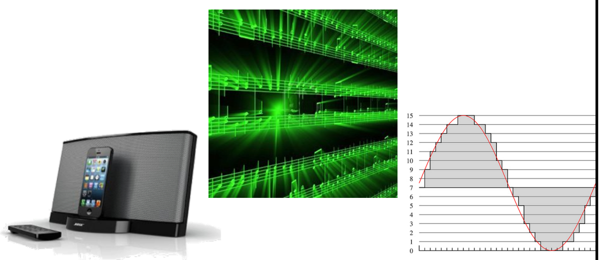


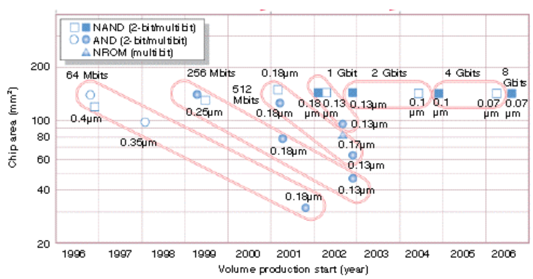
Lecture #11 – Psychoacoustic Model/Compression/MP3

ESE 150 – Digital Audio Basics

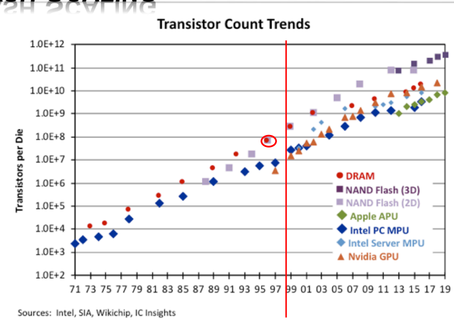
First MP3 Player

- MpMan -- 1998
- SaeHan Information Systems
  - South Korea
- 32MB of Flash memory

Flash Scaling



Flash Scaling



https://www.icinsights.com/news/bulletins/Transistor-Count-Trends-Continue-To-Track-With-Moores-Law/

PCM and CD Audio

- **PCM – Pulse Code Modulation**
  - CD-Quality Audio – state-of-the-art 1990s
    - Filtering/Sampling/Quantizing/Encoding using ADC
    - DAC/Reconstruction Filter
  - CD Quality Digital Audio uses PCM (uncompressed, lots of storage!)
    - 44,100 samples per second, each sample 16-bits
    - 1 sec. of music: 44,100 x 16bits = 705,600 bits or 86 kB
    - 60 seconds of music: 705,600 x 60 = 42,336,000 bits = 5167 kB = 5 MB
    - 3 minute song: 42,336,000 x 3 = 127,008,000 bits = 15 MB !
    - You want it in stereo??? 15 MB x 2 = 30 MB! (no compression!)

Observe

- **If we kept the CD Audio encoding format**
  - Could hold **one** song on the 1998 MpMan
  - (maybe 2 on the 64MB version)
- **For solid-state audio to be viable**
  - Needed more compact encoding for music

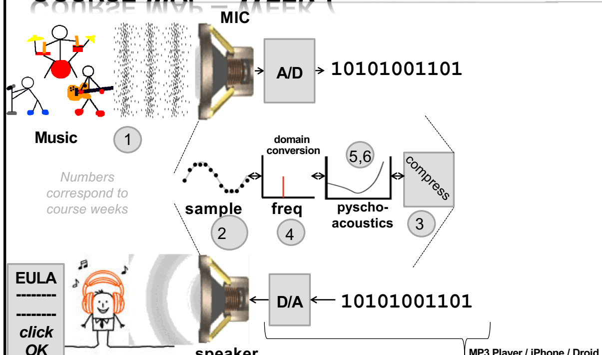
First MP3 Player

- MpMan -- 1998
- SaeHan Information Systems
  - South Korea
- 32MB of Flash memory
- Held 6 songs (MP3)
- (12 on 64MB version)
- 3 years before Apple iPod
  - October 2001
  - Initially hard disk
- Diamond Rio later 1998

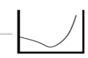
Lecture Topics

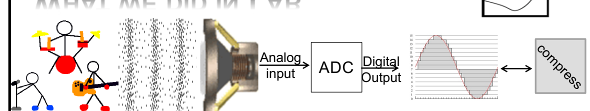
- Teaser
- Where are we?
- Preclass
- How do we take advantage of psychoacoustics in MP3 Achieve this 6—12x reduction from CD Audio
  - Review Tricks
  - Formulate Optimization
  - Adaptation (next Friday)
- Next Lab
- References

Course Map – Week 7



What we did in Lab…





- Week 1: Converted Sound to analog voltage signal
  - a "pressure wave" that changes air molecules w/ respect to time
  - a "voltage wave" that changes amplitude w/ respect to time
- Week 2: Sampled voltage, then quantized it to digital sig.
  - Sample: Break up independent variable, take discrete 'samples'
  - Quantize: Break up dependent variable into n-levels (need 2^n bits to digitize)
- Week 3: Compress digital signal
  - Use even less bits without using sound quality!
- Week 4: Before we compress…
  - Put our 'digital' data into another form…BEFORE we compress…less stuff to compress!
- Week 5: Psychoacoustics
  - Measured limits of human hearing; measured masking

Preclass

Preclass

- 4 critical bands
- 10 frequencies
- 16b amplitude
- Preclass 1: Bits to represent (no further encode)?

Preclass

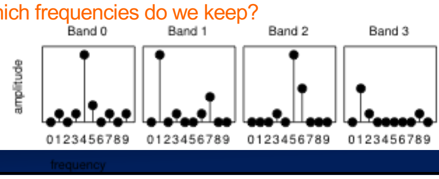
- 4 critical bands
- 10 frequencies
- 16b amplitude
- 107b encoding budget
- Preclass 2: amplitude quantization necessary to achieve budget?
- Preclass 3: frequencies (reduced sampling rate) can keep to achieve budget?

Preclass

- 4 critical bands
- 10 frequencies
- 16b amplitude
- 107b encoding budget
- Preclass 4
  - Bits to represent which frequency?
  - Bits to encode (frequency, amplitude) pair?
  - Number of (frequency,amplitude) pairs fit within budget?

Preclass

- 4 critical bands
- 10 frequencies
- 16b amplitude
- 107b encoding budget
- Preclass 5
  - Which frequencies do we keep?



Preclass

- 4 critical bands
- 10 frequencies
- 16b amplitude
- 107b encoding budget
- Preclass 6: which likely to sound best?
  - Amplitude quantization
  - Frequency quantization (reduce sampling rate)
  - Frequency selection based on masking

Tricks for Compression

- Quantization
- Sampling Rate / Frequency Quantization
- Critical Band Masking
  - Selective frequency dropping

Background

The MPEG-1 Standard

- **ISO (International Standards Organization)**
  - Looking for ways to reduce transmission requirements for digital video and audio (low bandwidth transmission of digital media)
  - 1988 – establishes a sub-committee of ISO:
    - *Moving* Picture Experts Group (MPEG)
    - Goal: Develop common standard for coding/compressing audio/video
      - To reduce size of data to transmit without sacrificing quality
      - Fraunhofer Institute and German University of Erlangen
        - Lots of basic research in Digital Audio Broadcast, tapped to be MPEG
  - Result: 1992: Finalized Standard called: MPEG-1 (Phase I)
    - 3 Parts: Audio/Video/System
    - Audio component: defined 3 layers: 1, 2, 3
      - Increasing levels of compression and complexity
      - MPEG-1, layer 3 achieves 12:1 compression ratio! *(for short MP3)*

The MPEG-1 Standard

- **MPEG-1, (3) Layers for Audio Coding:**

Complexity ↓

Coding	Ratio	Required bitrate
PCM CD Quality	1:1	1.4 Mbps
Layer I	4:1	384 kbps
Layer II	8:1	192 kbps
Layer III (MP3)	12:1	128 kbps

- **Why is PCM CD Quality 1.4 Mbps?**
  - Recall: 1 sec. of music: 44,100 x 16bits = 705,600 bits
  - Don't forget stereo (R/L): 2 x 706,600 = 1,413,200 (1.4Mbs)
    - Defines bandwidth requirements of network
  - Notice: 128 kbps was just about double modem speed in 1992
    - Enables transmission of audio (MP3) via modem!

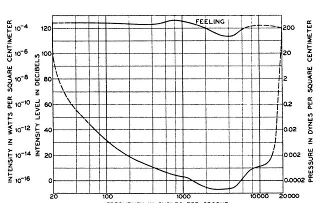
Optimizing Encoding

Knobs We Can Turn

- **Amplitude quantization**
- **Frequency quantization**
- **Frequencies kept (per critical band)**
  - Too soft
  - Masked
- **…and can perform lossless compression**

Bands Vary in Importance

- **Not equally sensitive across bands**
- **If quantize bands differently, where want finer resolution?**
- **Where tolerate more quantization?**



Knobs We Can Turn

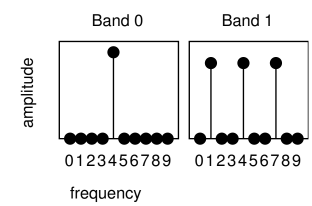
- **Amplitude quantization**
  - Per band
- **Frequency quantization**
  - Per band?
- **Frequencies kept (per critical band)**
  - Per band
- **…and can perform lossless compression**

Opportunities/Challenges

- **Trying to hit fixed bit rate, what challenge does lossless compression impose?**
  - Encounter many common frequencies, amplitudes?
  - Encounter many uncommon frequencies, amplitudes?

Opportunities/Challenges

- **What challenge/opportunities might these band spectra represent?**



Suggest

- **May want to do something smarter than**
  - Allocating fixed number of frequencies per band
  - Allocating fixed quantization to a band
- **Like to adapt our encoding to the data**
  - If more Huffman compressible, we get more frequencies
  - If fewer frequencies suffice for one band,
    - Allow more frequencies for another
    - …or allocate less quantization

Optimization Problem

- **How fit in the resource constraints (128Kb/s) while maximizing goodness (sound quality)?**
- **Optimization problems central to engineering**

Optimization Problem

- **How fit in the resource constraints (128Kb/s) while maximizing goodness (sound quality)?**
- **Quantify bits used** $\sum_{bands}\sum_{f \in freqs} Bits(f)$
- **Quantify goodness**

$$\sum_{f \in freqs} Error(f) \times W(f)$$

Quantifying Bit cost

- **Simple, fixed sample: Frequencies × Bits/freq**
- **Fixed frequencies per Band:**
  - Bands × (Frequencies/Band) × Bits/freq
- **Variable Frequencies per Band:**

$$\sum_{b \in bands} frequencies(b) \times (\text{bits/freq})$$

- **Variable Frequencies and quantization per Band:**

$$\sum_{b \in bands} frequencies(b) \times bit(b)$$

- **Huffman means different bits/frequency** $\sum_{bands}\sum_{f \in freqs} Bits(f)$

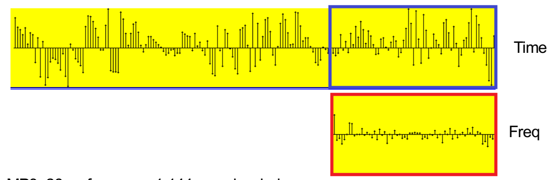
Goodness/Sound Quality

- **Error(Amp) = |Orig Amplitude – Encoded|**
  - Whole OrigAmplitude if dropped
  - |Orig Amplitude-Quantize(OrigAmplitude,bits)| if quantized
- **W(freq)**
  - 0 if below hearing threshold
  - 0 if masked
  - Value between 0 and 5 if partially masked in critical band
    - More critical bands, higher
  - Really depend on what already encoded

$$\sum_{f \in freqs} Error(f) \times W(f)$$

Reminder: A Window Operation

- **In the example below, we traverse the signal but only look at 64 samples at a time**



MP3: 26ms frames or 1,144 sample window

(sepwww.stanford.edu/oldsep/hale/FftLab.html)

Optimization Problem

- **How fit in the resource constraints (128Kb/s) while maximizing goodness (sound quality)?**
- **Quantify bits used:**
  - Cannot exceed 128Kb/s
  - = 3,408 b / 26ms frame

$$\sum_{bands} \sum_{f \in freqs} Bits(f)$$

- **Quantify goodness: minimize**

$$\sum_{f \in freqs} Error(f) \times W(f)$$

(as time permits)

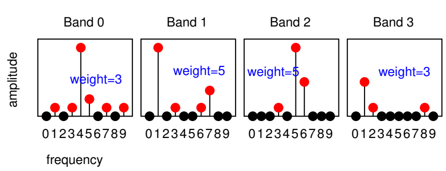
Approach

Approach

$$\sum_{bands} \sum_{f \in freqs} Bits(f)$$

- **Start with nothing**
- **Start with bitbudget**

$$\sum_{f \in freqs} Error(f) \times W(f)$$



Example Weight Function W(f)

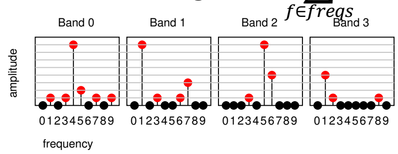
- **W(f)=CBWeight*Mask**
- **Mask = 0 if MaxAmp-FreqAmp<4**
- **(MaxAmp-FreqAmp)/MaxAmap otherwise**

Approach

$$\sum_{bands}\sum_{f \in freqs} Bits(f)$$

- Start with nothing
- Start with bitbudget

$$\sum_{f \in freqs} Error(f)\times W(f)$$



Bits = 0
Error = 190

	11 8 2 11	8 1 1 3	1 8 4	41 1
sum	14	13	13	6
weight	3	5	5	3
weighted	42	65	65	18

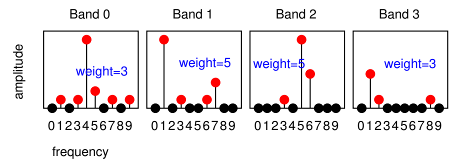
Approach

$$\sum_{bands}\sum_{f \in freqs} Bits(f)$$

- Start with nothing
- Start with bitbudget
- While(bitbudget>0)
  - Identify Largest Error component: Error(freq)×W(freq)
  - What do to reduce error?

$$\sum_{f \in freqs} Error(f)\times W(f)$$

Bits = 0
Error = 190

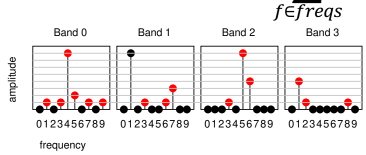


Approach

$$\sum_{bands}\sum_{f \in freqs} Bits(f)$$

- Start with nothing
- Start with bitbudget

$$\sum_{f \in freqs} Error(f)\times W(f)$$



Bits = 3+6=9
Error = 125

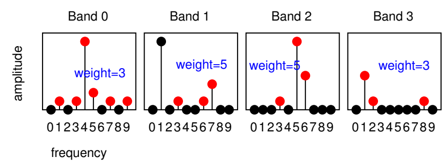
	11 8 2 11	0 0 0 0	1 8 4	41 1
sum	14		13	6
weight	3	5	5	3
weighted	42	0	65	18

Approach

$$\sum_{bands}\sum_{f \in freqs} Bits(f)$$

- Start with nothing
- Start with bitbudget
- While(bitbudget>0)
  - Identify Largest Error component: Error(freq)×W(freq)
  - What do to reduce error?

$$\sum_{f \in freqs} Error(f)\times W(f)$$

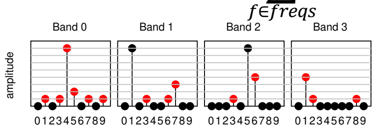


Approach

$$\sum_{bands}\sum_{f \in freqs} Bits(f)$$

- Start with nothing
- Start with bitbudget

$$\sum_{f \in freqs} Error(f)\times W(f)$$



Bits = 18
Error = 60

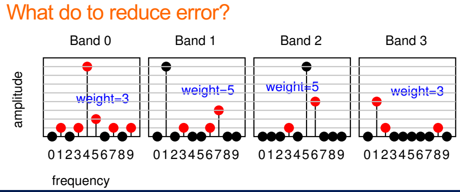
	11 8 2 11	0 0 0 0	0 00	41 1
sum	14			6
weight	3	5	5	3
weighted	42	0	0	18

Approach

$$\sum_{bands}\sum_{f \in freqs} Bits(f)$$

- Start with nothing
- Start with bitbudget
- While(bitbudget>0)
  - Identify Largest Error component: Error(freq)×W(freq)
  - What do to reduce error?

$$\sum_{f \in freqs} Error(f)\times W(f)$$

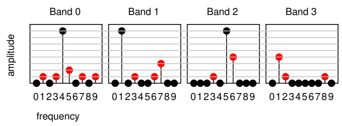


Approach

$$\sum_{bands} \sum_{f \in freqs} Bits(f)$$

- Start with nothing
- Start with bitbudget

$$\sum_{f \in freqs} Error(f) \times W(f)$$



Bits = 27
Error = 18

		0 0 0 0		0 0 0		41 1			
sum	0					6			
weight	3	5		5		3			
weighted	0	0		0		18			

Next Time

- Continue to refine into algorithm

Big Ideas

- Can use pyschoacoustics to compress audio
  - Eliminate portions of signal that human's don't notice
- Optimization
  - Identify Design Space (knobs)
  - Identify Costs and Constraints
  - Formulate quantitatively

Coming Up

- Feedback
- Lab 5 writeup today
- Lab 6 start on Monday
  - On syllabus
  - Start of 2 week lab
- Midterm on Wednesday
  - Rules/policies on syllabus now

References

Tutorials on Psychoacoustic Coding (in increasing order of abstraction and generality)

D. Pan, M. Inc, and I. L. Schaumburg. A tutorial on MPEG/audio compression. IEEE multimedia, 2(2):60–74, 1995.

Nikil Jayant, James Johnston, and Robert Safranek. Signal compression based on models of human perception. Proceedings of the IEEE, 81(10):1385–1422, 1993.

V. K. Goyal. Theoretical foundations of transform coding. IEEE Signal Processing Magazine, 18(5):9–21, 2001.

Lightweight Overview of MP3

Rassol Raissi. The theory behind mp3. Technical report, MP3' Tech, December 2002.

Scientific Basis of MP3 Coding Standard

J. D. Johnston. Transform coding of audio signals using perceptual noise criteria. IEEE Journal on selected areas in communications, 6(2):314–323, 1988.