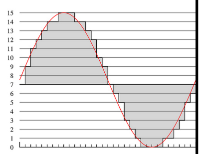
Penn Engineering

ESE



Lecture #6 – Psychoacoustic Model/Compression/MP3

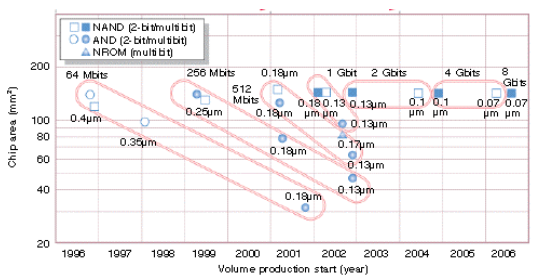
ESE 150 – Digital Audio Basics

Based on slides © 2009—2020 DeHon, Koditschek
Additional Material © 2014 Farmer

First MP3 Player

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

Flash Scaling



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

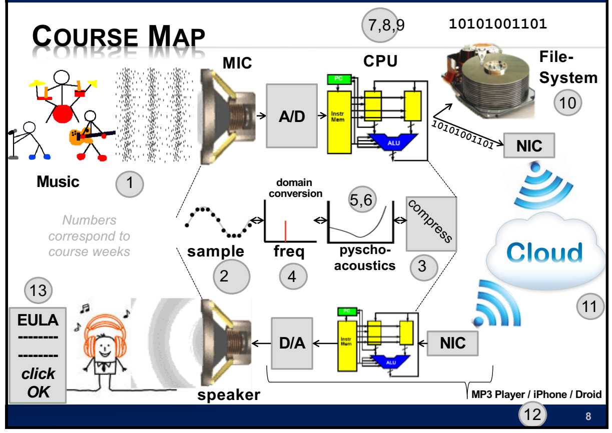
First MP3 Player

- 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

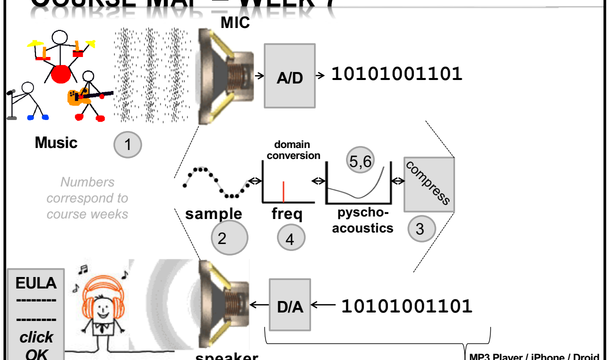
Lecture Topics

- 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 Lab
- References

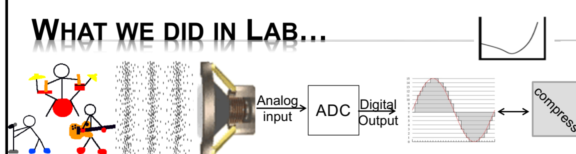
Course Map



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

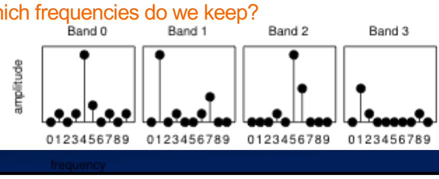
- 4 critical bands
- 10 frequencies
- 16b amplitude
- 107b encoding budget
- Preclass 2: amplitude quantization necessary to achieve budget?
- Preclass 3: frequencies (reduced sampling rate) can keep to achieve budget?

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
- Other tricks?

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!

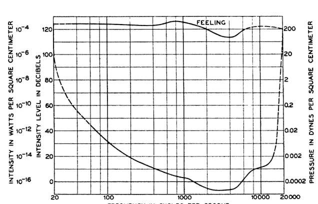
Optimizing Encoding

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

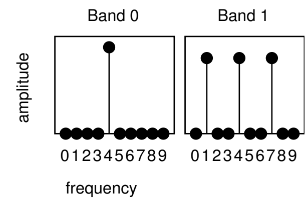
- **Amplitude quantization**
  - Per band
- **Frequency quantization**
  - Per band?
- **Frequencies kept (per critical band)**
  - Per band
- **…and can perform lossless compression**

Opportunities/Challenges

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

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(freq) = |OrigFreq Amplitude – Encoded|**
  - Whole OrigFreq if dropped
  - |OrigFreq-Quantize(OrigFreq,bits)| if quantized
- **W(freq)**
  - 0 if below hearing threshold
  - 0 if masked
  - Value between 0 and 1 if partially masked in critical band
  - Really depend on what already encoded

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

Optimization Problem

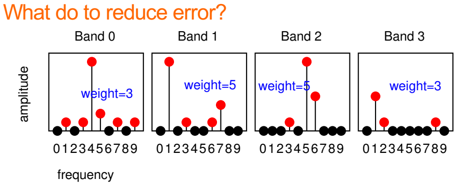
- **How fit in the resource constraints (128Kb/s) while maximizing goodness (sound quality)?**
- **Quantify bits used** $\sum_{bands} \sum_{f \in freqs} Bits(f)$
- **Quantify goodness**

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

Approach

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

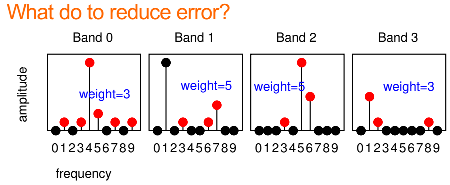
- **Start with nothing**
- **Start with bitbudget** $\sum_{f \in freqs} Error(f) \times W(f)$
- **While(bitbudget>0)**
  - Identify Largest Error component: Error(freq)×W(freq)
  - What do to reduce error?



Approach

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

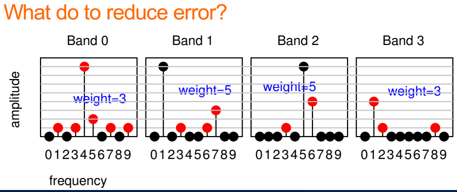
- **Start with nothing**
- **Start with bitbudget** $\sum_{f \in freqs} Error(f) \times W(f)$
- **While(bitbudget>0)**
  - Identify Largest Error component: Error(freq)×W(freq)
  - What do to reduce error?



Approach

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

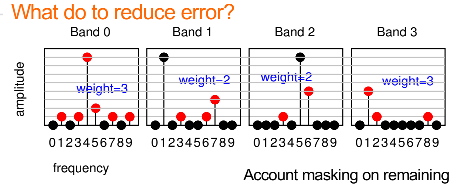
- **Start with nothing**
- **Start with bitbudget** $\sum_{f \in freqs} Error(f) \times W(f)$
- **While(bitbudget>0)**
  - Identify Largest Error component: Error(freq)×W(freq)
  - What do to reduce error?



Approach

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

- **Start with nothing**
- **Start with bitbudget** $\sum_{f \in freqs} Error(f) \times W(f)$
- **While(bitbudget>0)**
  - Identify Largest Error component: Error(freq)×W(freq)
  - What do to reduce error?



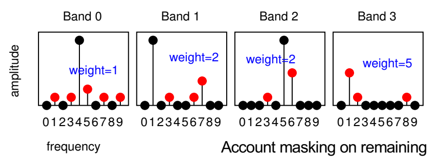
Account masking on remaining freqs.

Approach

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

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

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



Account masking on remaining freqs.

Also Quantization

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

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

- To keep simple, assumed fixed quant.
- Incrementally assign bits
- While(bitbudget>0)
  - Identify Largest Error component: Error(freq)×W(freq)
  - Assign more bits to that frequency
  - (return to example at end if time permits)

Approach (Greedy)

- Start with nothing
- Start with bitbudget
- While(bitbudget>0)
  - Identify Largest Error component
  - Allocate some bits to reduce error
    - Add frequency
    - Add quantization bits to band
    - Pick one to most reduce the error

Approach (Refine Greedy)

- Start with nothing
- Start with bitbudget
- While(bitbudget>0)
  - Identify Largest Δerror/Δbits
- Why might prefer?
- Large error, might take many bits to improve
  - Maybe could be better spending those bits to fix many problems?

Adaptive Refinement

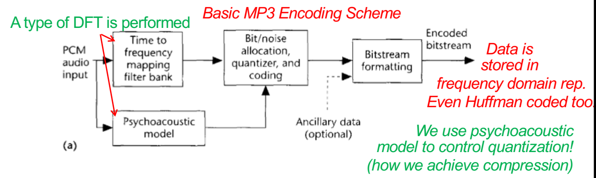
- Rediscovering where to allocate everything every time may be laborious
- Maybe we can get close and adjust?

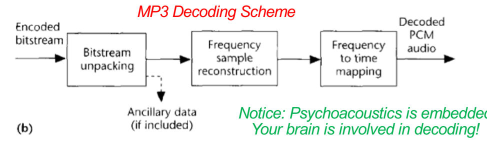
Approach (Adaptive)

- Start with budget guess
  - Quantization in bands
  - Frequencies to keep in each band
- Encode, compress
- What do if haven't used up all bits?
- What do if over budget?

PERCEPTUAL CODING & MP3

MP3 Flow Chart (Encoding/Decoding)

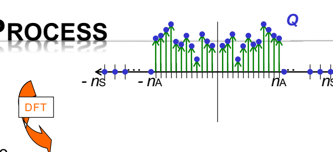




MP3 Encoding Process

- **All MP3 files broken into "Frames"**
  - Each frame stores 1152 Audio Samples
  - Lasts for 26 ms
  - Frame also divided further into 2 "granuels"
    - Each granuel contains 576 samples

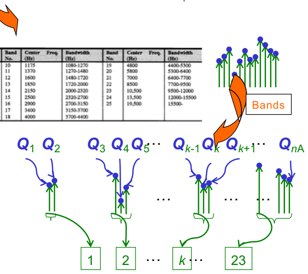
MP3 Encoding Process



acquire & transform "frame"

- Bring in 1152 PCM samples
- Use DFT/MDCT to transform to frequency-domain

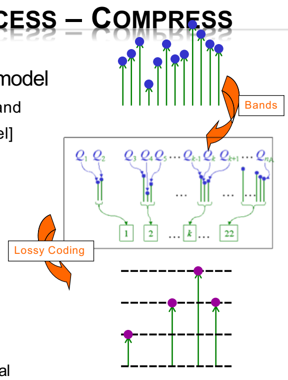
Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)
1	50	-100	10	1175	1080-1270	19	4800	4400-5300
2	150	100-200	11	1370	1270-1480	20	5800	5300-6400
3	250	200-300	12	1600	1480-1720	21	7000	6400-7700
4	350	300-400	13	1850	1720-2000	22	8500	7700-9500
5	450	400-510	14	2150	2000-2320	23	10,500	9500-12000
6	570	510-630	15	2500	2320-2700	24	13,500	12000-15500
7	700	630-770	16	2900	2700-3150	25	19,500	15500-
8	840	770-920	17	3400	3150-3700			
9	1000	920-1080	18	4000	3700-4400			



assign frequencies to bands

- use psychoacoustic model lookup to determine frequency bandwidth of each critical band
- separate samples into bands

MP3 Encoding Process – Compress



Compress based off masking model

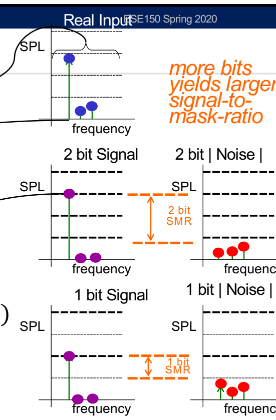
- Goal: minimize bits per critical band
  - $C_{band\ k(j)}$ = Round[$Q_{band\ k(j)}$, Level]
- by appeal to "masking" models
  - $Q_{band\ k} = C_{band\ k} + D_{band\ k}$
- where distortion ("perceptual noise")
  - $D_{band\ k}$
- between retained signal
  - $C_{band\ k}$
- and actual signal
  - $Q_{band\ k}$
- should be "masked" by retained signal

Compress – Up close

- E.g., Look at k[th] band
  - $Q_{band\ k} = (Q_{band\ k(1)}, \ldots, Q_{band\ k(m)})$
- Determine masking paradigm:
  - tone-masking-noise
  - noise-masking-tone
  - noise-masking-noise
- E.g., for tone masker,
  - Pick tone frequency, band k (j)
  - at maximal amplitude in the band
- Choose quantization level and compute error and bits.

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

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

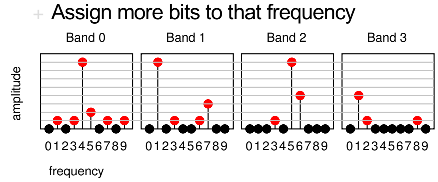


Extend Example

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

- Incrementally assign bits
- While(bitbudget>0)
  - Identify Largest Error component: Error(freq)×W(freq)
  - Assign more bits to that frequency

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



Weights
Unmasked 3 5 5 3
Masked 1 2 2 1

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
  - Algorithms to approach
  - Iterative/adaptive approach
    - Deal with effects that aren't completely predictable

Learn More

- Optimization –
  - continuous mathematical optimization ESE204, ESE504, ESE605
  - discrete optimization CIS121, CIS320
- Signal processing – ESE224

Coming Up

- Monday start of 2 week lab:
  - Perform perceptual compression
    - Convergence of everything seen first 6 weeks
  - Formal lab report on this 2 week lab
  - No weekly lab report next Friday
  - 2nd half Monday after Spring Break
- Wednesday: midterm (class time)
- Office Hours:
  - Cancel Thursday office hours after midterm
    - Thursday before Spring Break
  - (this Thursday office hours as usual)

Midterm – In Class

- Closed book, no notes
- Calculators allowed
- 4:35pm-5:50pm
- 5% of grade
  - prepare for final
- Last 2 year's midterm and answers
  - on 2018, 2019 syllabus

Topics

- Data representation in bits
- Sounds waves
- Sampling
- Quantization
- Nyquist
- Lossy/lossless compression
- Common case
- Frequency domain
- Psychoacoustics
- Perceptual coding

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.