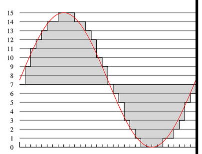
Penn Engineering



Lecture #12 – Psychoacoustic Compression Algorithm

ESE 150 – Digital Audio Basics

Based on slides © 2009—2021 DeHon, Koditschek
Additional Material © 2014 Farmer

Lecture Topics

- Lab vs. Lecture
- Big Picture reminder
- Review and preclass
- How do we take advantage of psychoacoustics in MP3 Achieve this 6—12x reduction from CD Audio
  - Review Tricks
  - Formulate Optimization
  - Algorithm for Adaptation
- References

Compare to Lab

- Lab 6
  - Capture spirit of reducing frequencies
  - Simplified – only taking loudest fraction in each band
    - Rather than being rigorous about masking
    - Or trying to hit some fixed rate
- Lecture 11 & 12
  - Help understand more what real MP3 encoding looks like
  - Thinking about fixed rate
    - And adaptation for variable rate encoding from Huffman
  - Formulating masking more explicitly
    - But still simplistic
  - Illustrating Optimization Approaches

Big Picture: Why Care

- MP3 shows
  - We can reduce bits needed too accurately represent music by 12x over PCM samples
- 1998
  - Necessary to make MP3 player viable at all on 32MB Flash memory
    - Small, affordable by masses
  - Also meant lower bandwidth to stream
- 2006
  - Necessary to make Spotify streaming viable
- 2021
  - With 64GB phones, maybe not essential for music on iPhone, but…

Big Picture: Why Care

- Networks / Internet bandwidth / Cell capacity
  - Allows us to reduce bandwidth needed for audio
  - Alternative is
    - Building out more wires and capacity
    - Fewer cell calls supported simultaneously in a region
  - Being efficient with bits
    - Reduces physical investment needed in wires
- Same ideas enable video
  - Viable to store (many) hours of video on phone/tablet
    - Of increasing resolution: HD, 4K….
  - Viable to stream video into homes
  - Zoom, Netflix, AppleTV, … all enabled by compression

Review and Preclass Setup/Reminder

Knobs we can turn

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

Preclass 1

- **44,300 samples/s**
- **16b**
- **26ms window**
- **a) How many bits?**
- 128Kb/s stereo → 64Kb/s per audio channel
- **b) How many bits per 26ms window?**
- **c) ratio?**

Optimization Problem

- **How fit in the resource constraints (128Kb/s) while maximizing goodness (sound quality)?**
- **Quantify bits used:** $\sum_{bands} \sum_{f \in freqs} Bits(f)$
  - Cannot exceed 128Kb/s
  - = 1,704 b / 26ms frame / channel
- **Quantify goodness: minimize** $\sum_{f \in freqs} Error(f) \times W(f)$

Preclass 2

- 576 frequencies
- 16b amplitude
- 1704b budget
- (frequency,amplitude) pairs to represent
- **How many frequencies can we keep?**
- **Conclude:** cannot keep all frequencies and hit budget

Preclass 3

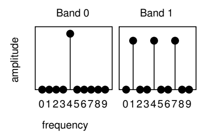
- **Bits if only have 3 non-masked, non-zero frequencies?**
- **Conclude:** some frames won't use all their bits

Preclass 4 and 5

- 60dB, 42dB, 30dB each are 25% of amplitudes
- Other magnitudes remaining 25%
- How many bits to represent 80 non-zero/non-masked frequencies?
- If all frequencies equally likely, how many bits to represent 80 non-zero/non-masked frequencies?
- **Conclude:** number of frequencies can keep depends on compressibility

Seen

- **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:** $\sum_{bands} \sum_{f \in freqs} Bits(f)$
  - Cannot exceed 128Kb/s
  - = 1,704 b / 26ms frame / channel
- **Quantify goodness: minimize** $\sum_{f \in freqs} Error(f) \times W(f)$

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
  - Really depend on what already encoded

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

Example Weight Function W(f)

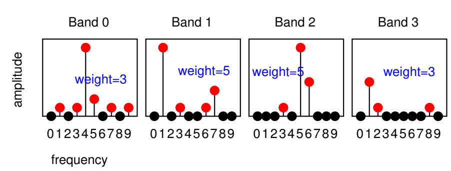
- **W(f)=CBWeight*Mask**
- **Mask = 0 if MaxAmp-FreqAmp<4**
- ~~(MaxAmp-FreqAmp)/MaxAmap otherwise~~
  - 1 otherwise
  - For example that follows, don't use second part

Approach

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

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

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

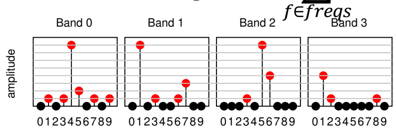


APPROACH

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

- Start with nothing
- Start with bitbudget

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



11 8 2 11 8 1 1 3 1 8 4 41 1

	Band 0	Band 1	Band 2	Band 3
sum	14	13	13	6
weight	3	5	5	3
weighted	42	65	65	18

Bits = 0
Error = 190

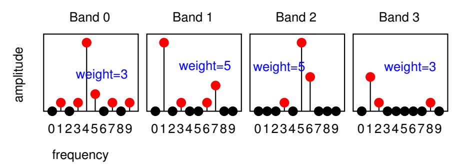
APPROACH

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

- Start with nothing
- Start with bitbudget
- While(bitbudget>0)
  - Identify Largest Error component: Error(freq)×W(freq)

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

Bits = 0
Error = 190

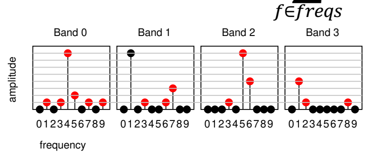


APPROACH

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

- Start with nothing
- Start with bitbudget

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



11 8 2 11 0 0 0 0 1 8 4 41 1

	Band 0	Band 1	Band 2	Band 3
sum	14		13	6
weight	3	5	5	3
weighted	42	0	65	18

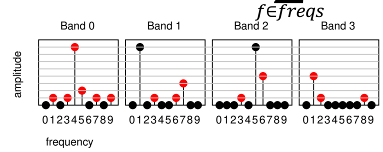
Bits = 3+6=9
Error = 125

APPROACH

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

- Start with nothing
- Start with bitbudget

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



11 8 2 11 0 0 0 0 0 0 0 41 1

	Band 0	Band 1	Band 2	Band 3
sum	14			6
weight	3	5	5	3
weighted	42	0	0	18

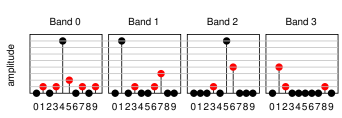
Bits = 18
Error = 60

APPROACH

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

- Start with nothing
- Start with bitbudget

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



0 0 0 0 0 0 0 41 1

	Band 0	Band 1	Band 2	Band 3
sum	0			6
weight	3	5	5	3
weighted	0	0	0	18

Bits = 27
Error = 18

ALSO QUANTIZATION

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

- 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
    - Go from 0 bits to 1 bit
      - 1 bit to 2 bits

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

Approach (Greedy)

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

Adaptive Refinement

- **Rediscovering where to allocate everything every time may be laborious**
  - Often same frequencies persist for more than 26ms
- **Maybe we can get close and adjust?**
  - Use critical band allocation from previous frames as a starting point guess
    - bits, frequencies, quantization
  - Try initial encoding with that

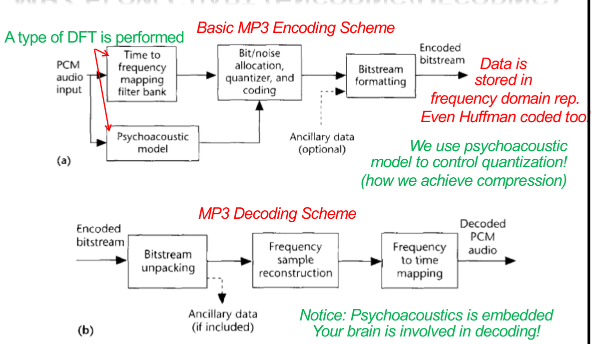
Approach (Adaptive)

- **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?**

Roundup

Perceptual Coding & MP3

MP3 Flow Chart (Encoding/Decoding)



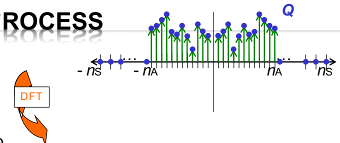
MP3 Encoding Process

- **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

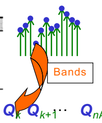
MP3 Encoding Process

acquire & transform "frame"

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

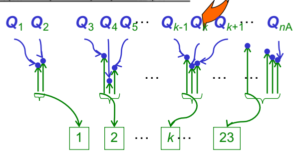


Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)	Band No.	Center Freq. (Hz)	Bandwidth (Hz)
1	50	-100	10	1175	1080-1270	19	4800	4400-5300
2	150	100-200	11	1370	1270-1480	20	5800	5300-6400
3	250	200-300	12	1600	1480-1720	21	7000	6400-7700
4	350	300-400	13	1850	1720-2000	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



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

- **Feedback (including Lab 6 part 1)**
- **No lab due today**
- **Monday: Lab 6 continued**
- **Wednesday: Spring Break?!? – no lecture**
  - No office hours
  - Thursday: no office hours
- **Next Friday: back for Lecture**
- **Saturday 2-3pm and 6-7pm office hours**
- **Sunday (3/14) formal Lab Due**

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.