the LPC parameters and pitch predictor parameters are *open-loop* quantized, and the gain is not quantized or transmitted at all — it is backward adapted. The LC-CELP algorithm is described in more details in the following subsections.

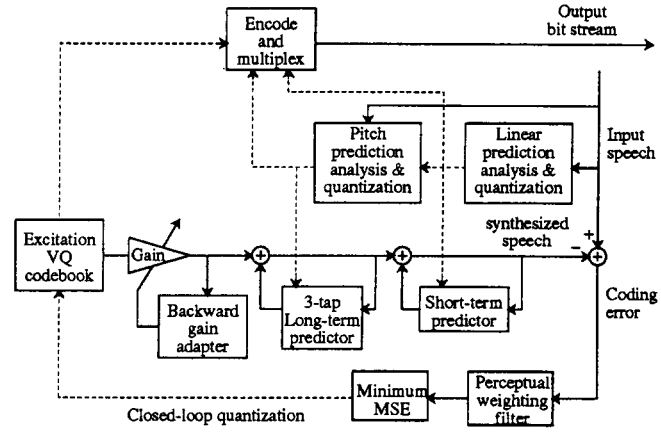

Fig. 1 Encoder of Low-Complexity CELP

2.1 LPC Predictor

The LPC frame size of the LC-CELP coder is 24 ms (192 samples at 8 kHz sampling). For each frame of 24 ms, there are 4 sub-frames each 6 ms (48 samples) long. A 24 ms Hamming window is centered at the middle of the 4th sub-frame. A 10th-order LPC predictor is obtained by using the autocorrelation method of LPC analysis. To save computation, there is no overlap between

0-7803-2431-5/95 $4.00

adjacent Hamming windows. In fact, we tried 30 ms Hamming window with 6 ms overlap, but this did not provide noticeable improvement in speech quality.

After LPC analysis, we perform a bandwidth expansion [3] of 15 Hz, corresponding to a bandwidth expansion factor of 0.9941. Then, the bandwidth-expanded predictor coefficients are converted to reflection coefficients for quantization. Most LPC-based coders use Line Spectral Pairs (LSPs), log-Area Ratios (LARs), or arc sine transform of RCs [4] for quantization of LPC parameters. To reduce the complexity, in LC-CELP we quantize reflection coefficients directly using a method that is mathematically equivalent to using the arc sine transform quantization approach.

Typically, with the arc sine transform approach, the 10 reflection coefficients are first converted to the arc sine domain and quantized there using the *nearest-neighbor search* rule. The 10 quantized arc sine values are then transformed back to reflection coefficients by applying the sine function.

One obvious way to avoid the calculation of the arc sine and sine functions during encoding is to apply the sine function to the quantizer levels in the arc sine domain and store such pre-computed results in a table. Then, in actual encoding, reflection coefficients are quantized directly using this table and the nearest-neighbor search rule. This short-cut, however, does not give the same result. The reason is that the sine function "warps" the scale, so the nearest neighbor search may not pick the same nearest neighbor as in the original approach.

We use the following approach instead. We first design a Lloyd-Max scalar quantizer in the arc sine domain for each reflection coefficient. We then calculate the quantizer cell boundaries by taking the mid-point between adjacent quantizer representative levels. Next, we apply the sine function to these cell boundaries and quantizer representative levels and stored the two sets of results in two separate tables. In actual encoding, each of the 10 unquantized reflection coefficients is directly compared with its own quantizer cell boundary table to identify which quantizer cell it belongs to. A binary tree search is used to speed up this process. Once the cell is identified, we use the index of that cell to extract the the corresponding quantizer representative level from the other table. The result is the quantized reflection coefficient. This approach is faster than the original arc sine quantization approach, because no sine or arc sine function evaluations are required, and because the binary tree search is efficient.

A total of 44 bits are used to quantize the 10 reflection coefficients. The bit allocation is 6,6,5,5,4,4,4,4,3,3 bits for the first through the tenth reflection coefficients. The quantized reflection coefficients are used for the fourth sub-frame, and we use linear interpolation to obtain reflection coefficients for the first three sub-frames. The reflection coefficients are converted to LPC predictor coefficients for each sub-frame.

2.2 Perceptual Weighting Filter

The perceptual weighting filter of LC-CELP has the same form as that of the original Low-Delay CELP coder [5]. Namely, it has a transfer function of

$$W(z) = \frac{1 - P(z/0.9)}{1 - P(z/0.4)},$$

where $P(z)$ is the transfer function of the interpolated LPC predictor, which varies from sub-frame to sub-frame.

2.3 Pitch Predictor

For each sub-frame, the pitch period is updated using the following fast search method. First, the input speech is passed through a tenth-order LPC inverse filter which has the same coefficients as the interpolated LPC predictor. The resulting LPC prediction residual is lowpass filtered at 1 kHz by a third-order elliptic filter and then 4:1 decimated. Then, we compute the correlation of the decimated LPC residual for time lags from 5 to 30. The lag τ which gives the largest correlation is identified. Since τ is the lag in the 4:1 decimated signal domain, the corresponding time lag that gives the maximum correlation in the original undecimated signal domain should lie between $4\tau - 3$ and $4\tau + 3$. To get the original time resolution, we then compute the correlation of the undecimated LPC residual for lags between $4\tau - 3$ and $4\tau + 3$. The lag p that gives the maximum correlation in this range is picked as the pitch period for use in the pitch predictor. This final pitch period is in the range of 20 to 120 samples and is encoded into 7 bits. To save computation, we do not use the pitch range of 121 to 147 samples even though 7-bit pitch encoding allows us to use it. The analysis window size for calculating the correlation in both domains is 6 ms — the same as the sub-frame size.

The 4:1 decimation technique is like a "pre-search" which quickly "zooms in" to the most likely region of the pitch period. A full-search in that neighborhood then finds the exact pitch period. This fast search method gives essentially the same pitch predictor performance as the conventional full search method in the undecimated signal domain, but it greatly reduces the computational complexity. Except for the third-order lowpass filtering, which takes only a small amount of computation, the pre-search procedure only needs 1/16 of the computation of the conventional full-search method. This is because the number of correlation values that need to be computed is reduced by a factor of 4, and the number of multiply-adds required for calculating each correlation value is also reduced by a factor of 4 because the decimated signal buffer has fewer samples. This decimation approach in pitch search has been used in the SIFT algorithm [4]. However, a parabolic interpolation is used in SIFT to find the pitch period in the undecimated domain, while here we perform a full search through the 7 lags around 4τ in order to find the pitch period more accurately.

For better coding efficiency, we use vector quantization (VQ) to jointly quantize the three pitch predictor taps into 6 bits. The distortion criterion of the VQ codebook search is the energy of the open-loop pitch prediction residual, rather than a more straightforward mean-squared error of the three taps themselves. The residual energy criterion gives better pitch prediction gain than the coefficient MSE criterion. However, it normally requires much higher complexity in the VQ codebook search, unless the following fast search method is employed.

Let b_1, b_2, and b_3 be the three pitch predictor taps and p be the pitch period determined above. Then, the three-tap pitch predictor has a transfer function of

$$B(z) = \sum_{i=1}^{3} b_i z^{-p+2-i}$$

Without loss of generality, we can assume the current sub-frame of LPC residual samples are $d(1)$, $d(2)$, ..., $d(L)$, where $L = 48$. Then, the energy of the open-loop pitch prediction residual is

$$D = \sum_{n=1}^{L} \left[d(n) - \sum_{i=1}^{3} b_i d(n-p+2-i) \right]^2$$

$$= E - 2\sum_{i=1}^{3} b_i \psi(2-p, i) + \sum_{i=1}^{3}\sum_{j=1}^{3} b_i b_j \psi(i, j) \ ,$$

where

$$\psi(i, j) = \sum_{n=1}^{L} d(n-p+2-i)d(n-p+2-j), \text{ and } E = \sum_{n=1}^{L} d^2(n) \ .$$

Note that D can be expressed as $D = E - \mathbf{c}^T\mathbf{y}$, where

$$\mathbf{c}^T = [\psi(2-p,1), \psi(2-p,2), \psi(2-p,3), \psi(1,2), \psi(2,3),$$

$$\psi(3,1), \psi(1,1), \psi(2,2), \psi(3,3)] \ , \text{ and}$$

$$\mathbf{y} = [2b_1, 2b_2, 2b_3, -2b_1b_2, -2b_2b_3, -2b_3b_1, -b_1^2, -b_2^2, -b_3^2]^T \ .$$

Therefore, minimizing D is equivalent to maximizing $\mathbf{c}^T\mathbf{y}$, the inner product of two 9-dimensional vectors. For each of the 64 candidate sets of pitch predictor taps in the 6-bit codebook, there is a corresponding 9-dimensional vector $\mathbf{y}$ associated with it. We can pre-compute and store the 64 possible 9-dimensional $\mathbf{y}$ vectors. When encoding the pitch predictor taps, the 9-dimensional correlation vector $\mathbf{c}$ is first computed. The 64 inner products between $\mathbf{c}$ and the 64 stored $\mathbf{y}$ vectors are calculated next, and the $\mathbf{y}$ vector giving the largest inner product is identified. The three quantized predictor taps are then obtained by multiplying the first three elements of this $\mathbf{y}$ vector by 0.5. This approach was first described in [6].

2.4 Backward-Adapted Gain

Each 48-sample sub-frame is divided into 12 vectors, where each vector has 4 samples. The gain for each vector is backward-adapted using a first-order gain predictor in the logarithmic domains. The backward gain adapter is similar to that of Low-Delay CELP [7], except that the gain predictor order is reduced from 10 to 1, and the gain predictor coefficient is updated once every 6 ms (one sub-frame) rather than once every 2.5 ms. The complexity goes down with such reduction of the gain predictor order and update frequency

The backward-adaptive gain is a critical feature which enables the LC-CELP coder to achieve low complexity. In CELP, the excitation VQ codebook search is a major part of the total coder complexity. Such codebook search complexity grows exponentially with increasing vector dimension if the excitation encoding rate is fixed. Therefore, a smaller vector dimension gives lower codebook search complexity. However, conventional CELP coders typically use 4 to 5 bits to encode the gain of each excitation vector. Thus, if the vector dimension is too small, the gain encoding bit-rate will take too high a percentage of the total bit-rate, and this will leave an insufficient bit-rate for the excitation shape VQ to achieve good speech quality. By making the gain backward-adaptive in LC-CELP, we do not have to transmit any bit for the gains, and this enables us to use small vector dimension to reduce complexity while maintaining good speech quality.

2.5 Excitation VQ

After the backward-adaptive gain is determined, each 4-dimensional excitation vector is vector quantized into 6 bits using a closed-loop codebook search. Of the 6 bits, 1 bit is for the sign, and the other 5 bits specify one of 32 candidate codevectors which cover a wide range of gain level (there is no gain-shape structure). The 32 codevectors are closed-loop optimized using a large speech database and a codebook training algorithm similar to the one described in [7].

The LC-CELP excitation codebook search is similar to other CELP coders. The input speech vector is first weighted by the perceptual weighting filter. The response of the weighted LPC filter to the pitch-predicted vector and previously quantized excitation vectors are then subtracted from the weighted speech vector. Let $\mathbf{x}(n)$ be the resulting codebook search target vector at time n. Let $\sigma(n)$ be the backward-adapted excitation gain, $\mathbf{y}_j$ be the j-th codevector in the 5-bit shape codebook, and g_i be the sign multiplier corresponding to the sign bit i ($g_0 = +1$ and $g_1 = -1$). Also, let $\mathbf{H}(n)$ be the lower triangular Toeplitz matrix populated by the impulse response of the weighted LPC filter [8]. Then, the codebook search involves finding the best combination of indices i and j that minimizes the following distortion.

$$D_{ij}(n) = \sigma^2(n) \| \tilde{\mathbf{x}}(n) - g_i \mathbf{H}(n)\mathbf{y}_j \|^2 \ ,$$

where $\tilde{\mathbf{x}}(n) = \mathbf{x}(n)/\sigma(n)$ is the gain-normalized VQ target vector. Expanding the terms gives us

$$D_{ij}(n) = \sigma^2(n) \left[\| \tilde{\mathbf{x}}(n) \|^2 - 2g_i \tilde{\mathbf{x}}^T(n)\mathbf{H}(n)\mathbf{y}_j + g_i^2 \| \mathbf{H}(n)\mathbf{y}_j \|^2 \right] .$$

Since $g_i^2 = 1$ and $\| \tilde{\mathbf{x}}(n) \|^2$ and $\sigma^2(n)$ are fixed during the codebook search, minimizing $D_{ij}(n)$ is equivalent to minimizing

$$\hat{D}_{ij}(n) = -g_i \mathbf{p}^T(n)\mathbf{y}_j + E_j(n) \ ,$$

where

$$\mathbf{p}^T(n) = 2\tilde{\mathbf{x}}^T(n)\mathbf{H}(n) \ , \text{ and } E_j(n) = \| \mathbf{H}(n)\mathbf{y}_j \|^2 \ .$$

Note that $E_j(n)$ is the energy of the j-th filtered codevector and is fixed over each sub-frame. To reduce the complexity, we pre-compute and store the 32 possible $E_j(n)$ terms at the beginning of each sub-frame and then use them repeatedly for the codebook search of the 12 vectors within the sub-frame. For each of the 12 vectors, $\mathbf{p}(n)$ is computed first. Then, for each codevector $\mathbf{y}_j$, the inner product $\mathbf{p}^T(n)\mathbf{y}_j$ is computed, the sign multiplier g_i is chosen to have the same sign as $\mathbf{p}^T(n)\mathbf{y}_j$, and the corresponding distortion $\hat{D}_{ij}(n)$ is evaluated. The combination of i and j that gives the minimum distortion is the winner.

2.6 Summary of Bit Allocation

The bit allocation of the 16 kb/s LC-CELP coder is summarized in Table 1.

Coder Parameter	Bits per Sub-frame	Bits per Frame
LPC	11	44
Pitch period	7	28
Pitch taps	6	24
Excitation sign	1×12	48
Excitation codevector	5×12	240
Total	96	384

Table 1 Bit allocation of 16 kb/s LC-CELP

3. CODER PERFORMANCE

Table 2 lists the MOS scores obtained in two formal subjective listening tests. The coders in the table are listed in a descending order of MOS. In both tests, LC-CELP achieved higher MOS scores than the ITU-T (formerly CCITT) 32 kb/s ADPCM standard. LC-CELP is also found to be fairly robust to tandem coding with itself and transcoding with several other speech coders. Each additional stage of LC-CELP only degrades the MOS by an average of 0.14.

Condition	MOS (test 1)	MOS (test 2)
Source speech (16-bit PCM)	4.02	4.17
64 kb/s G.711 μ-law PCM	3.94	4.17
16 kb/s LC-CELP	3.92	4.09
32 kb/s G.721 ADPCM	3.81	4.07
8 kb/s IS54 (VSELP)	-	3.59
13 kb/s GSM full rate	3.47	-

Table 2 Mean Opinion Scores of LC-CELP and other coders

4. CODER COMPLEXITY

The 16 kb/s LC-CELP coder described above has been implemented on the AT&T DSP32C chip. A full-duplex LC-CELP coder takes 3.9 MIPS on this chip. At a clock rate of 50 MHz, or 12.5 MIPS, one DSP32C chip can implement three full-duplex LC-CELP coders. A 66 MHz DSP3210 chip should be able to implement four LC-CELP coders.

Further complexity reduction of LC-CELP without speech quality degradation is still possible. For example, we used to obtain the LPC predictor coefficients by Durbin's recursion, perform bandwidth expansion, convert the resulting predictor coefficients to reflection coefficients, and then quantize the reflection coefficients. Alternatively, we can eliminate bandwidth expansion and use spectral smoothing [3] to achieve a similar effect. In this case, we can use the LeRoux-Gueguen algorithm [9] to compute only the reflection coefficients (without computing the predictor coefficients) and have a complexity lower than Durbin's recursion. Furthermore, the resulting reflection coefficients are directly quantized without having to convert to predictor coefficients and back. Another possibility for complexity reduction is to use a perceptual weighting filter with weighting factors of 1.0 and 0.8. This gives a simpler weighted LPC filter. In fact, R. H. Ketchum has implemented such a simplified LC-CELP coder on a 24-bit fixed-point DSP. He used only 2.9 MIPS to implement a full-duplex LC-CELP coder. Five full-duplex LC-CELP coders can fit into a 33 MHz fixed-point DSP chip, and 6 coders can fit into a 40 MHz chip. We did not have a chance to test this simplified LC-CELP coder in any MOS test. However, informal listening tests showed that this simplified version maintained essentially the same speech quality as the original LC-CELP coder described earlier.

5. CONCLUSION

We have described a 16 kb/s Low-Complexity CELP coder. Features of this coder include a fast LPC quantization scheme, an efficient open-loop pitch search and pitch predictor tap VQ, a backward-adaptive excitation gain, and a trained excitation VQ codebook with a small vector dimension and a small codebook size. This work proved that toll-quality speech is achievable at 16 kb/s with a complexity as low as 3 MIPS.

ACKNOWLEDGMENT

The author would like to thank David O. Anderton and Richard H. Ketchum for doing the floating-point and fixed-point DSP implementations of LC-CELP, respectively.

References

1. B.S. Atal and M.R. Schroeder, "Stochastic coding of speech signals at very low bit rates," *Proc. IEEE Int. Conf. Communications*, p. 48.1, Amsterdam, The Netherlands (May 1984).
2. W. B. Kleijn, D. J. Krasinski, and R. H. Ketchum, "Fast methods for the CELP speech coding algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-38, 8, pp. 1330-1342 (August 1990).
3. Y. Tohkura, F. Itakura, and S. Hashimoto, "Spectral smoothing technique in PARCOR speech analysis-synthesis," *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-26, pp. 587-596 (December 1978).
4. J. D. Markel and A. H. Gray, Jr., *Linear prediction of speech*, Springer-Verlag, New York (1976).
5. J.-H. Chen, "A robust low-delay CELP speech coder at 16 kbit/s," *Proc. IEEE Global Comm. Conf.*, pp. 1237-1241, Dallas, Texas (November 1989).
6. J.-H. Chen, *Low-bit-rate predictive coding of speech waveforms based on vector quantization*, Ph. D. dissertation, University of California, Santa Barbara (March 1987).
7. J.-H. Chen, "High-quality 16 kb/s speech coding with a one-way delay less than 2 ms," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,*, pp. 453-456, Albuquerque, New Mexico (April 1990).
8. I.M. Trancoso and B.S. Atal, "Efficient procedures for finding the optimum innovation in stochastic coders," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 2375-2379 (April 1986).
9. J. LeRoux and C. Gueguen, "A fixed point computation of partial correlation coefficients," *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-27, pp. 257-259 (1979)