A High-Quality Speech and Audio Codec With Less Than 10 ms Delay

Jean-Marc Valin, *Member, IEEE*, Timothy B. Terriberry, Christopher Montgomery, Gregory Maxwell

***Abstract*—With increasing quality requirements for multimedia communications, audio codecs must maintain both high quality and low delay. Typically, audio codecs offer either low delay or high quality, but rarely both. We propose a codec that simultaneously addresses both these requirements, with a delay of only 8.7 ms at 44.1 kHz. It uses gain-shape algebraic vector quantisation in the frequency domain with time-domain pitch prediction. We demonstrate that the proposed codec operating at 48 kbit/s and 64 kbit/s out-performs both G.722.1C and MP3 and has quality comparable to AAC-LD, despite having less than one fourth of the algorithmic delay of these codecs.**

EDICS Category: SPE-CODI, AUD-ACOD

***Index Terms*—audio coding, speech coding, super-wideband, low-delay, transform coding**

I. Introduction

Meeting increasing expectations for video-conferencing and other communication applications requires a high-quality, very low delay speech and audio codec. Decreasing the delay both reduces the perception of acoustic echo and enables new applications, such as remote music performances [1]. Popular speech codecs such as AMR-WB [2], G.729.1 [3], and Speex [4] have a low-to-medium quality range, do not support sampling rates above 16 kHz, and have total algorithmic delays ranging from 15 ms to 30 ms. On the other hand, commonly used audio codecs, such as MP3 and Vorbis [5], can achieve high quality but have delays exceeding 100 ms. None of these codecs provide both high quality and very low delay.

Since Code-Excited Linear Prediction (CELP) [6] was proposed in the 1980s, it has been the most popular class of speech coding algorithms. It is, however, generally limited to sampling rates below 16 kHz. In the authors' experience and as reported in [2], CELP's noise shaping is difficult to control when the spectrum has high dynamic range, as is common for sampling rates of 16 kHz and above. To the authors' knowledge, CELP has not been applied to speech codecs beyond a 16 kHz sampling rate. Even at 16 kHz, many CELP-based codecs do not use CELP for the entire audio band. AMR-WB applies CELP on a down-sampled 12.8 kHz signal, while G.729.1 uses an MDCT for frequencies above 4 kHz [3], and Speex encodes 16 kHz speech using a Quadrature Mirror Filter and two CELP encoders [4].

Unlike speech codecs, most audio codecs designed for music now use the Modified Discrete Cosine Transform (MDCT). The algorithmic delay of an MDCT-based codec is equal to the length of the window it uses. Unfortunately, the very short time window required to achieve delays below 10 ms does not give the MDCT sufficient frequency resolution to model pitch harmonics. Most codecs based on the MDCT use windows of 50 ms or more, although there are exceptions, such as G.722.1C and AAC-LD, which use shorter windows.

We propose a new algorithm called the Constrained-Energy Lapped Transform (CELT), detailed in Sections II and III, that uses the MDCT with very short windows. It explicitly encodes the energy of each spectral band, constraining the output to match the spectral envelope of the input, thus preserving its general perceptual qualities. It incorporates a time-domain pitch predictor using the past of the synthesis signal to model the closely-spaced harmonics of speech, giving both low delay and high resolution for harmonic signals, just like speech codecs [6]. For non-harmonic signals, the energy constraints prevent the predictor from distorting the envelope of the signal, and it acts as another vector quantisation codebook that only uses a few bits. The codec has the following characteristics:

- a 44.1 kHz sampling rate,
- a 8.7 ms algorithmic delay (5.8 ms frame size with 2.9 ms look-ahead),
- high quality speech around 48 kbit/s, and
- good quality music around 64 kbit/s.

We give the results of a number of experiments in Section IV. We performed subjective listening tests against several other codecs, and found CELT to equal or out-perform both G.722.1C and AAC-LD, while achieving significantly less delay than all of them. We also performed an objective analysis of the effect of transmission errors and found CELT to be robust to random packet loss rates up to 5% and bit error rates (BER) as high as 3×10^{-4} (0.03%).

II. Constrained-Energy Lapped Transform

One of the key issues with MDCT-based codecs is the time-frequency resolution. For example, a codec proposed in [7] uses a 35 ms window (17.5 ms frames) to achieve sufficient frequency resolution to resolve the fine structure of pitch harmonics in speech. CELT's very low delay constraint implies that it must use a very short MDCT and hence has poor frequency resolution. To mitigate the problem, we use a long-term predictor that extends far enough in the past to model an entire pitch period.

Another issue with using very short frames is that only a very small number of bits is available for each frame.

Jean-Marc Valin is with Octasic Inc., Montreal, Canada (part of the work was performed at the CSIRO ICT Centre, Marsfield, NSW, Australia). Christopher Montgomery is with RedHat Inc., USA. Gregory Maxwell is with Juniper Networks Inc., USA. All authors are with the Xiph.Org Foundation (email: jmvalin@ieee.org; tterribe@xiph.org; xiphmont@xiph.org; greg@xiph.org).
The authors would like to thank all the volunteers who participated in the listening tests.

CELT must limit or eliminate meta-information, such as that signalling bit allocation, and will usually have just a few bits available for some frequency bands. For that reason, we separate the coding of the spectral envelope from the coding of the details of the spectrum. This ensures that the energy in each frequency band is always preserved, even if the details of the spectrum are lost.

CELT is inspired by CELP [6], using the idea of a spectrally flat "excitation" that is the sum of an adaptive (pitch) codebook and a fixed (innovation) codebook. The excitation represents the details of the spectrum after the spectral envelope has been removed. However, unlike CELP, CELT mainly operates in the frequency domain using the Modified Discrete Cosine Transform (MDCT), so the excitation in CELT is a frequency-domain version of the excitation in CELP. Similarly, the adaptive codebook is based on a time offset into the past with an associated set of gains, and the innovation is the part of the excitation that is not predicted by the adaptive codebook.

The main principles of the CELT algorithm are that

- the MDCT output is split in bands approximating the critical bands;
- the encoder explicitly codes the energy in each band (spectral envelope) and the decoder ensures the energy of the output matches the coded energy exactly;
- the normalised spectrum in each band, which we call the excitation, is constrained to have unit norm throughout the process; and
- the long-term (pitch) predictor is encoded as a time offset, but with a pitch gain encoded in the frequency domain.

A block diagram of the CELT algorithm is shown in Fig. 1. The bit-stream is composed of 4 sets of parameters: the energy in each band, the pitch period, the pitch gains, and the innovation codewords. The most important variables are defined in Fig. 2.

The signal is divided into 256 sample frames, with each MDCT window composed of two frames. To reduce the delay, the overlap is only 128 samples, with a 128-sample constant region in the centre and 64 zeros on each side, as shown in Fig. 3. For the overlap region, we use the Vorbis [5] codec's power-complementary window:

$$w\left(n\right) = \sin\left[\frac{\pi}{2}\sin^2\left(\frac{\pi\left(n+\frac{1}{2}\right)}{2L}\right)\right] \ , \tag{1}$$

where $L = 128$ is the amount of overlap. Although a critically sampled MDCT requires a window that is twice the frame size we reduce the "effective overlap" with the zeros on each side and still achieve perfect reconstruction. This reduces the total algorithmic delay with very little cost in quality or bit-rate. We use the same window for the analysis process and the weighted overlap-and-add (WOLA) synthesis process.

A. Bands and Energy

CELT exploits the fact that the ear is mainly sensitive to the amount of energy in each critical band. The MDCT spectrum is thus divided into 20 bands of roughly one critical band each, although the lower frequency bands are wider due to the low MDCT resolution. We refer to these bands as the *energy bands*.

α	inter-frame energy prediction coefficient
β	inter-band energy prediction coefficient
μ	mean energy in a band (fixed, computed offline)
b	band index, loosely following the critical bands
E	energy in band b for frame ℓ (alternatively E_{dB} in dB scale)
$\tilde{E}$	quantised energy (alternatively $\tilde{E}_{dB}$ in dB scale)
g_a	adaptive codebook gain
$\tilde{g}_a$	quantised adaptive codebook gain
g_f	fixed (innovation) codebook gain
J	cost function for the innovation search
K	number of pulses assigned to a band
ℓ	frame index
L	length of the overlap (where the window is neither one nor zero)
N	number of MDCT samples in a band
n_k	position of the n^{th} innovation pulse
$\mathbf{p}$	normalised adaptive code vector (pitch or folding)
$\mathbf{r}$	residual after prediction (unquantised innovation)
S	coarse energy quantiser resolution (6 dB)
s_k	sign of the k^{th} innovation pulse
T	pitch period: time offset used for the long-term predictor
V	number of pulse combinations
w	window function
$\mathbf{x}$	excitation: MDCT coefficients after normalisation
$\tilde{\mathbf{x}}$	quantised excitation
$\mathbf{y}$	quantised innovation
$\mathbf{z}$	MDCT coefficients
$\tilde{\mathbf{z}}$	quantised MDCT coefficients

Figure 2. Summary of variable definitions. Many of these variables have indices b and ℓ, which are often omitted for clarity.

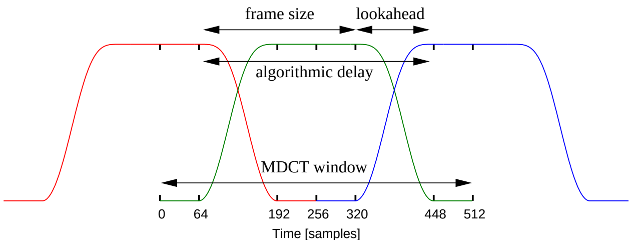


Figure 3. Power-complementary windows with reduced overlap. The frame size is 256 samples, with 128 samples overlap. The total algorithmic delay is 384 samples.

We normalise the MDCT spectrum in each band and transmit the energy separately. Let $\mathbf{z}_b\left(\ell\right)$ be the MDCT spectrum in band b at time frame ℓ. Then the normalised excitation in band b is

$$\mathbf{x}_b\left(\ell\right) = \frac{\mathbf{z}_b\left(\ell\right)}{\sqrt{E\left(b,\ell\right)}} \ , \tag{2}$$

where $E\left(b,\ell\right) = \mathbf{z}_b^T\left(\ell\right)\mathbf{z}_b\left(\ell\right)$ is the energy in band b, so that $\mathbf{x}_b^T\left(\ell\right)\mathbf{x}_b\left(\ell\right) = 1$.

A quantised version of the energy and the spectrum in each band is transmitted to the decoder so that the signal can be

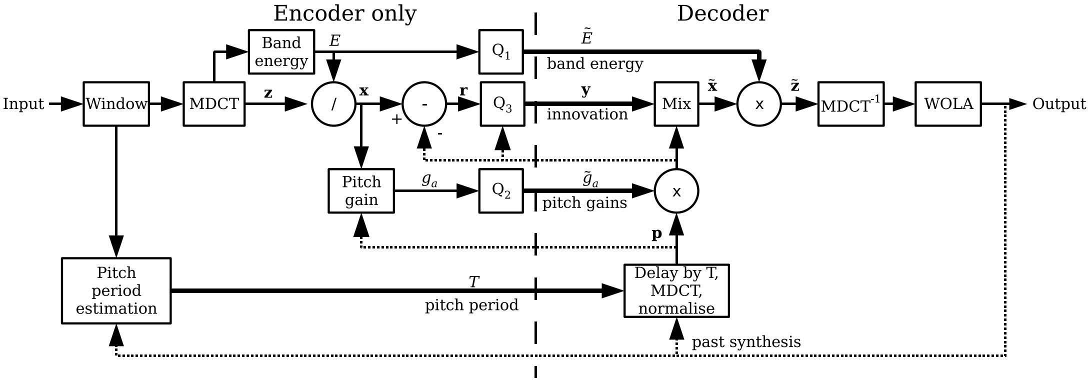


Figure 1. Overview of the CELT algorithm. The complete encoder includes the decoder part because the encoding process refers to previously decoded portions of the synthesis signal. Parameters transmitted to the decoder are shown in bold and the parameters that are synchronised between the encoder and the decoder are shown with dotted lines. Quantisers are denoted by the Q_x operators.

recovered using

$$\tilde{\mathbf{z}}_b = \sqrt{\tilde{E}\left(b,\ell\right)}\tilde{\mathbf{x}}_b\left(\ell\right) \ , \tag{3}$$

where the quantised excitation $\tilde{\mathbf{x}}_b\left(\ell\right)$ still obeys $\tilde{\mathbf{x}}_b^T\left(\ell\right)\tilde{\mathbf{x}}_b\left(\ell\right) = 1$. This gain-shape approach has the advantage of preserving the spectral envelope regardless of the bit-rate used to encode the "details" of the spectrum. It also means that the spectral envelope $E\left(b,\ell\right)$ must be encoded at sufficient resolution, since $\tilde{\mathbf{x}}_b\left(\ell\right)$ cannot compensate for the quantisation error in $\tilde{E}\left(b,\ell\right)$. This is unlike CELP codecs, where increasing the bit-rate of the excitation can partially compensate for quantisation error in the LP coefficients.

From here on, unless we are processing multiple bands or multiple frames at once, the frequency band b and time frame ℓ are omitted for clarity.

B. Pitch Prediction

CELT uses pitch prediction to model the closely-spaced harmonics of speech, solo instruments, or other highly periodic signals. By itself, our short block transform is only capable of resolving harmonics if the period is an exact multiple of the frame size. For any other period length, the current window will contain a portion of the period offset by some phase. We search the recently decoded signal data for a window that covers the same portion of the period with the same phase offset. While the harmonics will still not resolve into distinct MDCT bins, for periodic inputs the predictor will produce the same pattern of energy spreading.

The pitch predictor is specified by a period defined in the time domain and a set of gains defined in the frequency domain. The pitch period is the time offset to the window in the recent synthesis signal history that best matches the current encoding window. We estimate the period using the frequency-domain generalised cross-correlation between the zero-padded input window and the last $L_p = 1024$ decoded samples [8]. We use a weight function to normalise the response at each frequency by the magnitude of the input window's spectrum, which is a crude substitute for the perceptual weight CELP uses when computing time-domain cross-correlation. Because the delayed signal used for pitch cannot overlap with the current frame, the minimum delay possible is $N + L$ (384 samples). This corresponds to a 115 Hz fundamental, meaning that for female speakers the estimated period is usually a multiple of the real pitch period. Since the maximum period is equal to L_p, there are $L_p - N - L + 1 = 641$ possible time offsets.

Given the period, we compute the MDCT on the windowed, delayed synthesis signal and normalise it to have unit magnitude in each band. We apply the gain to the normalised signal, $\mathbf{p}$, in the frequency domain, allowing us to vary the gain as a function of frequency. We compute the gain in each band between 0 and 8 kHz as

$$g_a = g_{damp}\frac{\mathbf{x}^T\mathbf{p}}{\mathbf{p}^T\mathbf{p}} = g_{damp}\mathbf{x}^T\mathbf{p} \ , \tag{4}$$

where g_{damp} is the gain *damping factor* that also acts as an upper bound for the gain, since $\mathbf{x}^T\mathbf{p} \leq 1$. Above 8 kHz, the adaptive codebook uses spectral folding from the current frame, as described in Section III-C3. Because both $\mathbf{x}$ and $\mathbf{p}$ are normalised, the optimal gain may never exceed unity. This is unlike the CELP algorithm, where the optimal pitch gain may be greater than unity during onsets, as mentioned by [2], which limits the gain to 1.2 to prevent unstable behaviour. We limit the gain to $g_{damp} = 0.9$ to improve robustness to packet loss, not to avoid instability.

We apply the pitch gain in the frequency domain to account for the weakening of the pitch harmonics as the frequency increases. While a 3-tap time-domain pitch gain [9] works with an 8 kHz signal, it does not allow sufficient control of the pitch gain as a function of frequency for a 44.1 kHz signal.

C. Innovation

In a manner similar to the CELP algorithm, the adaptive codebook and fixed codebook contributions for a certain frequency band are combined with

$$\widetilde{\mathbf{x}} = \tilde{g}_a\mathbf{p} + g_f\mathbf{y} \ , \tag{5}$$

where $\tilde{g}_a$ is the quantised gain of the adaptive codebook contribution $\mathbf{p}$ and g_f is the gain for the fixed codebook contribution $\mathbf{y}$. Unlike CELP, the fixed codebook gain g_f does not need to be transmitted. Because of the constraint $\tilde{\mathbf{x}}_b^T\tilde{\mathbf{x}}_b = 1$ and knowing that $\mathbf{p}^T\mathbf{p} = 1$, the fixed codebook gain can be computed as

$$g_f = \frac{\sqrt{\tilde{g}_a^2\left(\mathbf{y}^T\mathbf{p}\right)^2 + \mathbf{y}^T\mathbf{y}\left(1-\tilde{g}_a^2\right)} - g_a\mathbf{y}^T\mathbf{p}}{\mathbf{y}^T\mathbf{y}} . \tag{6}$$

If $\tilde{g}_a = 0$, then (6) simplifies to $g_f = 1/\sqrt{\mathbf{y}^T\mathbf{y}}$, which only ensures that $g_f\mathbf{y}$ has unit norm.

III. Quantisation

This section describes each of the quantisers used in CELT. As shown in Fig. 1, we use three different quantisers: one for the band energies, one for the pitch gains, and one for the innovation. We entropy code the quantised results with a range coder [10], a type of arithmetic coder that outputs eight bits at a time. For other quantisers, entropy coding is not necessary, but we still use the range coder because it allows us to allocate a fractional number of bits to integers whose size is not a power of two. For example, using a range coder we can encode an integer parameter ranging from 0 to 2 using three symbols of probability 1/3. This requires only $\log_2 3 \approx 1.59$ bits instead of the 2 bits necessary to encode the integer directly.

A. Band Energy Quantisation (Q_1)

Efficiently encoding the energies $E(\ell, b)$ requires eliminating redundancy in both time and frequency domain. Let $E_{dB}(\ell, b)$ be the log-energy in band b at time frame ℓ We quantise this energy as

$$q_b(\ell) = \left\langle \frac{E_{dB}(\ell,b) - \mu_b - \alpha\tilde{E}_{dB}(\ell-1,b) - D(\ell,b)}{S} \right\rangle , \tag{7}$$

$$\tilde{E}_{dB}(\ell,b) = S\left(q_b(\ell) + \mu_b + \alpha\tilde{E}_{dB}(\ell-1,b) + D(\ell,b)\right) , \tag{8}$$

$$D(\ell+1,b) = D(\ell,b) + \mu_b + (1-\beta)\,Sq_b(\ell) \ , \tag{9}$$

where $\langle\cdot\rangle$ denotes rounding to the nearest integer, $q_b(\ell)$ is the encoded symbol, μ_b is the mean energy for band b (computed offline), S is the quantisation resolution in dB, α controls the prediction across frames, and β controls the prediction across bands. Not taking into account the fact that the prediction in (7) is based on the quantised energy, the 2-D z-transform of the prediction filter is

$$A(z_\ell, z_b) = \left(1 - \alpha z_\ell^{-1}\right) \cdot \frac{1 - z_b^{-1}}{1 - \beta z_b^{-1}} . \tag{10}$$

To find the optimal values for α and β, we measure the entropy in the prediction error prior to encoding. Fig. 4 shows that prediction can reduce the entropy by up to 33 bits per frame. The use of inter-frame prediction results in a reduction of 12 bits compared to coding frames independently. Based on Fig. 4, we select $\alpha = 0.8$, for which $\beta = 0.6$ is optimal. This provides close to optimal performance while being more robust to packet loss than higher values of α.

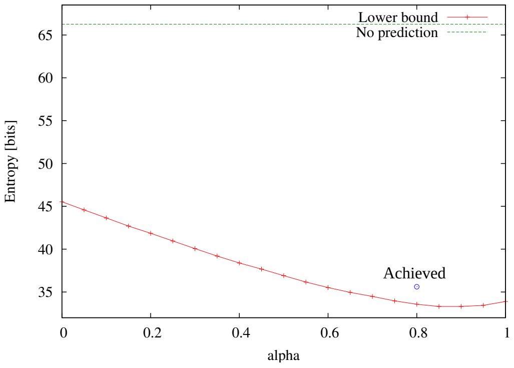


Figure 4. Entropy of the energy prediction error (quantised with 6 dB resolution) as a function of the inter-frame prediction coefficient α using the corresponding optimal value of β. The *lower bound* curve is a measurement of the entropy based on the probabilities measured on the same data.

We have found experimentally that the distribution of the prediction error q is close to a generalised Gaussian distribution of the form

$$N(E_{dB}) \propto e^{-\left|\frac{E-\mu}{\sigma}\right|^{\gamma}} , \tag{11}$$

with $\gamma \simeq 1.5$. While using (11) directly in the entropy coder would result in the minimal average bit-rate for encoding the energy, we prefer to use a Laplace distribution. By overestimating the least probable values, the Laplace distribution yields a more constant bit-rate over time, with very little penalty to the average bit-rate. The achieved rate is only 2 bits above the lower bound. Exponential-Golomb codes [11] could encode the Laplace-distributed variables with little penalty, but since the innovation (Q_3) requires a range coder, we re-use that here.

Once the energy is quantised and encoded at a coarse 6 dB resolution, a finer scalar quantisation step (with equiprobable symbols) is applied to achieve a variable resolution that depends on the frequency and the bit-rate. We use this coarse-fine quantisation process for two reasons. First, it ensures that most of the information is encoded in a few bits, which can easily be protected from transmission errors. Second, it allows us to adjust the fine quantisation of the energy information to control the total rate allocated to the energy. We have determined empirically that best results are obtained when the energy encoding uses about 1/5 of the total allocated bits.

B. Pitch Gain Quantisation (Q_2)

As described in Section II-B, we compute pitch gains for each band below 8 kHz. While there is a large correlation between the gains, simple prediction as used for Q_1 is insufficient. Instead, the gains are vector-quantised using a 128-entry codebook (7 bits). To limit the size of the codebook, only 8 values per entry are considered, so some adjacent bands are forced to have the same quantised gain. Using 8 bits for each value, the codebook requires only 1024 bytes of storage. When a special entry in the codebook composed of all zeros is used, the pitch period does not need to be encoded. Because the pitch gain is more sensitive to errors when its value is close to one, we optimise the codebook in the *warped* domain:

$$g_a^{(w)} = 1 - \sqrt{1 - g_a^2}\ . \tag{12}$$

The codebook is trained and stored in the warped domain $g_a^{(w)}$ so that we can search the codebook using the Euclidean distance metric. Once the quantised warped gains $\tilde{g}_a^{(w)}$ are found, the quantised gains are *unwarped* by

$$\tilde{g}_a = \sqrt{1 - \left(1 - \tilde{g}_a^{(w)}\right)^2}\ .$$

C. Innovation Quantisation (Q_3)

Because of the normalisation used in CELT, the innovation data lies on the surface of a hypersphere. While no optimal tessellation is known for a hypersphere in an arbitrary number of dimensions, a good approximation is a unit pulse codebook where a code vector $\mathbf{y}$ with K pulses is constructed as

$$\mathbf{y} = \sum_{k=1}^{K} s^{(k)} \varepsilon_{n^{(k)}}\ , \tag{13}$$

where $n^{(k)}$ and $s^{(k)}$ are the position and sign of the k^{th} pulse, respectively, and $\varepsilon_{n^{(k)}}$ is the $n^{(k)}$th elementary basis vector. The signs s_k are constrained such that $n^{(j)} = n^{(k)}$ implies $s^{(j)} = s^{(k)}$.

We search the codebook by minimising the square error between the residual $\mathbf{r} = \mathbf{x} - g_a\mathbf{p}$ and $\mathbf{y}$:

$$\begin{aligned}
\mathbf{y} =& \underset{\mathbf{y}}{\operatorname{argmin}} \left\|\mathbf{r} - g_f\mathbf{y}\right\|^2\ , & (14)\\
=& \underset{\mathbf{y}}{\operatorname{argmin}} \left(\mathbf{r}^T\mathbf{r} - 2g_f\mathbf{r}^T\mathbf{y} + g_f^2\mathbf{y}^T\mathbf{y}\right)\ , & (15)\\
=& \underset{\mathbf{y}}{\operatorname{argmin}} \left(\mathbf{r}^T\mathbf{r} + J\right)\ , & (16)\\
J =& -2g_f\mathbf{r}^T\mathbf{y} + g_f^2\mathbf{y}^T\mathbf{y}\ . & (17)
\end{aligned}$$

We only need to calculate J. The constant term, $\mathbf{r}^T\mathbf{r}$, can be omitted.

We perform the search one pulse at a time, constraining the sign to match that of $\mathbf{r}$ at each pulse position. Assuming that we have already selected $(k-1)$ pulses, we choose the next pulse position $n^{(k)}$ by optimising (15). Let $R_{yp}^{(k)} = \mathbf{p}^T\mathbf{y}^{(k)}$ with $\mathbf{y}^{(k)}$ containing k pulses and define $R_{ry}^{(k)}$ and $R_{yy}^{(k)}$ similarly. Then the corresponding $J^{(k)}$ can be computed efficiently for each new pulse using the recursion

$$\begin{aligned}
s^{(k)} =& \operatorname{sign}\left(r_{n^{(k)}}\right)\ , & (18)\\
R_{yp}^{(k)} =& R_{yp}^{(k-1)} + s^{(k)}p_{n^{(k)}}\ , & (19)\\
R_{ry}^{(k)} =& R_{ry}^{(k-1)} + s^{(k)}r_{n^{(k)}}\ , & (20)\\
R_{yy}^{(k)} =& R_{yy}^{(k-1)} + 2s^{(k)}y_{n^{(k)}}^{(k-1)} + 1\ , & (21)\\
g_f^{(k)} =& \frac{\sqrt{\tilde{g}_a^2\left(R_{yp}^{(k)}\right)^2 + R_{yy}^{(k)}\left(1 - g_a^2\right)} - \tilde{g}_a R_{yp}^{(k)}}{R_{yy}^{(k)}}\ , & (22)\\
J^{(k)} =& -2g_f R_{ry}^{(k)} + g_f^2 R_{yy}^{(k)}\ . & (23)
\end{aligned}$$

Again, if we have $\tilde{g}_a = 0$, then the cost function simplifies to the standard cost function $J = -\mathbf{r}^T\mathbf{y}/\sqrt{\mathbf{y}^T\mathbf{y}}$. Failing to take into account the pitch gain and using the standard cost function yields poorer performance than not using a pitch predictor at all, since a small error in the fixed codebook contribution may result in a large final error after (6) is applied if the adaptive codebook contribution is large.

The complexity of the search described above is $\mathcal{O}(KN)$. Assuming that the number of bits it takes to encode K pulses is proportional to $K\log_2 N$[1], we can rewrite the complexity of searching a codebook with b bits as $\mathcal{O}(bN/\log_2 N)$. By comparison, the complexity involved in searching a stochastic codebook with b bits is $\mathcal{O}\left(2^b N\right)$, which is significantly higher.

While the structure of the pulse codebook we use has similarities with ACELP [12], the search in CELT is direct and does not involve filtering operations. On the other hand, the cost function is more complex, since the fixed codebook gain depends on both the pitch gain and the code vector selected.

1) Reduced search complexity: For large codebooks, the complexity of the search procedure described above can be high. We adopt two strategies to reduce that complexity:

- selecting more than one pulse at a time when $K \gg N$, and
- using the simpler cost function $J = -\mathbf{r}^T\mathbf{y}/\sqrt{\mathbf{y}^T\mathbf{y}}$ for all but the last pulse.

When the number of pulses is large compared to the number of samples in a band, we can have a large number of pulses in each position – in some cases, up to 64 pulses in only 3 positions. Clearly, when starting the search, there is little risk in assigning more than one pulse to the position that minimises the cost function in (17). Therefore in each step we assign

$$n_p = \max\left(\left\lfloor\left(K - k_a\right)/N\right\rfloor, 1\right) \tag{24}$$

pulses, where k_a is the number of pulses that have already been assigned and $\lfloor\cdot\rfloor$ denotes truncation towards zero.

Although using $J = -\mathbf{r}^T\mathbf{y}/\sqrt{\mathbf{y}^T\mathbf{y}}$ as the cost function reduces quality, it is possible to find most pulses with it and to use the correct cost function only when placing the last pulse. This results in a speed gain without any significant quality degradation.

2) Pulse vector encoding: The pulse vector $\mathbf{y}$ found for each band needs to be encoded in the bit-stream. We assign a unique index to each possible $\mathbf{y}$ by recursively partitioning the codebook one pulse position at a time [13]. For K pulses in N samples, the number of codebook entries is

$$\begin{aligned}
V(N,K) = & V(N-1,K)\\
& + V(N,K-1) + V(N-1,K-1)\ , \qquad (25)
\end{aligned}$$

with $V(N,0) = 1$ and $V(0,K) = 0,\ K \neq 0$. The factorial pulse coding (FPC) method [14] also achieves a one-to-one and onto map from pulse vectors to an index less than $V(N,K)$, and using FPC would produce bit-identical decoded output, even though the compressed bit-stream would differ. The main advantage of the index assignment method used in CELT is that it does not require multiplications or divisions, and can be implemented without a lookup table.

The size of a codeword, $\log_2 V(N,K)$, is generally not an integer. To avoid rounding up the the next integer and wasting

[1] As we will see in Section III-C2, this is only an approximation.

an average of half a bit per band (10 bits per frame), we encode the integers using the range coder. We use equiprobable symbols, so the encoded size is perfectly predictable. The total overhead of this method is at most one bit for all bands combined [15], as has been observed in practice. We have also found that the loss due to the use of equiprobable symbols (as opposed to using the measured probability of each symbol) for encoding the innovation is negligible, as one would expect for a well-tuned vector quantisation codebook. For the complete codec operating at 64 kbit/s, we have measured that pulse vector coding results in a saving of 10.8 kbit/s when compared to encoding each scalar y_k value independently using an optimal entropy coder.

3) Sparseness prevention: The lack of pitch prediction above 8 kHz and the small number of pulses used at these frequencies yields a sparsely quantised spectrum, with few non-zero values. This causes the “birdie” artifacts commonly found in low bit-rate MPEG-1 Layer 3 (MP3) encodings. To mitigate this, we use a folded copy of the lower frequency spectrum for the adaptive code vector, $\mathbf{p}$. We encode a sign bit to allow the spectrum to be inverted. The principle is similar to [16], but is applied in the MDCT domain. The gain $\tilde{g}_a$ of this adaptive codebook is pre-determined, and depends only on the number of pulses being used and the width of the band. It is given by

$$\tilde{g}_a = \frac{N}{N + \delta K} \tag{26}$$

where $\delta = 6$ has been found to provide an acceptable compromise between avoiding birdies and the harshness that can result from spectral folding.

D. Bit allocation

Two of the parameter sets transmitted to the decoder are encoded at variable rate: the energy in each band, which is entropy coded, and the pitch period, which is not transmitted if the pitch gains are all zero. To achieve a constant bit-rate without a bit reservoir, we must adapt the rate of the innovation quantisation. Since CELT frames are very short, we need to minimise the amount of side information required to transmit the bit allocation. Hence we do not transmit this information explicitly, but rather infer it solely from the information shared between the encoder and the decoder. We first assume that both the encoder and the decoder know how many 8-bit bytes are used to encode the frame. This number is either agreed on when establishing the communication or obtained during the communication, e.g. the decoder knows the size of any UDP datagram it receives. Given that, both the encoder and the decoder can implement the same mechanism to determine the innovation bit allocation.

This mechanism is based solely on the number of bits remaining after encoding the energy and pitch parameters. A static table determines the bit-allocation in each band given only the number of bits available for quantising the innovation. The correspondence between the number of bits in a band and the number of pulses is given by (25). For a given number of innovation bits, the distribution across the bands is constant in time. This is equivalent to using a psychoacoustic masking curve that follows the energy in each band. Because the bands have a width of one Bark, the result models the masking occuring within each critical band, but not the masking across critical bands. Using this technique, no side information is required to transmit the bit allocation.

Table I
AVERAGE BIT ALLOCATION FOR EACH FRAME. FRACTIONAL BITS ARE DUE TO THE HIGH RESOLUTION OF THE ENTROPY CODER AND TO THE AVERAGING OVER ALL FRAMES.

	Constant bit-rate	
Parameter	46.9 kbit/s	64.8 kbit/s
Coarse energy (Q_1)	36.0	35.9
Fine energy (Q_1)	38.1	59.2
Pitch gain (Q_2)	7.0	7.0
Pitch period	8.9	8.9
Innovation (Q_3)	180.9	264.0
Unused	1.1	1.0
Total per frame	272	376

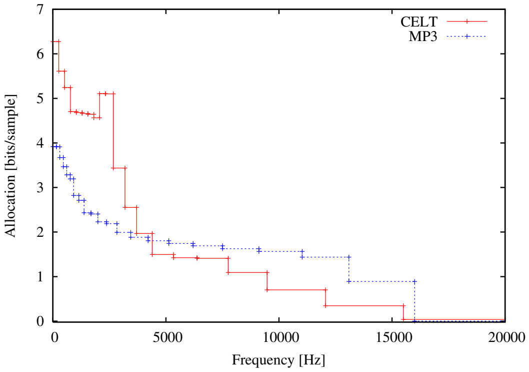


Figure 5. Average per-band bit allocation for CELT and MP3 at a 64 kbit/s constant bit-rate. The bit allocation shown is the number of innovation and fine energy quantisation bits allocated to one band divided by the number of MDCT bins in that band. The allocation for MP3 includes the bits from its scale factors that exceed 6dB resolution to ensure a fair comparison. The piecewise-constant lines also show the division of the bands for each codec. The average bit-rate for the quantised MDCT data, excluding the coarse energy/scalefactor quantisation and any meta-data, is around 55 kbit/s for each codec.

The average bit allocation for all parameters is detailed in Table I. In addition, Fig. 5 shows the average innovation and fine energy (Q_2 and Q_3) bit allocation as a function of frequency compared to MP3 for a bit-rate of 64 kbit/s. We see that CELT requires a higher bit-rate to code the low frequencies, which are often very tonal and difficult to encode with short frames. On the other hand, the energy constraint allows it to encode the high frequencies with fewer bits, while still maintaining good quality (see the next section for a quality comparison). Both codecs use approximately 10 kbit/s for the remaining parameters, despite the fact that MP3 frames are more than double the size of CELT frames (576-sample granules vs 256-sample frames).

IV. EVALUATION AND DISCUSSION

We implemented the CELT codec in C using both floating point and fixed point. The source code can be obtained at http://www.celt-codec.org/downloads/ and the results are

Table II
CHARACTERISTICS OF THE CODECS AT THE SAMPLING RATE USED.

	CELT	AAC-LD	G.722.1C	MP3
Rate (kHz)	44.1	44.1	32	44.1
Frame size (ms)	5.8	10.9	20	variable
Delay (ms)	8.7	34.8	40	>100
Bit-rate (kbit/s)	32-96	32-64	24,32,48	32-160

based on version 0.3.2 of the software. We used the floating-point version, but the fixed-point version does not cause noticeable quality degradation. Some audio samples are available at http://www.celt-codec.org/samples/tasl/ .

A. Other low-delay codecs

We compare CELT with three other codecs: AAC-LD [17], G.722.1C [18] and MPEG1 Layer III (MP3). The AAC-LD implementation tested is the one included in Apple's QuickTime Pro (with the "best quality" option selected). Although AAC-LD has a minimum delay of 20 ms, the Apple implementation uses 512-sample frames and a bit reservoir, which increases the total delay to 34.8 ms. The G.722.1C implementation was obtained from the Polycom website[2]. Despite the MP3 codec's high delay, the evaluation included it as a well known comparison point. We used the LAME MP3 encoder (CBR mode, with a 20 kHz low-pass filter and no bit reservoir), which significantly outperformed the dist10 reference MP3 encoder. Table II summarises the characteristics of all the codecs used in the evaluation. All the codecs compared have at least four times the delay of CELT.

Unlike AAC-LD and G.722.1C, the Fraunhofer Ultra-Low Delay (ULD) codec [19] can achieve coding delays similar to CELT using linear prediction with pre- and post-filtering [20]. Unfortunately, we were unable to obtain either an implementation or audio samples for that codec.

B. Subjective evaluation

Untrained listeners evaluated the basic audio quality of the codecs using the MUlti Stimulus test with Hidden Reference and Anchor (MUSHRA)[3] [21] methodology. They were presented with audio samples compressed with CELT, AAC-LD, G.722.1C, and MP3, in addition to low-pass anchors at 3.5 kHz and 7 kHz. The 7 kHz anchor is the upper bound achievable by wideband codecs such as G.722, AMR-WB, and G.729.1.

The first test included mostly speech samples, divided equally between male and female and encoded at 48 kbit/s. We used 2 British English speech samples from the EBU Sound Quality Assessment Material (SQAM) and 4 American English speech samples from the NTT Multi-Lingual Speech Database for Telephonometry[4]. The test also included two music samples: a pop music excerpt (Dave Matthews Band) and an orchestra excerpt (Danse Macabre). For this test, the CELT codec used 46.9 kbit/s (34 bytes per frame). Fig. 6 shows the average ratings by 15 untrained listeners[5].

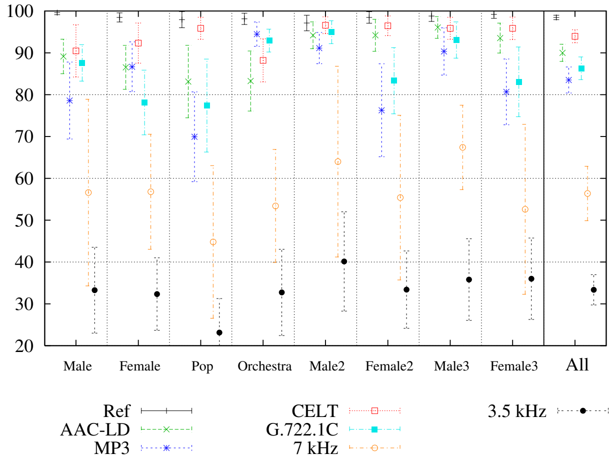


Figure 6. MUSHRA listening test results at 48 kbit/s with 95% confidence intervals.

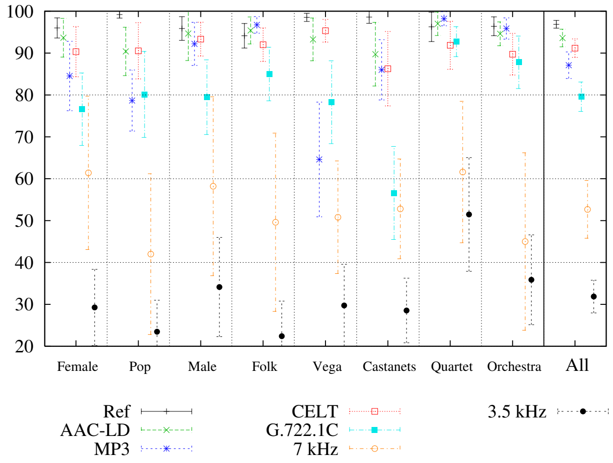


Figure 7. MUSHRA listening test results at 64 kbit/s with 95% confidence intervals.

A second listening test, at 64 kbit/s, included the following samples: male speech (SQAM), female speech (SQAM), *a cappella* singing (Suzanne Vega), vocal quartet (SQAM), pop music (Dave Matthews Band), folk music (Leahy), castanets (SQAM), and orchestra (Danse Macabre). For this test, the exact CELT bit-rate was 64.8 kbit/s (47 bytes per frame). The G.722.1C codec used 48 kbit/s, as this is the highest bit-rate it supports. Fig. 7 shows the average ratings for these samples were rated by the same 15 untrained listeners as in the previous test.

The error bars shown in Fig. 6 and 7 represent the 95% confidence interval for each codec, independently of the other codecs. However, since listeners were always directly comparing the same sample encoded with all codecs, a paired

[2] http://www.polycom.com/

[3] Using the RateIt graphical interface available at http://rateit.sf.net/

[4] We recovered the 44.1 kHz speech from the audio CD tracks and applied a notch filter to remove an unwanted 15.7 kHz tone from the recording.

[5] Due to an initial problem in generating the 7 kHz anchor, we only include results for it from the 5 listeners who took the test after the error was discovered. Results for other codecs and the 3.5 kHz anchor were not affected by this error and represent data from all 15 listeners.

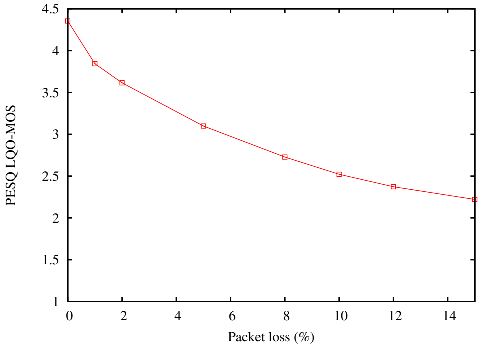


Figure 8. PESQ LQO-MOS as a function of the random packet loss rate.

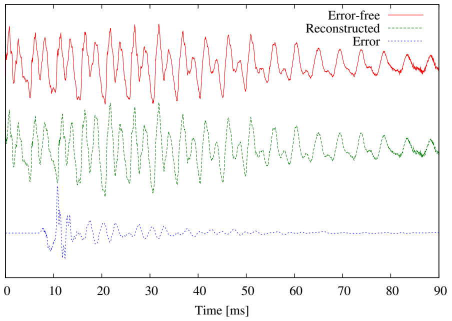


Figure 9. Decoder re-synchronisation after a missing frame. (top) Error-free decoding (middle) Reconstructed with missing packet (bottom) difference.

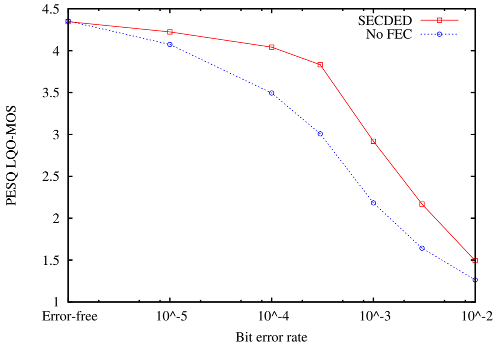


Figure 10. PESQ LQO-MOS as a function of the bit error rate both with SECDED on the the first 64 bits and without any FEC.

statistical test gives a better assessment of the statistical significance. At 48 kbit/s, CELT is found to out-perform all other codecs with greater than 95% confidence when using a paired permutation test (a paired t-test also shows greater than 95% confidence). At 64 kbit/s, the same statistical tests show that CELT out-performs all codecs except AAC-LD. The results for AAC-LD and G.722.1C are consistent with those reported at 32 kbit/s in [17] although in our 64 kbit/s test, G.722.1C was at a disadvantage, because it was the only codec operating at 48 kbit/s, the highest rate it supported.

The 7 kHz low-pass anchors included in the listening tests are equivalent to an uncompressed signal sampled at 16 kHz, which is the upper bound achievable by any wideband codec, such as G.722 and AMR-WB. Figs. 6 and 7 clearly show that listeners have a very strong preference for CELT and the other codecs over the 7 kHz low-pass anchor, demonstrating the benefit of a high sampling rate.

C. *Error robustness*

The next experiment measured the robustness of CELT to packet loss (frame erasure) and bit errors on speech data using the Perceptual Evaluation of Speech Quality (PESQ) [22] algorithm after down-sampling the decoder output to 8 kHz. We used PESQ because most other objective quality evaluation tools, such as PEAQ [23], are not designed to estimate quality in noisy channels. The test included 144 files from the NTT multilingual speech database, each from different speakers (72 male and 72 female), and 18 different languages.

The CELT codec is designed to be robust to packet loss. After a lost packet, two predictors need to be re-synchronised: the energy predictor and the pitch predictor. The re-synchronisation time of the energy predictor depends on the value α, while that of the pitch predictor depends on the pitch gain and period used in subsequent frames. In practise the re-synchronisation time is limited by the pitch predictor in voice segments.

Fig. 8 shows the PESQ LQO-MOS quality as a function of the random packet loss rate. The quality remains good at 5% random loss and degrades gracefully at higher loss rates. Fig. 9 shows the recovery from a lost packet during a voice segment. The recovery time is similar to that obtained by CELP codecs, which are also limited by pitch prediction. Informal listening tests verified that most speech utterances remain intelligible up to around 30% packet loss. Results for the Fraunhofer ULD codec show quality degradation on a MUSHRA test with 0.5% packet loss [24], the highest level they tested. However, one cannot infer the relative quality of CELT from this, as MUSHRA and PESQ MOS are not directly comparable.

Although robustness to bit errors is not a specific aim for the codec, we performed limited testing in bit-error conditions at 46.9 kbit/s (34 bytes per frame). Testing was performed in two different conditions: without any forward error correction (FEC) and with a simple 8-bit single-error-correcting, double-error-detecting (SECDED) code applied to the first 64 bits, which mainly consist of energy and pitch information. In cases where a double error was detected, the frame was considered lost. An evaluation of the speech quality as a function of the BER in Fig. 10 shows that robustness up to a 3×10^{-4} BER can be achieved at the cost of an 8 bit per frame (1.4 kbit/s) reduction in the codec's base bit-rate. Since the CELT bit-rate can be adjusted dynamically, a good strategy for transmission over a noisy channel, e.g. a wireless link, would be to adapt the bit-rate to the channel capacity, as the AMR codecs do [2].

D. *Complexity*

The floating point version of the codec requires approximately 30 MFLOPS for encoding and decoding in real-time at 44.1 kHz, or around 5% of a single core on a 2 GHz Intel Core 2 CPU, without CPU-specific optimisation. When implemented in fixed-point on a Texas Instruments TMS320C55x-

Table III
Implementation complexity of CELT on a TI TMS320C55x-family DSP. Per-channel data is persistent from one frame to another, while scratch data is only required while the codec is executing.

	Encoder	Decoder	Both
Computation (MIPS)	68	36	104
Per-channel RAM (kB)	5.1	4.6	9.7
Scratch data RAM (kB)	5.7	2.6	5.7
Table data ROM (kB)	6	3	6

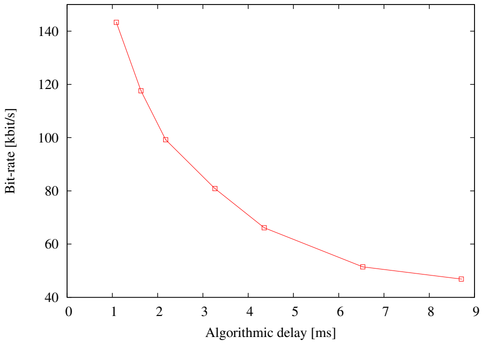


Figure 11. Bit-rate required as a function of the algorithmic delay to achieve a constant quality.

family DSP, it requires 104 MIPS to perform both encoding and decoding in real-time, using 7.7 kWords (15.4 kB) of data RAM. Table III gives more details of the complexity. This makes CELT comparable to AAC-LD, although more complex than G.722.1C, which has very low complexity.

E. Reducing the delay

We have shown results here for CELT operating with 256-sample frames and a 384-sample total algorithmic delay (8.7 ms at 44.1 kHz). The same codec can be used with even smaller frame sizes. Fig. 11 shows the bit-rate required to achieve a constant quality level when lowering the algorithmic delay. The reference quality is that obtained at 46.9 kbit/s with 8.7 ms delay and is measured using PQevalAudio[6], an implementation of the PEAQ basic model [23]. We observe that CELT scales well down to 3 ms delay, at which point the required bit-rate goes up very quickly. This is largely because the cost of encoding the band energies and the pitch information is nearly constant per frame, regardless of the frame size.

V. Conclusion

We proposed a new constrained-energy lapped transform (CELT) structure for speech coding at high sampling rates and very low-delay. The CELT algorithm can achieve high-quality coding at low delay by using an efficient algebraic shape-gain quantiser that preserves the spectral envelope of the signal, while minimising the side information transmitted.

Table IV
Detailed innovation bit allocation for a frame encoded at 64.8 kbit/s. For each band, we give the frequency (in MDCT bins) where the band starts, the width of the band (in MDCT bins), the number of pulses allocated to the band, and the number of bits required ($\log_2 V(N,K)$). Although no pulse is assigned to its innovation, band 19 still uses one bit for the folding sign (Section III-C3). Band 20, corresponding to frequencies above 20 kHz, is not coded and is set to zero at the decoder.

Frame #270 – Dave Matthews Band				
Band	Start	Width (N)	Pulses (K)	Bits
0	0	3	38	12.5
1	3	3	28	11.6
2	6	3	20	10.6
3	9	3	15	9.8
4	12	3	15	9.8
5	15	3	15	9.8
6	18	3	15	9.8
7	21	3	14	9.6
8	24	3	20	10.6
9	27	4	28	15.8
10	31	6	13	17.7
11	37	6	7	13.4
12	43	8	6	14.7
13	51	11	5	15.4
14	62	12	5	16.1
15	74	16	6	21.6
16	90	20	5	20.7
17	110	30	4	20.0
18	140	40	2	12.6
19	180	53	0	1
(20)	233	23	not coded	0
Innovation total				263.4
Overhead and padding				1.3

Additionally, a time-domain pitch predictor partially compensates for the poor frequency resolution obtained with the short MDCT windows. Results show that at 48 kbit/s and 64 kbit/s, CELT out-performs G.722.1C and MP3 on our test data and is comparable to AAC-LD, despite having less than one fourth of the algorithmic delay of the codecs to which it was compared.

There are still several ways to improve CELT, such as by incorporating better psychoacoustic masking in the dynamic bit allocation. This is a difficult problem both because there are few bits available for coding the allocation and because the analysis window is short.

Appendix A
Example bit-stream

Consider a single (typical) frame from the Dave Matthews Band excerpt encoded at 64.8 kbit/s. In that frame, the pitch gain is first encoded using 7 bits. Since the gain is non-zero, the pitch period is then encoded using 9.3 bits[7] ($\log_2 641$). The energy in each band is encoded with 94.7 bits, followed by the innovation, which requires 263.4 bits, as detailed in Table IV. The innovation bits in Table IV include the sparseness prevention signs (one each for bands 15-19), as explained in Section III-C3. One bit is left unused to account for overhead due to finite precision arithmetic in the range coder, for a total of 376 bits (47 bytes). The bit allocation for all other frames

[6] http://www-mmsp.ece.mcgill.ca/Documents/Software/Packages/AFsp/PQevalAudio.html

[7] Fractional bits are possible because we use a range coder with equiprobable symbols.

is very similar, with the main variation occuring in frames that do not have a pitch period (gain is zero).

References

[1] A. Carôt, U. Krämer, and G. Schuller, "Network music performance (NMP) in narrow band networks," in *Proc. 120^{th} AES Convention*, 2006.

[2] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen, "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE Trans. SAP*, vol. 10, no. 8, pp. 620– 636, 2002.

[3] S. Ragot, B. Kovesi, R. Trilling, D. Virette, N. Duc, D. Massaloux, S. Proust, B. Geiser, M. Gartner, S. Schandl, H. Taddei, Yang Gao, E. Shlomot, H. Ehara, K. Yoshida, T. Vaillancourt, R. Salami, Mi Suk Lee, and Do Young Kim, "ITU-T G.729.1: An 8-32 kbit/s scalable coder interoperable with G.729 for wideband telephony and voice over IP," in *Proc. ICASSP*, 2007, pp. 529–532.

[4] J.-M. Valin and C. Montgomery, "Improved noise weighting in CELP coding of speech – applying the Vorbis psychoacoustic model to Speex," in *Proc. 120^{th} AES Convention*, 2006.

[5] C. Montgomery, "Vorbis I specification," http://www.xiph.org/vorbis/doc/Vorbis_I_spec.html, 2004.

[6] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction(CELP): High-quality speech at very low bit rates," in *Proc. ICASSP*, 1984, pp. 937–940.

[7] F. Nordén and P. Hedelin, "Companded quantization of speech mdct coefficients," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 2, pp. 163–173, 2005.

[8] Charles H. Knapp and G. Clifford Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[9] J.-H. Chen, "Toll-quality 16 kb/s CELP speech coding with very low complexity," in *Proc. ICASSP*, 1995, pp. 9–12.

[10] G. Nigel and N. Martin, "Range encoding: An algorithm for removing redundancy from a digitised message.," in *Proc. of the Institution of Electronic and Radio Engineers International Conference on Video and Data Recording*, 1979.

[11] S. W. Golomb, "Run-length encodings," *IEEE Transactions on Information Theory*, vol. 12, no. 3, pp. 399–401, 1966.

[12] C. Laflamme, J.-P. Adoul, H. Y. Su, and S. Morissette, "On reducing computational complexity of codebook search in CELP coder through the use of algebraic codes," in *Proc. ICASSP*, 1990, vol. 1, pp. 177–180.

[13] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. on Information Theory*, vol. 32, pp. 568–583, 1986.

[14] James P. Ashley, Edgardo M. Cruz-Zeno, Udar Mittal, and Weimin Peng, "Wideband coding of speech using a scalable pulse codebook," in *Proc. of the 2000 IEEE Workshop on Speech Coding*, Sept. 2000, pp. 148–150.

[15] T. B. Terriberry, "On the overhead of range coders," http://people.xiph.org/~tterribe/notes/range.html, 2008.

[16] J. Makhoul and M. Berouti, "High-frequency regeneration in speech coding systems," in *Proc. ICASSP*, 1979.

[17] M. Lutzky, M. Schnell, M. Schmidt, and R. Geiger, "Structural analysis of low latency audio coding schemes," in *Proc.* 119^{th} *AES convention*, 2005.

[18] M. Xie, D. Lindbergh, and P. Chu, "From ITU-T G.722.1 to ITU-T G.722.1 Annex C: A new low-complexity 14kHz bandwidth audio coding standard," *Journal of Multimedia*, vol. 2, no. 2, 2007.

[19] S. Wabnik, G. Schuller, J. Hirschfeld, and U. Krämer, "Reduced bit rate ultra low delay audio coding," in *Proc. 120^{th} AES Convention*, 2006.

[20] G. D. T. Schuller, B. Yu, D. Huang, and B. Edler, "Perceptual audio coding using adaptive pre-and post-filters and lossless compression," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 6, pp. 379–390, 2002.

[21] ITU-R, *Recommendation BS.1534-1: Method for the subjective assessment of intermediate quality level of coding systems*, 2001.

[22] ITU-T, *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, 2001.

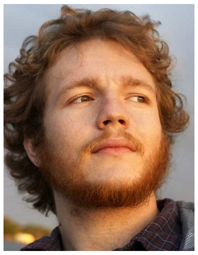
[23] ITU-R, *Recommendation BS.1387: Perceptual Evaluation of Audio Quality (PEAQ) recommendation*, 1998.

[24] S. Wabnik, G. Schuller, J. Hirschfeld, and U. Kraemer, "Packet loss concealment in predictive audio coding," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 227–230.

Jean-Marc Valin (S'03-M'05) holds B.Eng. ('99), M.A.Sc. ('01) and Ph.D. ('05) degrees in electrical engineering from the University of Sherbrooke. His Ph.D. research focused on bringing auditory capabilities to a mobile robotics platform, including sound source localisation and separation. In 2002, he authored the Speex open source speech codec, which he keeps maintaining to this date. Since 2005, he is a software lead architect at Octasic Inc. and his interests include acoustic echo cancellation and audio coding.

Timothy B. Terriberry received dual B.S. and M.S. degrees from Virginia Tech in 1999 and 2001, respectively, in both Mathematics and Computer Science, and a Ph.D. in Computer Science from the Univeristy of North Carolina at Chapel Hill in 2006. He has volunteered for the Xiph.Org Foundation – a non-profit organization that develops free, open multimedia protocols and software – since 2002.

Christopher (Monty) Montgomery founded the Xiph.Org Foundation and authored Ogg Vorbis and other open-source packages. He holds a B.S. in Electrical Engineering and Computer Science from MIT and a M.Eng. in Computer Engineering from the Tokodai in Japan. He is currently a Senior Engineer at Red Hat.



Gregory Maxwell is a Senior Systems Engineer for Juniper Networks in Herndon, Virginia and has volunteered for the Xiph.Org Foundation since 1999. His interests include spatial audio, audio compression, and radio systems.