Xiph.Org Foundation

The Opus Codec

To be presented at the 135th AES Convention
2013 October 17–20 New York, USA

This paper was accepted for publication at the 135th AES Convention. This version of the paper is from the authors and not from the AES.

High-Quality, Low-Delay Music Coding in the Opus Codec

Jean-Marc Valin[1], Gregory Maxwell[1], Timothy B. Terriberry[1], and Koen Vos[2]

[1] *Mozilla, Xiph.Org*

[2] *Microsoft*

Correspondence should be addressed to Jean-Marc Valin (`jmvalin@jmvalin.ca`)

ABSTRACT

The IETF recently standardized the Opus codec as RFC6716. Opus targets a wide range of real-time Internet applications by combining a linear prediction coder with a transform coder. We describe the transform coder, with particular attention to the psychoacoustic knowledge built into the format. The result out-performs existing audio codecs that do not operate under real-time constraints.

1. INTRODUCTION

In RFC 6716 [1] the IETF recently standardized Opus [2], a highly versatile audio codec designed for interactive Internet applications. This means support for speech and music, operating over a wide range of changing bitrates, integration with the Real-Time Protocol (RTP), and good packet loss concealment, with a low algorithmic delay.

Opus scales to delays as low as 5 ms, even lower than AAC-ELD (15 ms). Applications such as network music performance require these ultra-low delays [3]. Despite the low delay, Opus is competitive with high-delay codecs designed for storage and streaming, such as Vorbis and the HE and LC variants of AAC, as the evaluations in Section 6 show.

Opus supports

- Bitrates from 6 kb/s to 510 kb/s,
- Five audio bandwidths, from narrowband (8 kHz) to fullband (48 kHz),
- Frame sizes from 2.5 ms to 60 ms,
- Speech and music, and
- Mono and stereo coupling.

In-band signaling can dynamically change all of the above, with no switching artifacts.

We created the Opus codec from two core technologies: Skype's SILK [4] codec, based on linear prediction, and Xiph.Org's CELT codec, based on the

Modified Discrete Cosine Transform (MDCT). Section 2 presents a high-level view of their unification into Opus. After many major incompatible changes to the original codecs, the result is open source, with patents licensed under royalty-free terms[1]. The reference encoder is professional-grade, and supports

- Constant bit-rate (CBR) and variable bit-rate (VBR) rate control,
- Floating point and fixed-point arithmetic, and
- Variable encoder complexity.

Its CBR produces packets with exactly the size the encoder requested, without a bit reservoir to imposes additional buffering delays, as found in codecs such as MP3 or AAC-LD. It has two VBR modes: Constrained VBR (CVBR), which allows bitrate fluctuations up to the average size of one packet, making it equivalent to a bit reservoir, and true VBR, which does not have this constraint.

This paper focuses on the CELT mode of Opus, which is used primarily for encoding music. Although it retains the fundamental principles of the original algorithms published in [5, 6] reviewed in Section 3, the CELT algorithm used in Opus differs significantly from that work. We have psychoacoustically tuned the bit allocation and quantization, as Section 4 outlines, and designed additional tools to conceal artifacts, which Section 5 describes.

2. OVERVIEW OF OPUS

Opus operates in one of three modes:

- SILK mode (speech signals up to wideband),
- CELT mode (music and high-bitrate speech), or
- Hybrid mode (SILK and CELT simultaneously for super-wideband and fullband speech).

Fig. 1 shows a high-level overview of Opus. CELT always operates at a sampling rate of 48 kHz, while SILK can operate at 8 kHz, 12 kHz, or 16 kHz. In hybrid mode, the crossover frequency is 8 kHz, with SILK operating at 16 kHz and CELT discarding all frequencies below the 8 kHz Nyquist rate.

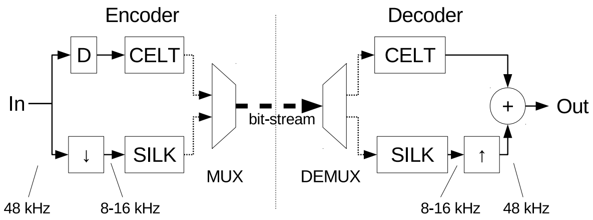

Fig. 1: High-level overview of the Opus codec.

CELT's look-ahead is 2.5 ms, while SILK's look-ahead is 5 ms, plus 1.5 ms for the resampling (including both encoder and decoder resampling). For this reason, the CELT path in the encoder adds a 4 ms delay. However, an application can restrict the encoder to CELT and omit that delay. This reduces the total look-ahead to 2.5 ms.

2.1. Configuration and Switching

Opus signals the mode, frame size, audio bandwidth, and channel count (mono or stereo) in-band. It encodes this in a *table-of-contents* (TOC) byte at the start of each packet [1]. Additional internal framing allows it to pack multiple frames into a single packet, up to a maximum duration of 120 ms. Unlike the rest of the Opus bitstream, the TOC byte and internal framing are not entropy coded, so applications can easily access the configuration, split packets into individual frames, and recombine them.

Configuration changes that use CELT on both sides (or between wideband SILK and Hybrid mode) use the overlap of the transform window to avoid discontinuities. However, switching between CELT and SILK or Hybrid mode is more complicated, because SILK operates in the time domain without a window. For such cases, the bitstream can include an additional 5 ms *redundant CELT frame* that the decoder can overlap-add to bridge and gap between the discontinuous data. Two redundant CELT frames—one on each side of the transition—allow smooth transitions between modes that use SILK at different sampling rates. The encoder handles all of this transparently. The application may not even notice that the mode has changed.

3. CONSTRAINED-ENERGY LAPPED TRANSFORM (CELT)

Like most transform coding algorithms, CELT is based on the MDCT. However, the fundamental idea

[1] http://opus-codec.org/license/

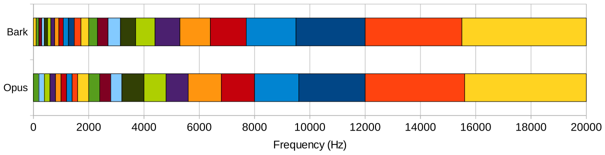

Fig. 2: CELT band layout vs. the Bark scale.

behind CELT is that the most important perceptual aspect of an audio signal is the spectral envelope. CELT preserves this envelope by explicitly coding the energy of a set of bands that approximates the auditory system's critical bands [7]. Fig. 2 illustrates the band layout used by Opus. The format itself incorporates a significant amount of psychoacoustic knowledge. This not only reduces certain types of artifacts, it also avoids coding some parameters.

CELT uses flat-top MDCT windows with a fixed overlap of 2.5 ms, regardless of the frame size, as Fig. 3 shows. The overlapping part of the window is the Vorbis [8] power-complementary window

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

Unlike the AAC-ELD low-overlap window, the Opus window is still symmetric. Compared to the full-overlap of MP3 or Vorbis, the low overlap allows lower algorithmic delay and simplifies the handling of transients, as Section 3.1 describes. The main drawback is increased spectral leakage, which is problematic for highly tonal signals. We mitigate this in two ways. First, the encoder applies a first-order pre-emphasis filter $A_p(z) = 1 - 0.85z^{-1}$ to the input, and the decoder applies the inverse de-emphasis filter. This attenuates the low frequencies (LF), reducing the amount of leakage they cause at higher frequencies (HF). Second, the encoder applies a perceptual prefilter, with a corresponding postfilter in the decoder, as Section 5.1 describes.

Fig. 4 shows a complete block diagram of CELT. Sections 4 and 5 describe the various components.

3.1. Handling of Transients

Like other transform codecs, Opus controls pre-echo primarily by varying the MDCT size. When the encoder detects a transient, it computes multiple short MDCTs over the frame and interleaves the output coefficients. For 20-ms frames, there are 8 MDCTs with full-overlap, 5 ms windows. We constrain the band sizes to be a multiple of the number of short MDCTs, so that the interleaved coefficients form bands of the same size, covering the same part of the spectrum in each block, as the corresponding band of a long MDCT.

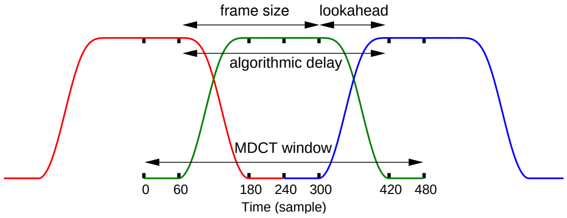

Fig. 3: Low-overlap window used for 5 ms frames.

4. QUANTIZATION AND ENCODING

Opus supports any bitrate that corresponds to an integer number of bytes per frame. Rather than signal a rate explicitly in the bitstream, Opus relies on the lower-level transport protocol, such as RTP, to transmit the payload length. The decoder, not the encoder, makes many bit allocation decisions automatically based on the number of bits remaining. This means that the encoder must determine the final rate early in the encoding process, so it can make matching decisions, unlike codecs such as AAC, MP3, and Vorbis. This has two advantages. First, the encoder need not transmit these decisions, avoiding the associated overhead. Second, they allow the encoder to achieve a target bitrate exactly, without repeated encoding or bit reservoirs. Even though entropy coding produces variable-sized output, these dynamic adjustments to the bit allocation ensure that the coded symbols never exceed the number of bytes allocated for the frame by the encoder earlier in the process. In the vast majority of cases, the encoder also wastes less than two bits.

Opus encodes most symbols using a range coder [9]. Some symbols, however, have a power-of-two range and approximately uniform probability. Opus packs these as *raw bits*, starting at the end of the packet, back towards the end of the range coder output, as Fig. 5 illustrates. This allows the decoder to rapidly switch between decoding symbols with the range coder and reading raw bits, without interleaving the

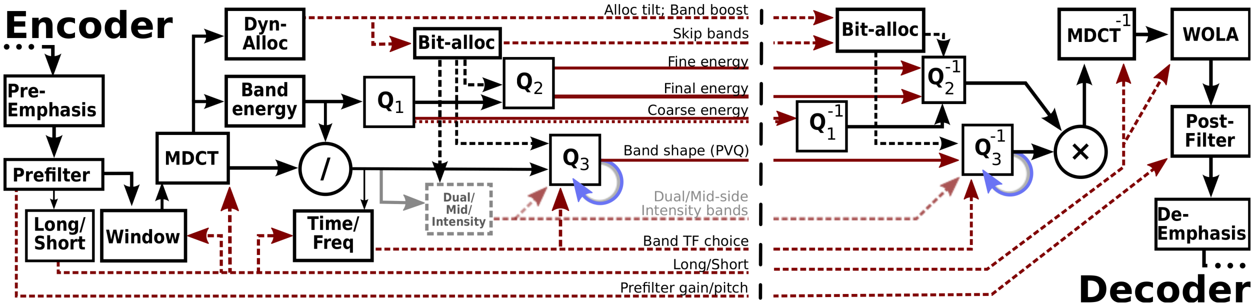

Fig. 4: Overview of the CELT algorithm.

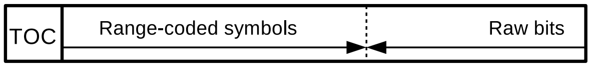

Fig. 5: Layout and coding order of the bitstream.

data in the packet. It also improves robustness to bit errors, as corruption in the raw bits does not desynchronize the range coder. A special termination rule for the range coder, described in Section 5.1.5 of RFC 6716 [1], ensures the stream remains decodable regardless of the values of the raw bits, while using at most 1 bit of padding to separate the two.

4.1. Coarse Energy Quantization (Q_1)

The most important information encoded in the bitstream is the energy of the MDCT coefficients in each band. Band energy is quantized using a two-pass coarse-fine quantizer. The coarse quantizer uses a fixed 6 dB resolution for all bands, with inter-band prediction and, optionally, inter-frame prediction. The 2D z-transform of the predictor is

$$A\left(z_\ell, z_b\right) = \left(1 - \alpha z_\ell^{-1}\right) \cdot \frac{1 - z_b^{-1}}{1 - \beta z_b^{-1}} \ , \qquad (2)$$

where ℓ is the frame index and b is the band. Inter-frame prediction can be turned on or off for any frame. When enabled, both α and β are non-zero and depend on the frame size. When disabled, $\alpha = 0$ and $\beta = 0.15$. Inter-frame prediction is more efficient, but less robust to packet loss. The encoder can use packet loss statistics to force inter-frame prediction off adaptively. The prediction residual is entropy-coded assuming a Laplace probability distribution with per-band variances trained offline.

4.2. Bit Allocation

Rather than transmitting scale factors like MP3 and AAC or a floor curve like Vorbis, CELT mostly allocates bits implicitly. After coarse energy quantization, the encoder decides on the total number of bytes to use for the frame. Then both the encoder and decoder run the same *bit-exact* bit allocation function to partition the bits among the bands. CELT interpolates between several static allocation prototypes (see Fig. 6) to achieve the target rate.

Some bands may not receive any bits. The decoder reconstructs them using only the energy, generating fine details by spectral folding, as Section 4.4.1 details. When a band receives very few bits, the sparse spectrum that could be encoded with them would sound worse than spectral folding. Such bands are automatically skipped, redistributing the bits they would have used to code their spectrum to the remaining bands. The encoder can also skip more bands via explicit signaling. This allows it to give the skip decisions some hysteresis between frames.

After the initial allocation, bands are encoded one at a time. In practice, a band may use slightly more or slightly fewer bits than allocated. The difference propagates to subsequent bands to ensure that the final rate still matches the overall target. Automatically adjusting the allocation based on the actual bits used makes achieving CBR easy.

The implicit allocation produces a nearly constant signal-to-noise ratio in each band, with the LF coded at a higher resolution than the HF. It approximates the real masking curve well without any signaling, and achieves good quality by itself, as demonstrated

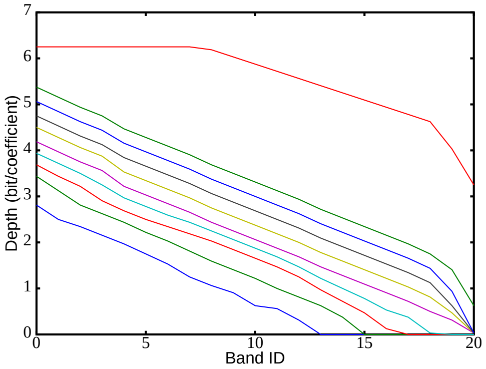

Fig. 6: Static bit allocation curves in bits/sample for each band and for multiple bitrates.

by earlier versions of the algorithm [5, 6]. However, it does not cover two theoretical phenomena:

1. Tonality: tones provide weaker masking than noise, requiring a finer resolution. Since tones usually have harmonics in many bands, the encoder increases the total rate for these frames.
2. Inter-band masking: a band may be masked by neighboring bands, though this is weaker than intra-band masking.

CELT provides two signaling mechanisms that adjust the implicit allocation: one that changes the tilt of the allocation, and one that boosts specific bands.

4.2.1. Allocation Tilt

The allocation tilt parameter changes the slope of the bit allocation as a function of the band index by up to ±5/64 bit/sample/band, in increments of 1/64 bit/sample/band. Although in theory the slope of the masking threshold should follow the slope of the signal's spectral envelope, we have observed that LF-dominated signals require more bits in the LF, with a similar observation for HF-dominated signals.

4.2.2. Band Boost

When a specific band requires more bits, the bitstream includes a mechanism for increasing its allocation (reducing the allocation of all other bands). Versions 1.0.x and earlier of the Opus reference implementation rarely use this band boost. However, newer versions use it to improve quality in the following circumstances:

- In transients frames, bands dominated by the leakage of the shorter MDCTs receive more bits.
- Bands that have significantly larger energy than surrounding bands receive more bits.

CELT does not provide a mechanism to reduce the allocation of a single band because it would not be worth the signaling cost.

4.3. Fine Energy Quantization (Q_2)

Once the per-band bit allocation is determined, the encoder refines the coarsely-quantized energy of each band. Let a be the total allocation for a band containing N_{DoF} degrees of freedom[2]. We approximate (30) from [10] to obtain the fine energy allocation:

$$a_f = \frac{a}{N_{DoF}} + \frac{1}{2}\log_2 N_{DoF} - K_{fine} \ , \tag{3}$$

where K_{fine} is a tuned *fine allocation offset*. We round the result to an integer and code the refinement data as raw bits. Bands where $N_{DoF} = 2$ get slightly more bits, and we slightly bias the allocation upwards when adding the first and second fine bit.

If any bits are left unused at the very end of the frame, each band may add one additional bit per channel to refine the band energy, starting with bands for which a_f was rounded down.

4.4. Pyramid Vector Quantization (Q_3)

Let $\mathbf{X}_b$ be the MDCT coefficients for band b. We normalize the band with the unquantized energy,

$$\mathbf{x}_b = \frac{\mathbf{X}_b}{\|\mathbf{X}_b\|} \ , \tag{4}$$

producing a unit vector on an N-sphere, coded with a pyramid vector quantizer (PVQ) [11] codebook:

$$S(N,K) = \left\{ \frac{\mathbf{y}}{\|\mathbf{y}\|}, \ \mathbf{y} \in \left\{ \mathbb{Z}^{\mathbb{N}} : \sum_{i=0}^{N-1} |y_i| = K \right\} \right\} \ ,$$

where K is the L_1-norm of $\mathbf{y}$, i.e. the number of *pulses*. The codebook size obeys the recurrence

$$\begin{aligned} V(N,K) &= V(N,K-1) \\ &\quad + V(N-1,K) + V(N-1,K-1) \ , \end{aligned} \tag{5}$$

[2] Usually equal to the number of coefficients. When stereo coupling is used on a band with more than 2 coefficients, the combined band has an additional degree of freedom.

with $V(N,0) = 1$ and $V(0,K) = 0,\ K > 0$. Because $V(N,K)$ is rarely a power of two, we use the range coder with a uniform probability to encode the codeword index, derived from $\mathbf{y}$ using the method of [11]. When $V(N,K)$ is larger than 255, the index is renormalized to fall in the range $[128, 255]$ and the least significant bits are coded using raw bits. The uniform probability allows both the encoder and the decoder to choose K such that $\log_2 V(N,K)$ achieves allocation determined in Section 4.2.

4.4.1. Spectral Folding

When a band receives no bits, the decoder replaces the spectrum of that band with a normalized copy of MDCT coefficients from lower frequencies. This preserves some temporal and tonal characteristics from the original band, and CELT's energy normalization preserves the spectral envelope. Spectral folding is far less advanced than spectral band replication (SBR) from HE-AAC and mp3PRO, but is computationally inexpensive, requires no extra delay, and the decision to apply it can change frame-by-frame.

4.5. Stereo

Opus supports three different stereo coupling modes:

1. Mid-side (MS) stereo
2. Dual stereo
3. Intensity stereo

A coded band index denotes where intensity stereo begins: all bands above it use intensity stereo, while all bands below it use either MS or dual stereo. A single flag at the frame level chooses between them.

4.5.1. Mid-Side Stereo

We apply MS stereo coupling separately on each band, after normalization. Because we code the energy of each channel separately, MS stereo coupling never introduces cross-talk between channels and is safe even when dual stereo is more efficient. Let $\mathbf{x}_l$ and $\mathbf{x}_r$ be the normalized band for the left and right channels, respectively. The orthogonal mid and side signals are computed as

$$\mathbf{M} = \frac{\mathbf{x}_l + \mathbf{x}_r}{2} \ , \tag{6}$$

$$\mathbf{S} = \frac{\mathbf{x}_l - \mathbf{x}_r}{2} \ . \tag{7}$$

Opus encodes the mid and side as normalized signals $\mathbf{m} = \mathbf{M}/\|\mathbf{M}\|$ and $\mathbf{s} = \mathbf{S}/\|\mathbf{S}\|$. To recover $\mathbf{M}$ and $\mathbf{S}$ from $\mathbf{m}$ and $\mathbf{s}$, we need to know the ratio of $\|\mathbf{S}\|$ to $\|\mathbf{M}\|$, which we encode as the angle

$$\theta_s = \arctan \frac{\|\mathbf{S}\|}{\|\mathbf{M}\|} \ . \tag{8}$$

Explicitly coding θ_s preserves the stereo width and reduces the risk of *stereo unmasking* [12, 7], since it preserves the energy of the difference signal, in addition to the energy in each channel. We quantize θ_s uniformly, deriving the resolution the same way as the fine energy allocation. Uniform quantization of θ_s achieves optimal mean-squared error (MSE).

Let $\hat{\mathbf{m}}$ and $\hat{\mathbf{s}}$ be the quantized versions of $\mathbf{m}$ and $\mathbf{s}$. We can compute the reconstructed signals as

$$\hat{\mathbf{x}}_l = \hat{\mathbf{m}} \cos \hat{\theta}_s + \hat{\mathbf{s}} \sin \hat{\theta}_s \ , \tag{9}$$

$$\hat{\mathbf{x}}_r = \hat{\mathbf{m}} \cos \hat{\theta}_s - \hat{\mathbf{s}} \sin \hat{\theta}_s \ . \tag{10}$$

As a result of the quantization, $\hat{\mathbf{m}}$ and $\hat{\mathbf{s}}$ may not be orthogonal, so $\hat{\mathbf{x}}_l$ and $\hat{\mathbf{x}}_r$ may not have exactly unit norm and must be renormalized.

The MSE-optimal bit allocation for $\mathbf{m}$ and $\mathbf{s}$ depends on $\hat{\theta}_s$. Let N be the size of the band and a be the total number of bits available for $\mathbf{m}$ and $\mathbf{s}$. Then the optimal allocation for $\mathbf{m}$ is

$$a_{mid} = \frac{a - (N-1)\log_2 \tan \theta_s}{2} \ . \tag{11}$$

The larger of $\mathbf{m}$ and $\mathbf{s}$ is coded first, and any unused bits are given to the other channel. As a special case, when $N = 2$ we use the orthogonality of $\mathbf{m}$ and $\mathbf{s}$ to code one of the channels using a single sign bit.

4.5.2. Dual Stereo

Dual stereo codes the normalized left and right channels independently. We use this only when the correlation between the channels is not strong enough to make up for the cost of coding the θ_s angles.

4.5.3. Intensity Stereo

Intensity stereo also works in the normalized domain, using a single mid channel with no side. Instead of θ_s, we code a single *inversion* flag for each band. When set, we invert the right channel, producing two channels 180 degrees out of phase.

4.6. Band splitting

At high bitrates, we allocate some bands hundreds of bits. To avoid arithmetic on large integers in the PVQ index calculations, we split bands with more than 32 bits, using the same process as MS stereo. **M** and **S** are set to the first and second half of the band, with θ_s indicating the distribution of energy between the two halves. If a band contains data from multiple short MDCTs, we bias the bit allocation to account for pre-echo or forward masking using $\hat{\theta}_s$. If one sub-vector still requires more than 32 bits, we split it recursively. This recursion stops after 4 levels (1/16th the size of the original band), which puts a hard limit on the number of bits a band can use. This limit lies far beyond the rate needed to achieve transparency in even the most difficult samples.

5. PSYCHOACOUSTIC IMPROVEMENTS

We can achieve good audio quality using just the algorithms described above. However, four different psychoacoustically-motivated improvements make coding artifacts even less audible.

5.1. Prefilter and Postfilter

The low-overlap window increases leakage in the MDCT, resulting in higher quantization noise on highly tonal signals. Widely-spaced harmonics in periodic signals provide especially little masking. Opus mitigates this problem using a pitch-enhancing post-filter. Unlike speech codec postfilters, we run a matching prefilter on the encoder side. The pair provides perfect reconstruction (in the absence of quantization), allowing us to enable the postfilter even at high bitrates. Although the filters look like a pitch predictor, unlike standard pitch prediction we apply the prefilter to the unquantized signal, allowing pitch periods shorter than the frame size. The gain and period are transmitted explicitly. When these change between two frames, the filter response is interpolated using a 2.5 ms cross-fade window equal to the square of the $w(n)$ power-complementary window. We use a 5-tap prefilter with an impulse response of

$$A(z) = 1 - g \cdot \left[a_{p,2} \left(z^{-T-2} + z^{-T+2} \right) + a_{p,1} \left(z^{-T-1} + z^{-T+1} \right) + a_{p,0} z^{-T} \right] \ , \quad (12)$$

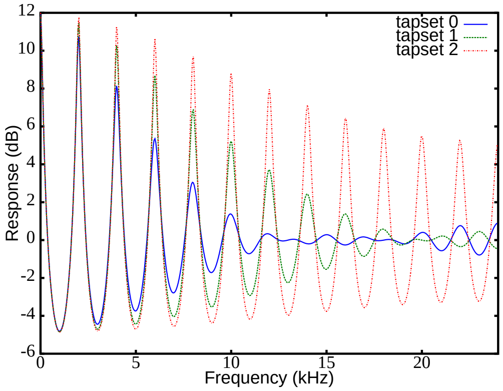

Fig. 7: Frequency response of the different postfilter tapsets for $T = 24$, $g = 0.75$.

where T is the pitch period, g is the gain, and $a_{p,i}$ are the coefficients of tapset p. We choose one of three different tapsets to control the range of frequencies to which we apply the enhancement. They are

$$\begin{aligned} a_{0,\cdot} &= [0.80\ 0.10\ 0] \ , \\ a_{1,\cdot} &= [0.46\ 0.27\ 0] \ , \\ a_{2,\cdot} &= [0.30\ 0.22\ 0.13] \ . \end{aligned} \quad (13)$$

The pitch period lies in the range $[15, 1022]$, and the gain varies between 0.09 and 0.75. Fig. 7 shows the frequency response of each tapset for a period of $T = 24$ (2 kHz) and a gain $g = 0.75$.

Subjective testing conducted by Broadcom on an earlier version of the algorithm demonstrated the postfilter's effectiveness [13].

5.2. Variable Time-Frequency Resolution

Some frames contain both tones and transients, requiring both good time resolution and good frequency resolution. Opus achieves this by selectively modifying the time-frequency (TF) resolution in each band. For example, Opus can have good frequency resolution for LF tonal content while retaining good time resolution for a transient's HF. We change the TF resolution with a Hadamard transform, a cheap approximation of the DCT. When using multiple short MDCTs (good time resolution), we increase the frequency resolution of a band by applying the Hadamard transform to the same coefficient across multiple MDCTs. This can increase the frequency resolution by a factor of 2 to 8, decreasing the time resolution by the same amount.

The Hadamard transform of consecutive coefficients increases the time resolution of a long MDCT. This

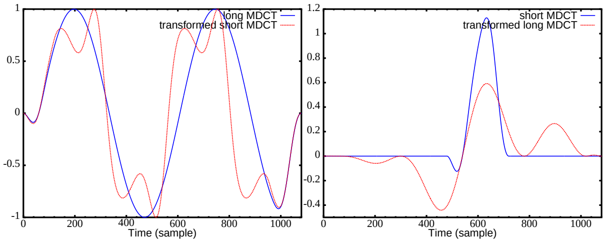

Fig. 8: Basis functions with modified time-frequency resolution for a 20 ms frame. Left: fourth basis function of a long MDCT vs. the equivalent basis function from TF modification of 8 short MDCTs. Right: First ("DC") basis function of a short MDCT vs. the equivalent basis function from TF modification of the first 8 coefficients of a long MDCT.

yields more time-localized basis functions, although they have more ringing than the equivalent short MDCT basis functions. Fig. 8 illustrates basis functions produced by adaptively modifying the time-frequency resolution for a 20 ms frame.

5.3. Spreading Rotations

A common type of artifact in transform codecs is tonal noise, also known as *birdies*. When quantizing a large number of HF MDCT coefficients to zero, the few remaining non-zero coefficients sound tonal even when the original signal did not. This is most noticeable in low-bitrate MP3s. Opus greatly reduces tonal noise by applying *spreading rotations*. The encoder applies these rotations to the normalized signal prior to quantization, and the decoder applies the inverse rotations, as Fig. 9 shows.

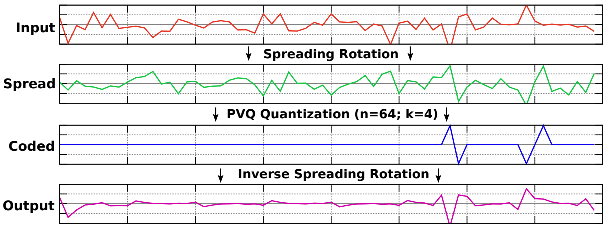

Fig. 9: Spreading example.

We construct the spreading rotations from a series of 2D Givens rotations. Let $\mathbf{G}\left(m,n,\theta_r\right)$ denote a Givens rotation matrix by angle θ_r between coefficients m and n in some band with N coefficients, with angles near $\pi/4$ implying more spreading. Then the spreading rotations are

$$\mathbf{R}\left(\theta_r\right) = \prod_{k=0}^{N-3} \mathbf{G}\left(k,k+1,\theta_r\right) \cdot \prod_{k=2}^{N} \mathbf{G}\left(N-k,N-k+1,\theta_r\right) \ . \quad (14)$$

In other words, we rotate adjacent coefficient pairs one at a time from the beginning of the vector to the end, and then back. We determine θ_r from the band size, N, and the number of pulses used, K:

$$\theta_r = \frac{\pi}{4}\left(\frac{N}{N+\delta K}\right)^2 \ , \quad (15)$$

where δ is the *spreading constant*. Once per frame, the encoder selects δ from one of three values: 5, 10, or 15, or disables spreading completely.

In transient frames, we apply the spreading rotations to each short MDCT separately to avoid pre-echo. When vectors of more than 8 coefficients need to be rotated, we apply an additional set of rotations to pairs of coefficients $\lfloor\sqrt{N}\rfloor$ positions apart, using the angle $\theta_r^{'} = \frac{\pi}{2} - \theta_r$. This spreads the energy within large bands more widely.

5.4. Collapse Prevention

In transients at low bitrates, Opus may quantize all of the coefficients in a band corresponding to a particular short MDCT to zero. Even though we preserve the energy of the entire band, this quantization causes audible drop outs, as Fig. 10 shows on the left. The decoder detects *holes* that occur when a short MDCT receives no pulses in a given band, or when folding copies such a hole into a higher band, and fills them with pseudo-random noise at a level equal to the minimum band energy over the previous two frames. The encoder transmits one flag per frame that can disable collapse prevention. We do this after two consecutive transients to avoid putting too much energy in the holes. Fig. 10 shows the result of collapse prevention on the right. The short drop outs around each transient are no longer audible.

6. EVALUATION AND RESULTS

This section presents a quality evaluation of Opus's CELT mode on music signals. More complete evaluation data on Opus is available at [14].

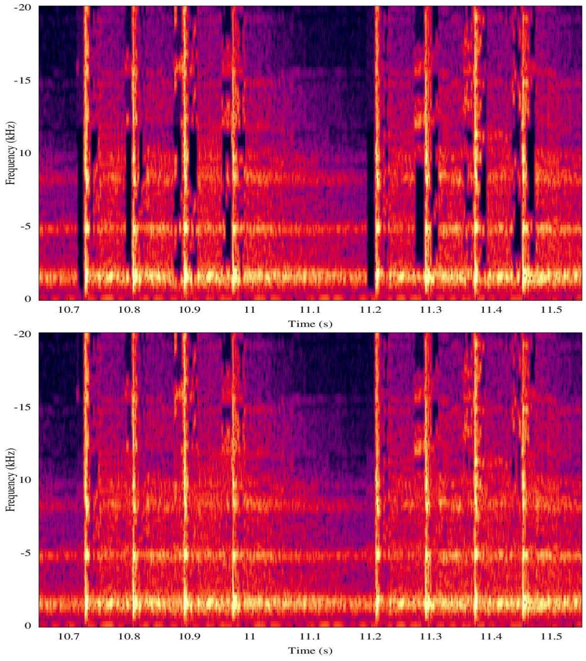

Fig. 10: Extreme collapse prevention example for castanets at 32 kb/s mono. Top: without collapse prevention. Bottom: with collapse prevention.

6.1. Subjective Quality

Volunteers of the HydrogenAudio forum[3] evaluated the quality of 64 kb/s VBR Opus on fullband stereo music with headphones. 13 listeners evaluated 30 samples using the ITU-R BS.1116-1 methodology [15] with

- The Opus [2] reference implementation (v0.9.2),
- Apple's HE-AAC[4] (QuickTime v7.6.9),
- Nero's HE-AAC[5] (v1.5.4.0), and
- Ogg Vorbis (AoTuV[6] v6.02 Beta).

Apple's AAC-LC at 48 kb/s served as a low anchor. Fig. 11 shows the results. A pairwise resampling-based free step-down analysis using the max(T) algorithm [16, 17] reveals that Opus is better than the other codecs with greater than 99.9% confidence.

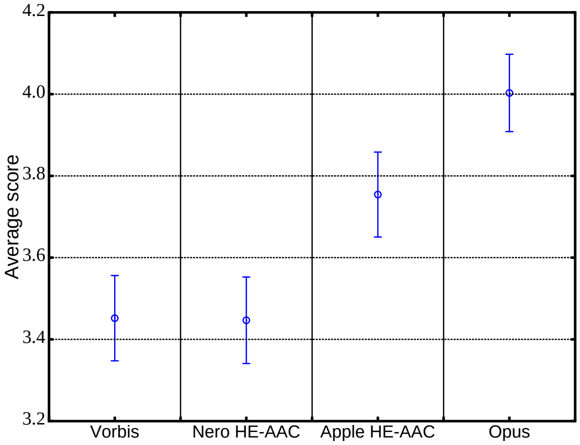

Fig. 11: Results of the 64 kb/s evaluation. The low anchor (omitted) was rated at 1.54 on average.

Apple's HE-AAC was better than both Nero's HE-AAC and Vorbis with greater than 99.9% confidence. Nero's HE-AAC and Vorbis were statistically tied. A simple ANOVA analysis gives the same results.

6.2. Cascading Performance

In broadcasting applications, audio streams are compressed and recompressed multiple times. According to [18], typical broadcast chains may include up to 5 lossy encoding stages. For this reason, we compare the cascading quality of Opus to both Vorbis and MP3 using PQevalAudio [19], an implementation of the PEAQ basic model [20]. Fig 12 plots quality as a function of bitrate and the number of cascaded encodings. Opus performs better than MP3 and Vorbis in the presence of cascading, with 64 kb/s Opus even out-performing 128 kb/s MP3. Although the Opus quality with 5 ms frames is lower than for 20 ms frames, it is still acceptable, and better than MP3.

7. CONCLUSION AND FUTURE WORK

By building psychoacoustic knowledge into the Opus format, we minimize the side information it transmits and the impact of coding artifacts. This allows Opus to achieve higher music quality than existing non-real-time codecs, even under cascading. Since Opus was only recently standardized, we are continuing to improve its encoder, experimenting with such things as look-ahead and automatic frame size switching for non-real-time encoding.

[3] http://hydrogenaudio.org/

[4] With constrained VBR, as it cannot run unconstrained

[5] http://www.nero.com/enu/company/about-nero/nero-aac-codec.php

[6] http://www.geocities.jp/aoyoume/aotuv/

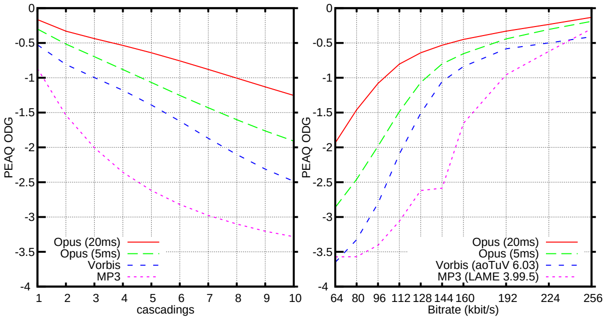

Fig. 12: Cascading quality. Left: Quality degradation vs. number of cascadings at 128 kb/s. Right: Quality degradation vs. bitrate after 5 cascadings.

8. REFERENCES

[1] J.-M. Valin, K. Vos, and T. B. Terriberry. Definition of the Opus Audio Codec. RFC 6716, `http://www.ietf.org/rfc/rfc6716.txt`, September 2012.

[2] Opus website. `http://opus-codec.org/`.

[3] A. Carôt. *Musical Telepresence – A Comprehensive Analysis Towards New Cognitive and Technical Approaches.* PhD thesis, University of Lübeck, 2009.

[4] K. Vos, S. Jensen, and K. Sørensen. SILK speech codec. IETF Internet-Draft `http://tools.ietf.org/html/draft-vos-silk-02`.

[5] J.-M. Valin, T. B. Terriberry, and G. Maxwell. A full-bandwidth audio codec with low complexity and very low delay. In *Proc. EUSIPCO*, 2009.

[6] J.-M. Valin, T. B. Terriberry, C. Montgomery, and G. Maxwell. A high-quality speech and audio codec with less than 10 ms delay. *IEEE Trans. Audio, Speech and Language Processing*, 18(1):58–67, 2010.

[7] B. C.J. Moore. *An Introduction to the Psychology of Hearing.* fifth edition, 2004.

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

[9] G. Nigel and N. Martin. Range encoding: An algorithm for removing redundancy from a digitised message. In *Proc. Video and Data Recording Conference*, 1979.

[10] H. Krüger, R. Schreiber, B. Geiser, and P. Vary. On logarithmic spherical vector quantization. In *Proc. ISITA*, 2008.

[11] T. R. Fischer. A pyramid vector quantizer. *IEEE Trans. on Information Theory*, 32:568–583, 1986.

[12] J. D. Johnston and A. J. Ferreira. Sum-difference stereo transform coding. In *Proc. ICASSP*, volume 2, pages 569–572, 1992.

[13] R. Chen, T. B. Terriberry, J. Skoglund, G. Maxwell, and H. T. M. Nguyet. Opus testing. In *Proc. codec WG,* 80^{th} *IETF meeting*, pages 1–4, Prague, 2011. `http://www.ietf.org/proceedings/80/slides/codec-4.pdf`.

[14] C. Hoene, J.-M. Valin, K. Vos, and J. Skoglund. Summary of opus listening test results. IETF Internet-Draft `http://tools.ietf.org/html/draft-ietf-codec-results`, 2012.

[15] ITU-R. *Recommendation BS.1116-1: Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems*, 1997.

[16] Peter H. Westfall and S. Stanley Young. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment.* Wiley Series in Probability and Statistics. John Wiley & Sons, New York, January 1993.

[17] Gian-Carlo Pascutto. Bootstrap. `http://www.sjeng.org/bootstrap.html`, 2011.

[18] D. Marston and A. Mason. Cascaded audio coding. EBU Technical Review, 2005.

[19] P. Kabal. An examination and interpretation of ITU-R BS.1387: Perceptual evaluation of audio quality. Technical report, TSP Lab, ECE Dept., McGill University, `http://www.TSP.ECE.McGill.CA/MMSP/Documents`, May 2002.

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