VERY LOW COMPLEXITY INTERPOLATIVE SPEECH CODING AT 1.2 TO 2.4 KBPS

Yair Shoham

Acoustics & Audio Communication Dept.
Bell Laboratories, Lucent Technologies
700 Mountain Ave.
Murray Hill, NJ 07974, USA
ys@research.bell-labs.com

ABSTRACT

The recently-introduced waveform interpolation (WI) coders [2, 4, 5, 6] provide good-quality speech at low rates but may be too complex for commercial use. This paper proposes new approaches to low-complexity WI speech coding at rates of 1.2 and 2.4 kbps. The proposed coders are 4 to 5 times faster than the previously reported ones . At 2.4 kbps, the complexity is about 7.5 and 2.5 MFLOPS for the encoder and decoder, respectively. At 1.2 kbps, the complexity is about 6 and 2.3 MFLOPS for the encoder and decoder, respectively. Informal subjective evaluation shows that, at 2.4 kbps, the quality is close to that of the high-complexity coders. The quality does not significantly degrade at 1.2 kbps and it is considered sufficient for messaging applications.

1. INTRODUCTION

In spite of a constant increase in telecommunication network capacity, interest in low-rate speech compression is still there for various current and future applications. Typical examples of such applications are paging services, answering machines, secure (military) communications and possibly, future wireless, and the internet. Significant progress has recently been achieved in low-rate speech coding with the introduction of new algorithms based on waveform interpolation (WI) and signal modeling techniques [2, 4]. However, the complexity of these WI coders, is currently too high to be commercially viable for low-cost terminals. This paper describes new approaches to low-complexity WI coding (LCWI) that utilize the advantages of interpolative coding but greatly simplify the analysis and synthesis operations.

The proposed LCWI coders are 4 to 5 times faster than the higher-complexity WI coders [4]. At 2.4 kbps, the complexity is about 7.5 and 2.5 MFLOPS for the encoder and decoder, respectively. At 1.2 kbps, the complexity is about 6 and 2.3 MFLOPS for the encoder and decoder, respectively. The small computational load of these coders make them suitable in multi-tasking environment for two-way digital speech communication that can be easily implemented on inexpensive DSP or host platforms.

Informal subjective evaluation shows that, at 2.4 kbps, the quality is close to that of the higher-complexity coders. The quality does not significantly degrade at 1.2 kbps and it is considered sufficient for messaging applications.

2. THE BASICS OF INTERPOLATIVE SPEECH CODING

The main techniques used in WI coding are waveform interpolation and signal decomposition [1, 2, 3, 4]. In the following, these are briefly reviewed and a high-level description of a WI coder is given.

2.1. Waveform interpolation

Waveform interpolation as defined in [1, 2, 3, 4] is based on the following signal representation. Let $S(t,K)$ be a DFT of a pitch-long snapshot taken at time t from a signal using a time-varying pitch period $P(t)$. A time-varying cycle function is defined as

$$c(t) \;=\; \int^{t} \frac{1}{P(\nu)} d\nu \qquad (1)$$

and a signal $s(t)$ is obtained by an inverse DFT (IDFT) of $S(t,K)$ with respect to $c(t)$:

$$s(t) \;=\; \sum_{K} S(t,K)\; \mathrm{e}^{j2\pi K c(t)} \qquad (2)$$

The spectrum $S(t,K)$ and the pitch function $P(t)$ are derived only at pre-determined update instances t_m and are interpolated at times $t_{m-1} < t < t_m$. The spectrum $S(t_m,K)$ of the current update is *phase-aligned* with $S(t_{m-1},K)$ for maximum correlation to provide for a smooth spectral evolution.

The WI method defined above works well for any signal, including non-periodic ones (with an arbitrary pitch), as long as the update intervals $t_m - t_{m-1}$ are less than half the pitch period. Therefore, no voicing classification is needed.

2.2. Signal decomposition

The WI encoder decomposes the speech signal for efficient compression. First, standard 10th-order linear prediction (LP) analysis is performed once per frame to obtain the

spectral envelope (LP) parameters and the LP residual signal. A sequence of aligned LP spectra is derived at times t_m as mentioned above. The spectra are also RMS-normalized to remove energy level variations. Each harmonic (each K) of the LP spectra $S(t_m, K)$ is now *temporally* filtered (along the time axis) by two complementary lowpass and highpass 20 Hz 20-tap filters. The lowpass and highpass filters generate two sequence of spectra representing slowly evolving (SEW) and rapidly evolving (REW) waveforms, respectively.

2.3. The basic WI coder

Figure 1 shows a high-level block diagram of a WI coder., At the encoder, pitch-long LP residual waveforms are extracted, DFT'ed, normalized and aligned. The resulting spectra are split into SEW's and REW's by temporal filtering. The WI encoder transmits the pitch and the quantized values of the LP parameters, the gain, the SEW and the REW spectra. The WI decoder reconstructs the LP envelope spectrum the gain, the REW and the SEW spectra. The SEW and the REW are combined to form the the quantized LP residual spectra. These are shaped and scaled by the LP and gain parameters and interpolated as describe above to form the coded speech signal. Table 1 shows typical frame and bit allocations for 1.2 kbps and 2.4 kbps WI coders.

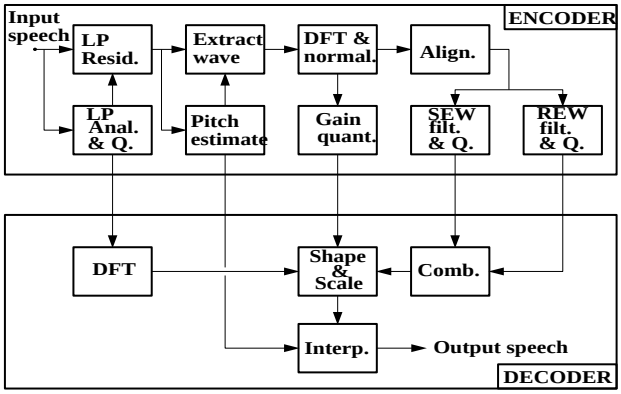

Figure 1. Basic waveform interpolation coder.

Rate (kpbs)	1.2	2.4
Frame (msec)	37.5	25.0
LPC	25	30
Pitch	7	7
SEW	0	7
REW	5	6
Gain	8	10

Table 1. Typical frame sizes and bit allocations for 1.2 and 2.4 kbps coders

3. LOW-COMPLEXITY WAVEFORM INTERPOLATION

The basic waveform interpolation process defined above is enormously complex due to the intense DFT operations and the non-regular sampling of the trigonometric functions, implied by (2). To reduce the complexity, the spectra $S(t_n, K)$ are augmented to a fixed radix-2 size by zero-padding and an IFFT is used. Also, a special low-complexity variant of *cubic cardinal spline* interpolation is used to approximate (2), as explained below.

The general spline representation [7, 8, 9] is defined as

$$s(t) \;=\; \sum_{n\,=-\infty}^{\infty} q_n \; B_k(t-n) \qquad (3)$$

where q_n are the spline coefficients and $B_k(t)$ is the spline continuous-time basis function, built of piecewise kth order polynomials, and has a small finite support of size $k+1$. The polynomials are determined by requiring the $0th$ to $(k-1)st$ order derivatives of $B_k(t)$ to be continuous at $t = n$. Interpolation by splines is done in two steps. First, the sampled data $s(n)$ is transformed to spline coefficients q_n. Next, q_n is used to interpolate $s(t)$.

Cardinal splines representation [7] is based on using the data $s(n)$ as the spline coefficient, namely, $s(n) = q_n$, so that no transform is needed. However, the support of the basis function is not local but is as long as the data, Also, additional conditions are imposed: $B_k(t) = 0$ for $t = n$ and $t \neq 0$. The *pseudo cardinal splines* are defined by forcing the basis function to have a *finite-support*. For a 3rd-order basis used here, the support is $-2 \leq t \leq 2$ and the condition $B_3(1) = B_3(-1) = 0$ has to be satisfied. To accommodate this condition, it can be shown that one continuity condition has to be given up. Therefore, the basis function is determined by letting the 2nd derivative have arbitrary values at the edges $t = -2$ and $t = 2$. Note that the basis function and its 1st derivative are zero at these points. Deriving the basis function under these conditions, the following interpolation formula is obtained in a matrix form:

$$s(d) = [\,d^3, d^2, d, 1\,] \begin{bmatrix} -0.75 & 1.25 & -1.25 & 0.75 \\ 1.50 & -2.25 & 1.50 & -0.75 \\ -0.75 & 0.00 & 0.75 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.00 \end{bmatrix} \begin{bmatrix} q_{n-1} \\ q_n \\ q_{n+1} \\ q_{n+2} \end{bmatrix} \qquad (4)$$

$$0 \;\leq\; d = t - n \;\leq 1$$

Setting $t = n$, one indeed gets $s(n) \;=\; q_n$ as expected.

The complete low-complexity interpolation proceeds as follows. The current spectrum is zero-padded to a fixed radix-2 size N and an IFFT is performed. This results in an N-vector $Q = \{q_n\}$. For each time t, the cycle function $Nc(t)$ points to a set of 4 coefficients in Q. These are interpolated with the corresponding coefficients from the previous update. Finally, (4) is used to get $s(t)$. Note that $Nc(t)$ increments modulo N to form a periodic signal consistent with the current pitch.

4. LOW-COMPLEXITY SIGNAL DECOMPOSITION

The spectral alignment and the SEW/REW filtering are complex operations that generate two high-resolution sequences of complex spectra to be coded and transmitted. These operations can be replaced by much simpler ones, based on the following considerations.

At 2.4 kbps and below, the bit budget is too small for a useful representation of this data since less than 0.1 bit per spectral component is available. Therefore, in low-rate WI coding, the phase spectrum is removed. REW is parametrized into only 5 parameters using polynomial curve fitting. The magnitude SEW is coded only at an 800 Hz baseband. The higher band is estimated assuming that $SEW(K) + REW(K) = 1$. This amounts to a gross under-representation of the LP spectrum, especially when the spectrum is taken over short frames. Since the signal quality is totally dominated by the quantizer, there is no point in doing a very expensive high-resolution analysis. Therefore, simpler signal decomposition and coding methods were adopted, using simpler notions of structured and unstructured signal components.

4.1. The unstructured component

The unstructured random-like component is defined as the difference between the properly aligned normalized current and previous spectra, referred to as the random spectrum (RS). The spectrum $RS(K)$ is smoothed by 2nd-order orthogonal polynomial expansion using 3 parameters per spectrum. Inspecting lots of smoothed SEW's and RS's has revealed that both spectra are almost always monotonically increasing with frequency. Typically, only 3 bits are available for coding the RS. Figure 2 shows a codebook of 8 normalized spectra that were found to best represent the RS at this bit rate. Coding the RS involves explicit orthogonal polynomial expansion analysis of the the full-resolution original spectrum [4]. However, the constellation of the 3-bit RS codebook leads to a simplified coding procedure. The curves in figure 2 are monotonically increasing with their indices. Therefore, they can be uniquely pointed to by knowing the areas under them, which is equivalent to their energies.

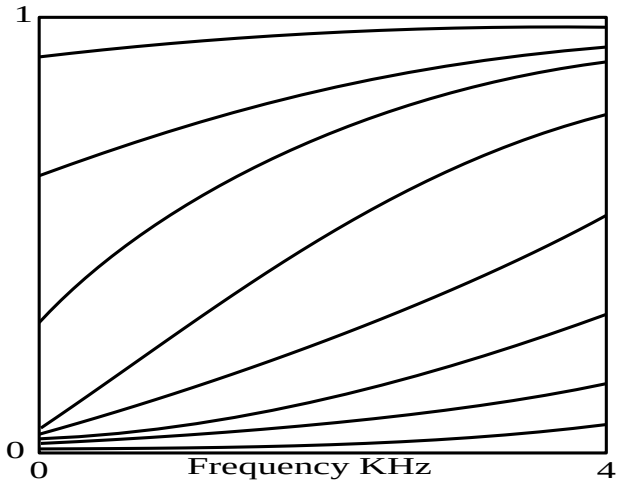

Figure 2. Eight smoothed spectra of the RS codebook.

It can be shown that, if the spectral update rate is close to the pitch frequency, the energy of the RS spectrum is closely proportional to a factor $u = 1 - C(P)$ where P is the pitch period, $C(.)$ is the standard correlation function and $C(P)$ is the pitch-lag correlation of the signal. The parameter u is used as an initial "soft index" to the codebook. Using a mapping table, u is mapped into an index in the range [0,7] which points to an RS curve. This approach has four major advantages from the encoder complexity standpoint. First, no explicit high-resolution RS need to be generated. Second, no alignment is needed. Third, no filtering is required. Fourth, no curve fitting is required. The parameter u defined above is derived at a fixed rate of 4 times per frame. It is efficiently quantized as a 4D vector by a 6-bit VQ. At the receiver, the decoded u-values are mapped into a set of orthogonal polynomial parameters and a smoothed RS spectrum is generated.

The decoded RS represents a magnitude spectrum. The complete complex RS is obtained by adding a random phase spectrum, consistent with the notion of an unstructured signal. The random phase is obtained by a random sampling of a phase table containing 128 2D vectors of radius 1.

4.2. The structured component

Given the pitch period P for the current frame, an extended frame containing an integral number M of pitch periods is determined. It usually overlaps the nominal frame. A pitch-size average spectrum $AS(K)$, is obtained by applying DFT to this frame, decimating the MP-size spectrum by the factor M and normalizing the result. To reduce the DFT complexity, the extended frame is first upsampled to a radix-2 size $N > MP$ and then an FFT is used. The upsampling is done using cubic spline interpolation as described in section 3. The structured spectrum $SS(K)$ is now defined by $SS(k) = AS(k) - RS(K)$. Note that no alignment and no filtering are used. As in the WI coder, only the *baseband* containing the lower 20% of the AS spectrum is coded. The rest of the AS magnitude spectrum is assumed flat with $AS(K) = 1$. The AS baseband is coded and transmitted once per frame using 10-dimensional 7-bit VQ of a variable dimension which is the lower of $D = 0.2 * P/2$ or $D = 10$. If $D < 10$, only the first D terms of the codevectors are used. At the receiver, the quantized AS baseband is interpolated at the synthesis update rate and the quantized $SS(K)$ spectrum is computed.

The magnitude spectrum $SS(K)$ represents a periodic signal. Therefore, a fixed phase spectrum, drawn from a male speaker, is attached to it to provide for some level of phase dispersion. The phase table has 64 complex values of radius 1. It is held in the same phase table used by the RS (the first 64 entries), so, no extra ROM locations are needed. The resulting complex SS is combined with complex RS to form the final quantized LP spectrum for the current update.

5. THE LCWI CODER

This section describes the overall LCWI coder as shown in figure 3 and provides more details not mentioned earlier.

At the encoder, LP analysis is applied to the input speech and a 10D vector of line spectral frequencies (LSF) is extracted once per frame. The LSF vector is quantized by a new fast classified VQ [10] that is 4 times faster than the VQ of earlier WI coders. An input speech gain vector is composed of 4 RMS's of pitch-size overlapping subframes

spaced uniformly within the frame. For the 2.4 kbps coder, the gain is expressed as a 4D *normalized* vector scaled by one "super-gain". The normalized vector is coded by a 6-bit VQ. The *log* of the super-gain is *differentially* coded by a 4-bit quantizer. In the 1.2 kbps coder a single 8-bit 4D VQ is applied to the log-gains. The AS spectrum and the u-coefficients are computed once and twice per frame, respectively. The 2D u-vector is VQ'ed using 5 and 6 bits for the 1.2 and 2.4 kbps coders, respectively. In the 2.4 kbps coder, the AS baseband is coded by 10D VQ using 7 bits. In the 1.2 kbps coder, the AS is *not* transmitted.

At the decoder, the gain and super-gain (if applicable) are decoded and combined into RMS values. The LP coefficient are decoded once per frame and an LP spectrum is obtained by applying DFT to the LP vector. The u-vector is decoded, mapped into an expansion parameter set and a smoothed magnitude RS is generated. A random phase is attached to it to generate the complex RS. The AS baseband is decoded and expanded and the SS magnitude spectrum is obtained by subtracting the RS. The SS phase is added to form the complex SS which is combined with the RS. The result is shaped by the LP spectrum, scaled by the gain and applied to the WI module, which outputs the coded speech. A mild post-filtering is applied to reshape the overall coding noise of the quantizers and the spline interpolator. All the encoder operations are executed at a rate determined by the update rate control (URC). The URC uses the received pitch to set the number of updates per frame in proportion to the pitch frequency. This balances the computational load over the pitch range.

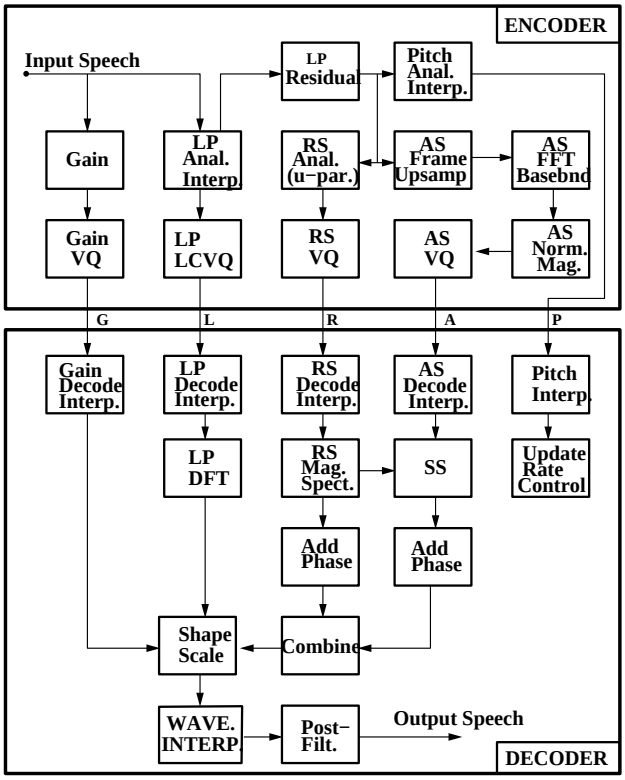

Figure 3. Main block diagram of the LCWI coder

6. SUMMARY

This paper describes low-complexity speech coders based on waveform interpolation. It focuses on new low-complexity methods for waveform interpolation and signal decomposition and provides a general description of the proposed coders for coding rates of 1.2 and 2.4 kbps. The LCWI coders are 4 to 5 times faster than the full-complexity WI coder [4]. At 2.4 kbps, the complexity is about 7.5 and 2.5 MFLOPS for the encoder and decoder, respectively. At 1.2 kbps, the complexity is about 6 and 2.3 MFLOPS for the encoder and decoder, respectively. The LCWI coders use about 1.5K words of RAM and about 11K words of ROM. The small computational load of these coders make them suitable for multi-tasking environment since they free the processor up to handle other tasks like networking. Informal subjective evaluation shows that, at 2.4 kbps, the quality is close to that of the full-complexity version. The quality does not significantly degrade at 1.2 kbps and it is considered sufficient for messaging applications.

REFERENCES

[1] Y. Shoham, *High-quality speech coding at 2.4 to 4.0 kbps based on time-frequency interpolation*, Proc. ICASSP'93 pp. II167-70.

[2] Y. Shoham, *High-quality speech coding at 2.4 kbps based on time-frequency interpolation*, Proc. Eurospeech'93, pp. 741-4.

[3] W.B. Kleijn, J. Haagen, *A speech coder based on decomposition of characteristic waveforms*, Proc. ICASSP'95 pp. 508-11.

[4] W.B. Kleijn, Y. Shoham, D. Sen, R. Hagen, *A low-complexity waveform interpolation coder*, Proc. ICASSP'96, pp. 212-5.

[5] I.S. Burnett, G.J. Bradley, *New techniques for multi-prototype waveform coding at 2.84 kbps*, Proc. ICASSP'95, pp. 261-264.

[6] D.H. Pham, I.S. Burnett, *Quantisation techniques for prototype waveforms*, Proc. Int. Symp. Sig. Proces. and App., ISSPA, Aug. 1996, pp. 53-56.

[7] M. Unser, A. Aldroubi, M. Eden, *B-Spline Signal Processing: Part I - Theory*, IEEE Trans. on Sig. Proc. Vol. 41, No. 2, Feb. 1993.

[8] M. Unser, A. Aldroubi, M. Eden, *B-Spline Signal Processing: Part II - Efficient Design*, IEEE Trans. on Sig. Proc. Vol. 41, No. 2, Feb. 1993.

[9] H. Hou, H. C. Andrews, *Cubic Splines for Image Interpolation and Digital filtering*, IEEE Trans, on Acoust. Sp. & Sig. Proc. Vol. ASSP-26, NO. 6, Dec. 1978.

[10] J. Zhou, Y. Shoham, A. Akansu, *Simple fast vector quantization of the line spectral frequencies*, Proc. ICSLP'96, Vol. 2, pp. 945-8, Oct. 1996. (also available on CDROM).

[11] J.H. Chen A. Gersho, *Adaptive postfiltering for quality enhancement of coded speech*, IEEE Trans. Speech and Audio Processing, Vol. 3, 1995, pp. 59-71.