Introduction

- Need to code digital audio efficiently and with quality.
  - Storage of high fidelity audio: CD, DAT, MD…
  - Interpersonal communication: video conferencing, multimedia applications...
  - Standards for HDTV.
  - Digital sound broadcasting
  - Two types of codecs:
    - Waveform coding (time domain).
    - Perceptual coding (frequency domain).
Outline

- Sampling techniques (PCM)
  - Linear
  - Non-linear
    - μ-Law
    - A-Law
- Generic Coding Techniques
  - DPCM
  - ADPCM
- Psychoacoustic Coding
  - MPEG

Sampling(1)

<table>
<thead>
<tr>
<th>Standard</th>
<th>BW (Hz)</th>
<th>Compression</th>
<th>Sample Rate (kHz)</th>
<th>Precision (bits)</th>
<th>Bit rate (kbps)</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMA-ADPCM</td>
<td>200-20000</td>
<td>ADPCM</td>
<td>8 - 44.1</td>
<td>4</td>
<td>32-350</td>
<td>Telephone, CD</td>
</tr>
<tr>
<td>G.711</td>
<td>200-3200</td>
<td>μ-law PCM</td>
<td>8</td>
<td>8</td>
<td>64</td>
<td>Telephone</td>
</tr>
<tr>
<td>G.722</td>
<td>50-7000</td>
<td>DPCM</td>
<td>16</td>
<td>4</td>
<td>64</td>
<td>AM Radio</td>
</tr>
<tr>
<td>G.728</td>
<td>200-3200</td>
<td>low-delay CELP</td>
<td>8</td>
<td>2</td>
<td>16</td>
<td>Telephone</td>
</tr>
<tr>
<td>G.723</td>
<td>200-3200</td>
<td>ADPCM</td>
<td>8</td>
<td>8</td>
<td>5.3 or 6.3</td>
<td>H.323</td>
</tr>
<tr>
<td>Audio CD</td>
<td>20-20000</td>
<td>Linear PCM</td>
<td>44.1</td>
<td>16</td>
<td>1411.2 (stereo)</td>
<td>CD</td>
</tr>
<tr>
<td>MPEG-1</td>
<td>20-20000</td>
<td>sub-band coding</td>
<td>32-48</td>
<td>2-15</td>
<td>256-384</td>
<td>near-CD</td>
</tr>
</tbody>
</table>

IMA: Interactive Multimedia Association
CELP: Code excited linear prediction
Chapter 5: Audio Compression

Sampling(2)

- Pulse Amplitude Modulation (PAM)
  - Each sample's amplitude is represented by 1 analog value.
- Sampling theory (Nyquist)
  - If input signal has maximum frequency (bandwidth) $f$, sampling frequency must be at least $2f$.
  - With a low-pass filter to interpolate between samples, the input signal can be fully reconstructed.

Pulse Code Modulation

- Each bit of resolution adds 6 dB of dynamic range.
- 16-bit samples has 96 dB SNR.
- Greater the number of quantization levels, lower the quantization noise
  - Tradeoff between quantization noise performance vs. data rate.
- Number of bits required depends on the amount of noise that is tolerated
  - $6.02N \approx SNR$
Uniform or Linear Quantization

- Quantization levels are evenly spaced.
- 16-bit samples provide plenty of dynamic range.
- CDs do this.
- Compression factor 1:1.
- File formats
  - .WAV (MS)
  - .AIFF (Unix and Mac)
- ...but SNR is worse for low-level signals than for high-level signals.

Nonuniform Quantization

- Humans are less sensitive to changes in “loud” sounds than “quiet” sounds
- Nonuniform quantization of the signal’s amplitude
  - Quantization step-size decreases logarithmically with signal level
  - Low-amplitude samples represented with greater accuracy than high-amplitude samples.
  - *Logarithmic-compressed quantizer.*
Chapter 5: Audio Compression

Non-linear Sampling

\[ f(x) = 127 \times \text{sign}(x) \times \frac{\ln(1 + \mu|x|)}{\ln(1 + \mu)} \]

(x normalized to [-1, 1])

\(\mu\)-Law ("mu-law") Companding
\(\mu\)-Law Companding

- Provides 14-bit quality (dynamic range) with an 8-bit encoding
- Used in North American & Japanese ISDN voice service
- Simple to compute encoding.
- Compression factor 2:1.
- File format
  - .au (Sun audio file format).

\(\mu\)-Law Encoding

<table>
<thead>
<tr>
<th>Input Amplitude</th>
<th>Step Size</th>
<th>Segment</th>
<th>Quantization</th>
<th>Code Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>0000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34(1)-35(2)</td>
<td>2</td>
<td>0001</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>62(29)-63(30)</td>
<td>4</td>
<td>0011</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>99(31)-107(34)</td>
<td>8</td>
<td>0111</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>124(91)-127(94)</td>
<td>16</td>
<td>1111</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>128(95)-135(102)</td>
<td>32</td>
<td>1111</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>248(215)-255(222)</td>
<td>64</td>
<td>1111</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>256(223)-271(238)</td>
<td>128</td>
<td>1111</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>496(463)-512(479)</td>
<td>256</td>
<td>1111</td>
<td>63</td>
<td>63</td>
</tr>
</tbody>
</table>

Codes the absolute value of the input (with bias of 33) in signed-magnitude.
**Outline**

- Sampling techniques (PCM)
  - Linear
  - Non-linear
    - μ-Law
    - A-Law

- Generic Coding Techniques
  - DPCM
  - ADPCM

- Psychoacoustic Coding
  - MPEG
Difference Encoding

- Exploit temporal redundancy in samples.
- Difference between 2 x-bit samples can be represented with significantly fewer than x-bits.
- Transmit the difference (rather than the sample).
- Differential-PCM (DPCM)

DPCM

Signal to be encoded

\[ x \]

\[ \Delta x = x - \hat{x} \]

Prediction Difference/Error

Quantizer

\[ \Delta x \]

\[ \Delta \tilde{x} \]

Entropy Coder

\[ x \text{-bit DPCM “difference”} \]

Predictor

\[ \tilde{x} \]

\[ \hat{x} \]

Reconstructed Signal

\[ \tilde{x} = \tilde{x} + \Delta \tilde{x} \]

Simple difference

\[ \tilde{x}(n) = \tilde{x}(n-1) \]

Linear predictor

\[ \tilde{x}(n) = \sum_{k=0}^{N} h_k \tilde{x}(n-k) \]

PCM

\[ \hat{x}(n) = 0 \]
Slope Overload Problem (1)

- Prediction differences $x(n)$ are too large for the quantizer to handle. Encoder fails to track rapidly changing signals.
  - Especially near the Nyquist frequency.
- Differences in frequency cannot be represented with a smaller number of bits!

Slope Overload Problem (2)

- Error introduced leads to severe distortion in the higher frequencies

\[ \hat{x}(n) = \bar{x}(n-1) \]
\[ \tilde{x}(n) = \bar{x}(n-1) + \Delta x^v(n) \]
Adaptive DPCM (ADPCM)

- Use a larger step-size to encode differences between high-frequency samples & a smaller step-size for differences between low-frequency samples.
- Use previous sample values to estimate changes in the signal in the near future.

ADPCM

- To ensure differences are always small...
  - Adaptively change the step-size (quanta).
  - (Adaptively) attempt to predict next sample value.
Outline

- Sampling techniques (PCM)
  - Linear
  - Non-linear
    - μ-Law
    - A-Law
- Generic Coding Techniques
  - DPCM
  - ADPCM
- Psychoacoustic Coding
  - MPEG

Psychoacoustic Principles

- Based on extensive studies of human perception.
- Average human does not hear all frequencies the same way.
- Limitations of the human sensory system leads to facts that can be used to cut out unnecessary data in an audio signal.
- Two main properties of the human auditory system:
  - Absolute threshold of hearing
  - Auditory masking
Absolute Threshold of Hearing

\[ T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/100)^4 \]

- Range is about 20 Hz - 20 kHz, most sensitive at 2 kHz to 4 kHz.
- Dynamic range (quietest to loudest is about 96 dB)
- Normal voice range is about 500 Hz - 2 kHz.

Simultaneous Masking

- The presence of tones at certain frequencies makes us unable to perceive tones at other “nearby” frequencies
  - Humans cannot distinguish between tones within 100 Hz at low frequencies and 4 kHz at high frequencies.
**Chapter 5: Audio Compression**

**Masking Threshold**

Approximates a triangular function modeled by spreading function, $SF(x)$ (db), where $x$ has units of bark:

$$SF(x) = 15.81 + 7.5(x + 0.474) - 17.5\sqrt{1+(x + 0.474)^2}$$

**Simultaneous Masking Example**

Listen to these played together:

- **Signal**
- **Noise**
- **Signal + Noise (SNR = 24 dB)**
Critical Bands(1)

- Human auditory
  - Limited and frequency dependant resolution
  - 25 critical bands
- Bark: Barkhausen
  - 1 Bark = width of one critical band
  - Critical band number (Bark) for a given frequency, \( z(f) \):
    - \( f < 500 \text{Hz} \Rightarrow z(f) = f/100 \)
    - \( f > 500 \text{Hz} \Rightarrow z(f) = 9 + 4 \log(f/1000) \)
  - Also given as
    \[
    z(f) = 13.0 \arctan(0.76 f) + 3.5 \arctan(f^2/56.25)
    \]

Critical Bands(2)

<table>
<thead>
<tr>
<th>Band</th>
<th>Lower Bound (Hz)</th>
<th>Center (Hz)</th>
<th>Upper Bound (Hz)</th>
<th>Bandwidth (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>50</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>150</td>
<td>290</td>
<td>140</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>250</td>
<td>390</td>
<td>140</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>350</td>
<td>490</td>
<td>140</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>450</td>
<td>590</td>
<td>140</td>
</tr>
<tr>
<td>6</td>
<td>610</td>
<td>550</td>
<td>690</td>
<td>140</td>
</tr>
<tr>
<td>7</td>
<td>730</td>
<td>650</td>
<td>790</td>
<td>140</td>
</tr>
<tr>
<td>8</td>
<td>850</td>
<td>750</td>
<td>890</td>
<td>140</td>
</tr>
<tr>
<td>9</td>
<td>990</td>
<td>890</td>
<td>990</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>1090</td>
<td>990</td>
<td>1190</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>1190</td>
<td>1090</td>
<td>1290</td>
<td>100</td>
</tr>
<tr>
<td>12</td>
<td>1290</td>
<td>1190</td>
<td>1390</td>
<td>100</td>
</tr>
<tr>
<td>13</td>
<td>1390</td>
<td>1290</td>
<td>1490</td>
<td>100</td>
</tr>
<tr>
<td>14</td>
<td>1490</td>
<td>1390</td>
<td>1590</td>
<td>100</td>
</tr>
<tr>
<td>15</td>
<td>1590</td>
<td>1490</td>
<td>1690</td>
<td>100</td>
</tr>
<tr>
<td>16</td>
<td>1690</td>
<td>1590</td>
<td>1790</td>
<td>100</td>
</tr>
<tr>
<td>17</td>
<td>1790</td>
<td>1690</td>
<td>1890</td>
<td>100</td>
</tr>
<tr>
<td>18</td>
<td>1890</td>
<td>1790</td>
<td>1990</td>
<td>100</td>
</tr>
<tr>
<td>19</td>
<td>1990</td>
<td>1890</td>
<td>2090</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>2090</td>
<td>1990</td>
<td>2190</td>
<td>100</td>
</tr>
<tr>
<td>21</td>
<td>2190</td>
<td>2090</td>
<td>2290</td>
<td>100</td>
</tr>
<tr>
<td>22</td>
<td>2290</td>
<td>2190</td>
<td>2390</td>
<td>100</td>
</tr>
<tr>
<td>23</td>
<td>2390</td>
<td>2290</td>
<td>2490</td>
<td>100</td>
</tr>
<tr>
<td>24</td>
<td>2490</td>
<td>2390</td>
<td>2590</td>
<td>100</td>
</tr>
<tr>
<td>25</td>
<td>2590</td>
<td>2490</td>
<td>2690</td>
<td>100</td>
</tr>
</tbody>
</table>

Chapter 5: Audio Compression
Temporal Masking

- If we hear a loud sound, then it stops, it takes a little while until we can hear a soft tone nearby.
  - As much as 50 ms before and 200 ms after.
- Example:
  - Play 1 kHz masking tone at 60 dB and 1.1 kHz test tone at 40 dB.

![Temporal Masking Diagram](image)

Global Masking Threshold

- Additive process
  - The minimum level of audibility in the presence of masking noise
  - Time varying function of frequency at each frequency at a given time

![Global Masking Threshold Diagram](image)
Summary

- Auditory masking
- Range of masking is 1 Bark (or critical band)
- Two factors for masking - frequency masking and temporal masking.
- How can we use this information for compression?

MPEG-1 Audio

  - Audio and video standard for encoding and decoding.
  - Specify general functionality of application.
- MPEG-1: 1.5 Mbps for audio and video
  - 1.2 Mbps for video
  - 0.3 Mbps for audio
    - Uncompressed CD audio - 44,100 samples/sec * 16 bits/sample * 2 ch > 1.4 Mbps
- Compression ratio from 2.7:1 to 42:1
  - 16 bit stereo sampled at 48 kHz is reduced to 256 kbps
- Sampling frequency
  - 32, 44.1 and 48 kHz
- One or two audio channels
  - Monophonic, Dual-monophonic, Stereo, Joint Stereo
MPEG-1 Audio Layers

- **Layer 1**
  - DCT type filter with one frame and equal frequency spread per band.
  - Psychoacoustic model only uses frequency masking.

- **Layer 2**
  - Use three frames in filter (before, current, next, a total of 1152 samples).
  - This models a little bit of the temporal masking.

- **Layer 3**
  - Better critical band filter is used (non-equal frequencies)
  - Psychoacoustic model includes temporal masking effects, and takes into account stereo redundancy.
  - Huffman coder.
  - *Known as MP3*

MPEG Codec Block Diagram

[Diagram of MPEG Codec Block Diagram]
Sub-band Filtering

- 32 PCM samples yields 32 subband samples.
  - Each sub-band corresponds to a frequency band evenly spaced from 0 to Nyquist frequency.
  - For example, @48 kHz sampling rate, each sub-band is 750 Hz wide.
- Samples out of each filter are grouped into blocks, called frames.
  - Blocks of 12 for Layer 1 (384 samples).
  - Blocks of 36 for Layers 2 and 3 (1152 samples).
- @48 kHz, 32x12 represents 8 ms of audio.

Subbands vs. Critical Bands

- 32 constant-width subbands do not accurately reflect the ear’s critical bands
Psychoacoustic Model

- **FFT** gets detailed spectral information about the signal.
  - 512-point FFT for Layer 1
  - 1024-point for Layer 2 and 3
- From each subband’s samples, separate **tonal** (sinusodial) and **nontonal** (noise) maskers in the signal.
- Determine **masking threshold** for each sub-band.
- Determine the **global masking threshold**.
- Determine the **minimal masking threshold** in each sub-band.
- Calculate the **signal-to-mask ratio (SMR)** in each sub-band, which is the ratio of signal energy to minimum masking threshold.

Spectral Analysis: FFT

- Transforms PCM samples from time to frequency domain.
- Needs finer frequency resolution for an accurate calculation of the masking thresholds.
- Incoming audio sample $s(n)$ normalized according to FFT length $N$.
  \[ x(n) = \frac{s(n)}{N^{2b-1}} \]
  - $N$ - FFT length
  - $b$ - number of bits per sample
- Segment the signal into 512 samples => 12 ms frames @44.1kHz
Discrete Fourier Transform

- Spectral Analysis requires Time ↔ Frequency transform.
- Discrete Fourier Transform (DFT) provides uniformly spaced samples of the Discrete-Time Fourier Transform (DTFT).
- DFT and Inverse-DFT
  \[ X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2 \pi n k}{N}} \quad k = 0, \ldots, N - 1 \]
- Euler’s formula
  \[ e^{-jx} = \cos x + j \sin x \]

Time vs. Frequency Interpretation

\[ H(\Omega) \]

\[ h(t) \]
Fast Fourier Transform (1)

- DFT has $O(N^2)$ complexity.
- Fast Fourier Transform (FFT) is a faster algorithm with $O(N \log N)$ complexity.
- Basic idea => separate $X[k]$ into even, $Y[k]$, and odd-numbered, $Z[k]$, samples.
  
  \[ X[k] = Y[k] + e^{-j \frac{2\pi k}{N}} Z[k] \quad \text{for} \quad k = 0, 1, \ldots, N/2 - 1 \]
  \[ X[k + N/2] = Y[k] - e^{-j \frac{2\pi k}{N}} Z[k] \quad \text{for} \quad k = 0, 1, \ldots, N/2 - 1 \]

- N-point DFT computations are broken into two (N/2)-point DFTs.
- (N/2)-point DFTs can be further broken into two (N/4)-point DFTs, and so on…

Fast Fourier Transform (2)

- DFT has $O(N^2)$ complexity.
- Fast Fourier Transform (FFT) is a faster algorithm with $O(N \log N)$ complexity.

\[
X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi k n}{N}}
\]

\[
= \sum_{n=0}^{\frac{N}{2}-1} x[2n] e^{-j \frac{2\pi k n}{N}} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1] e^{-j \frac{2\pi k (2n+1)}{N}}
\]

\[
= \sum_{n=0}^{\frac{N}{2}-1} x[n] e^{-j \frac{2\pi k n}{N/2}} + e^{-j \frac{2\pi k}{N}} \sum_{n=0}^{\frac{N}{2}-1} z[n] e^{-j \frac{2\pi k n}{N/2}}
\]

- DFTs of even sample
- DFTs of odd sample
Fast Fourier Transform (3)

\[ Y[k] = \sum_{n=0}^{N-1} y[n] e^{-j2\pi k \frac{n}{N}} \]
\[ Z[k] = \sum_{n=0}^{N-1} z[n] e^{-j2\pi k \frac{n}{N}} \]

\[ X[k] = Y[k] + e^{-j\frac{2\pi k}{N} Z[k]} \text{ for } k = 0,1,\ldots, \frac{N}{2} - 1 \]

First half

\[ X[k + N/2] = Y[k] - e^{-j\frac{2\pi k}{N} Z[k]} \text{ for } k = 0,1,\ldots, \frac{N}{2} - 1 \]

Second half

Fast Fourier Transform (4)

\[ X[k + N/2] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi k \frac{n}{N}} \]
\[ = \sum_{a=0}^{N/2-1} x[2a] e^{-j2\pi k \frac{2a}{N}} + \sum_{a=0}^{N/2-1} x[2a+1] e^{-j2\pi k \frac{2a+1}{N}} \]
\[ = \sum_{a=0}^{N/2-1} x[2a] e^{-j2\pi \frac{k}{N} (2a)} e^{-j\pi \frac{k}{N}} + \sum_{a=0}^{N/2-1} x[2a+1] e^{-j2\pi k \frac{2a+1}{N}} \]
\[ = \sum_{a=0}^{N/2-1} x[2a] e^{-j2\pi \frac{k}{N} (2a)} - \sum_{a=0}^{N/2-1} x[2a+1] e^{-j2\pi \frac{k}{N} (2a+1)} \]
\[ = \sum_{a=0}^{N/2-1} y[n] e^{-j2\pi \frac{k}{N} \frac{n}{2}} + \sum_{a=0}^{N/2-1} z[n] e^{-j2\pi \frac{k}{N} \frac{n}{2}} Z[k] \]

Chapter 5: Audio Compression
Fast Fourier Transform (5)

DFTs of even samples


\[ Z[0] \quad W^0_Z \quad Z[1] \quad W^1_Z \]

\[ Z[2] \quad W^2_Z \quad Z[3] \quad W^3_Z \]

DFTs of odd samples


\[ W^k_N = e^{-j \frac{2\pi k}{N}} \]

Stage 3

8-point FFTs

\[ X[k] = Y[k] + e^{-j \frac{2\pi k}{N}} Z[k] \]

\[ X[k + N/2] = Y[k] - e^{-j \frac{2\pi k}{N}} Z[k] \]

Stage 1

2-point FFTs

Stage 2

4-point FFTs

Stage 3

8-point FFTs

Fast Fourier Transform (6)
Data Windowing(1)

- DFT inherently assumes that data is a single period of a periodically repeating waveform!
- Sampled values of the signal are multiplied by a (window) function which tapers toward zero at either end. The sampled signal, rather than starting and stopping abruptly, "fades" in and out.
- This reduces the effect of the discontinuities where the mismatched sections of the signal join up.

\[
\text{Hann Window} \quad w[n] = \frac{1}{2} \left[ 1 - \cos \left( \frac{2 \pi n}{N} \right) \right]
\]

Data Windowing(2)
Data Windowing (3)

- Overlapped window: 512 samples (12 ms frame) and 448 samples (10.9 ms)

Spectral Analysis: PSD

- Power Spectral Density, $P(k)$, using 512-point FFT ($N=512$)

$$P(k) = 90.302 \text{ dB} + 10 \log \left( \sum_{n=0}^{N-1} w(n)x(n)e^{-j\frac{2\pi kn}{N}} \right)^2 \text{ dB}$$

- Generates $N/2$ frequency components or bins.
Identify Tonal and Noise Maskers (1)

- **Tonal maskers** are signals that generate pure tone, i.e., harmonically rich.
- **Noise masker** has no single dominant freq., i.e., more noise like.
- Tonal and noise maskers have different masking characteristics.
- Component considered tonal if \( P(k) - P(k \pm \Delta_k) \geq 7 \text{ dB} \), where \( \Delta_k \) is Bark distance.
  - \( \Delta_k \) varies as function of frequency range

CD-quality pop music sampled @44.1 kHz, 16 bits per sample
Identify Tonal and Noise Maskers (2)

- **Tonal maskers**, $P_{TM}(k)$

  \[ P_{TM}(k) = 10 \log_{10} \sum_{j=-1}^{1} 10^{0.1P(k+j)} \]

  - $P(k)$ - Output of FFT
  - Energies from 3 adjacent spectral components centered around the peak are combined.

- **Noise maskers**, $P_{NM}(k)$

  \[ P_{NM}(k) = 10 \log_{10} \sum_{j} 10^{0.1P(j)} \]

  - Energies from all spectral components within each critical band are combined.

Decimation & Reorganization of Maskers

- Only retain maskers that satisfy
  \[ P_{TM,NM}(k) \geq T_q(k) \]

- Sliding 0.5 Bark-window is used to replace any pair of maskers occurring within a distance 0.5 Bark by the stronger of the two.
Calculation of Individual Masking Threshold(1)

Piecewise linear function and approximates the spreading function

\[ SF(x) = 15.81 + 7.5(x + 0.474) - 17.5\sqrt{1 + (x + 0.474)^2} \]

\[
SF(i,j) = \begin{cases} 
17\Delta_z - 0.4P_{TM,NM}(j) + 11, & -3 \leq \Delta_z < -1 \\
(0.4P_{TM,NM}(j) + 6)\Delta_z, & -1 \leq \Delta_z < 0 \\
-17\Delta_z, & 0 \leq \Delta_z < 1 \\
(0.15P_{TM,NM}(j) - 17)\Delta_z - 0.15P_{TM,NM}(j), & 1 \leq \Delta_z < 8 \\
\end{cases}
\]

\[ \Delta_z = z(i) - z(j) \]

Calculation of Individual Masking Threshold(2)

- **Tonal masking threshold,** \( T_{TM}(i,j) \)

\[
T_{TM}(i,j) = P_{TM}(j) - 0.275z(j) + SF(i,j) - 6.025
\]

- \( P_{TM}(j) \) - SPL of the tonal masker in frequency bin \( j \)
- \( z(j) \) - Bark frequency of frequency bin \( j \)
- \( SF(i,j) \) - spread of masking from bin \( j \) to maskee bin \( i \)

- **Noise masking threshold,** \( T_{NM}(i,j) \)

\[
T_{NM}(i,j) = P_{NM}(j) - 0.175z(j) + SF(i,j) - 2.025
\]

- \( P_{NM}(j) \) - SPL of the tonal masker in frequency bin \( j \)
- \( z(j) \) - Bark frequency of frequency bin \( j \)
- \( SF(i,j) \) - spread of masking from bin \( j \) to maskee bin \( i \)
Individual Masking Threshold Example

Calculation of Global Masking Threshold

- **Global masking threshold**, $T_g(i)$

$$T_g(i) = 10 \log \left( 10^{0.1T_q(i)} + \sum_{l=1}^L 10^{0.1T_m(i,l)} + \sum_{m=1}^M 10^{0.1T_n(i,m)} \right)$$

- $L$ - number of tonal maskers
- $M$ - number of noise maskers
Global Masking Threshold Example

\[ P_{\text{mask}}(k) \leq T_q(k) \]

Chapter 5: Audio Compression 59

Spread of Masking

- A masker centered within one critical band has some predictable effect on detection thresholds in other critical bands.
Masking Model Example

- Suppose the levels of the first 16 of the 32 sub-bands are:

<table>
<thead>
<tr>
<th>Band</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level (dB)</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>60</td>
<td>35</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

- The level of the 8th band is 60 dB, the pre-computed masking model specifies a masking of 12 dB in the 7th band and 15 dB in the 9th.
  - The signal level in 7th band is 10 (< 12 dB), so ignore it.
  - The signal level in 9th band is 35 (< 15 dB), so send it.
- Only the signals above the masking level needs to be sent.

Scaling

- Block of 12 samples for each subband is scaled to normalise the peak signal level within a subband.
  - Largest signal quantized using 6-bit scale-factor.
- The receiver needs to know the scale factor and quantisation levels used.
  - Information included along with the samples
- The resulting overhead is very small compared with the compression gains.
Bit Allocation

- For each audio frame, bits must be distributed across the sub-bands from a predetermined number of bits defined by the target bit rate.
  - 192 kbps target rate => 8 ms/frame @48KHz => ~1.5 kbits/frame.
- Objective is to minimize noise-to-mask ratio (NMR) over all sub-bands, i.e., minimize quantization noise.
- Many different variations and optimizations.

Quantization Noise

- Equivalent to $x[n] = Q[x[n]] + e[n]$
- $\text{SNR} = 20\log(x[n]/e[n])$
  - More bits allocated to quantization, smaller $e[n]$!
**Bit Allocation**

Iterative process:
- For each sub-band, determine \( NMR = SNR - SMR \)
- Allocate bits one at a time to the sub-band with the lowest \( NMR \). Recalculate \( NMR \) values.
- Iterate until all bits used.

**SNR**

\[
SNR = 20 \log \left( \frac{V}{V_{QN}} \right) = 20 \log \frac{2^{N-1}}{1/2} = 6.02N \text{(dB)}
\]

Each bit increases resolution by \(-6\text{dB}\)

**Bitstream Organization**

- **Frame Header** (32 bits)
  - Syncword
  - Bit-rate and sampling information.
  - Number of channels (1 or 2)
  - Misc. other stuff.

- **Encoded Samples**
  - 384 for Layer 1, 1152 for Layers 2 and 3

- **Side Info**
  - Encoding parameters (Layer specific)
Frame Header

- **Sampling rate frequency index**
  - 00 - 44.1 kHz
  - 01 - 48 kHz
  - 10 - 32 kHz
  - 11 - reserved

- **Pad bit**
  - 0 - Frame is not padded
  - 1 - Frame is padded with one extra slot

- **Private bit**

- **Channel Mode**
  - 00 - Stereo
  - 01 - Joint stereo (Stereo)
  - 10 - Dual channel (2 mono channels)
  - 11 - Single channel (Mono)

- **MPEG Audio version ID**
  - 00 - MPEG Version 2.5 (later extension of MPEG 2)
  - 01 - reserved
  - 10 - MPEG Version 2 (ISO/IEC 13818-3)
  - 11 - MPEG Version 1 (ISO/IEC 11172-3)

- **Protection bit**
  - 0 - Protected by CRC (16-bit CRC follows header)
  - 1 - Not protected

- **Bit rate index**
  - Layer 1: 32 - 448 kbps
  - Layer 2: 32 - 384 kbps
  - Layer 3: 32 - 320 kbps

- **Layer description**
  - 00 - reserved
  - 01 - Layer III
  - 10 - Layer II
  - 11 - Layer I

- **Sampling rate frequency index**
  - 00 - 44.1 kHz
  - 01 - 48 kHz
  - 10 - 32 kHz
  - 11 - reserved

- **Pad bit**

- **Private bit**

- **Misc.**
  - Copyright
  - Original

Frame Structure

<table>
<thead>
<tr>
<th>(0xff00)</th>
<th>sync</th>
<th>Framelen</th>
<th>Bit-allocation</th>
<th>Scale-factors</th>
<th>Audio Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit</td>
<td>16-bit</td>
<td>4x32=128-bit</td>
<td>6xN=&gt;192-bit (max.)</td>
<td>12 samples x N</td>
<td></td>
</tr>
</tbody>
</table>

- **Framelen** - frame length in bytes;
- **Bit-allocation** - 4 bits allocated to each subband (0-31).
  - 2-15 bits, ‘0000’ indicates no sample.
- **Scale-factors** - 6 bits
  - Multiplier that sizes the samples to fully use the range of the quantizer
  - Has variable length depending on \( N \) = the number of subbands with non-zero bit allocation.
- **Audio Data** - subbands 0-31 stored in order, with each subband including 12 samples
MPEG Audio Layer 3 (MP3)

- Layer 1
  - 384 samples per frame
  - Up to 448 kbps
- Layer 2
  - 1152 samples per frame
  - Up to 384 kbps
- Layer 3 (MP3)
  - Range of bit-rates from 8 kbps to 320 kbps
  - Better filterbank
  - Improved psychoacoustic model
  - Better bit-allocation process
  - Entropy coding
  - ...

MP3 Block Diagram
Improvements

- Equally spaced Sub-bands do not accurately reflect the ear’s critical bands.
  - \( f < 500\text{Hz} \Rightarrow z(f) \approx f/100 \)
  - \( f > 500\text{Hz} \Rightarrow z(f) \approx 9 + 4 \log_2 (f/1000) \)
- Each sub-band further analyzed using Modified DCT (MDCT) to create 18 samples (for total of 576 samples).
  - Increases the potential for redundancy removal.
  - Better tracking of masking threshold.
- MP3 also specifies a MDCT block length of 6.
- Lots of bit allocation options for quantizing frequency coefficients.
- Quantized coefficients Huffman coded.

Effectiveness of MPEG Audio

<table>
<thead>
<tr>
<th>Layer</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1.5</td>
</tr>
<tr>
<td>II</td>
<td>2...4</td>
</tr>
<tr>
<td>III</td>
<td>&gt; 7.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer</th>
<th>Target bit rate</th>
<th>Ratio</th>
<th>Quality @ 64 kbits</th>
<th>Quality @ 128 kbits</th>
<th>Theoretical Min. Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>192 kbps</td>
<td>4:1</td>
<td>-</td>
<td>-</td>
<td>19 ms</td>
</tr>
<tr>
<td>II</td>
<td>128 kbps</td>
<td>6:1</td>
<td>2.1 to 2.6</td>
<td>4+</td>
<td>35 ms</td>
</tr>
<tr>
<td>III</td>
<td>64 kbps</td>
<td>12:1</td>
<td>3.6 to 3.8</td>
<td>4+</td>
<td>59 ms</td>
</tr>
</tbody>
</table>

5: perfect, 4: just noticeable, 3: slightly annoying, 2: annoying, 1: very annoying