FFT-Based Digital Audio Compression

Scott Levine and Julius O. Smith III (jos@ccrma.stanford.edu)
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

March 31, 2019

- Subband Coding
- Transform Coding
- Princen-Bradley Filter Bank
- Dolby AC-2 and AC-3
- MPEG Audio Compression (MUSICAM)
- JPEG Image Compression

References

- M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice-Hall, 1995.
- H. Malvar and D. Staelin, "The lot: Transform coding without blocking effects", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 17, no. 4, pp. 553–559, Apr. 1989.
- J. P. Princen and A. B. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation", *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 34, no. 5, pp. 1153–1161, Oct. 1986.

Subband Coding

Quantize outputs of critically sampled filter bank

- If filter bank mimics hearing (e.g., constant-Q), quantization can be based on *auditory masking*
- Quantized filterbanks outputs can be *entropy coded*, e.g., Huffman
- FFT efficiently implements uniform filter bank (cf. Portnoff on implementing the phase vocoder using the FFT)

Transform Coding

Quantize outputs of critically sampled STFT

- Need $R = M = N$ for critical sampling (Hop size = Window length = DFT length) $\Longrightarrow$ rectangular window
- Quantization noise causes *discontinuities* in reconstruction due to rectangular window
- Need smooth post-window (synthesis filter) to hide frame-to-frame discontinuities, e.g., weighted overlap-add with $w(n) = \sqrt{\text{Hanning}(n)}$
- Smooth windows require at least 50% overlap $\Longrightarrow$ 200% initial data expansion
- Is there a better way?

Princen-Bradley Filter Bank

- Alternate DCT and DST using 50% OLA, constant-OLA window, and quarter-frame rotation
- $\mathrm{DCT}(x) \approx \text{re}\{\mathrm{FFT}(x)\} = \frac{X(\omega_k)+X^*(\omega_k)}{2} \longleftrightarrow \frac{x+\text{FLIP}(x)}{2}$
- Thus, DCT data is *time aliased* with its flip
- Similarly, $\mathrm{DST}(x) \approx \text{im}\{\mathrm{FFT}(x)\} = \frac{X(\omega_k)-X^*(\omega_k)}{2j} \longleftrightarrow \frac{x-\text{FLIP}(x)}{2}$
- Thus, DST data is *time aliased* with *minus* its flip
- Alternating DCT and DST in this way *cancels* aliasing
- This is "time-domain aliasing cancellation"
- Princen-Bradley filter bank = special case of "Lapped Orthogonal Transforms (LOT)" (see Malvar)
  - Let number of filter bank channels $= N$
  - Let length of each channel analysis filter be M
  - LOT = Critically sampled FIR filter bank with $M = 2N$

Dolby AC-2 and AC-3

- Original AC-2: fixed factor of 6 "transparent" compression for 44.1kHz 16-bit audio
- Now adjustable from 64 to 192 kilobits/sec/channel (ratios from 11 to 3.7 for 44.1kHz 16-bit audio)
- Mono algorithm (no use of stereo correlation)
- Can decode 2 channels in real time on 1 Motorola DSP5600x at 25MHz
- Uses Princen-Bradley Filterbank (DCT,DST)
- FFT can be used to compute DCT and DST for speed
- Nominal frame size = 512 samples at 44.1kHz (12ms)
- Second frame size (128) chosen for transients
- 256 FFT bins partitioned into 40 critical bands
- Masking pattern estimated
- One exponent per critical band (K. Brandenburg)
- Mantissa bit allocation based on signal to masking ratio

MUSICAM

MUSICAM = "Masking-pattern Universal Subband Integrated Coding and Multiplexing"

- Commonly referred to as "MPEG Audio"
- Compresses 44.1kHz 16-bit audio from 706 Kbits/sec down to around 128 Kbits/sec (ratio = 5.5)
- Quality is "transparent"
- Subband coder
  - 32-band uniform FIR filter bank
  - Uniformly spaced filters allow use of fast transform
  - Less delay than a dyadic constant-Q filter bank
  - Analysis filters are length 512 $\Longrightarrow$ length $512/32 = 16$ polyphase channel filters
- FFT used in parallel with filter bank
  - Masking pattern based on spectral power estimate
- No entropy coding

JPEG Image Compression

- Compresses individual images (no motion prediction as in MPEG)
- Baseline JPEG quantizes 2D DCT of 8×8 block of pixels
- Specialized, optimized FFT-like DCT transforms used
- Colors processed separately
- DCT blocks ordered in fixed "raster" pattern
- DCT approximates the Karhunen-Loeve transform (equal in the limit as transform size $\to \infty$)
- Compression ratio variable
- Progressive coding supported
  - Low-frequency DCT coefficients sent first
  - Higher frequency DCT coefficients sent later
- Hierarchical ("pyramidal") resolution coding supported (HF coding differential wrt LF)
- Lossless predictive coding also supported (no DCT)
- "Blocking" artifacts possible due to non-overlapping DCT blocks

- Uses entropy coding (Huffman)