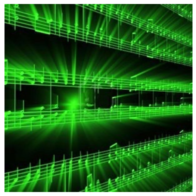
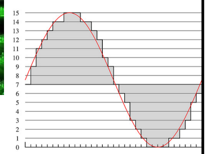
ESE150 Spring 2023

Lecture #11 – Psychoacoustic Model/Compression/MP3

ESE 1500 – Digital Audio Basics

Based on slides © 2009—2023 DeHon, Koditschek
Additional Material © 2014 Farmer

1

ESE150 Spring 2023

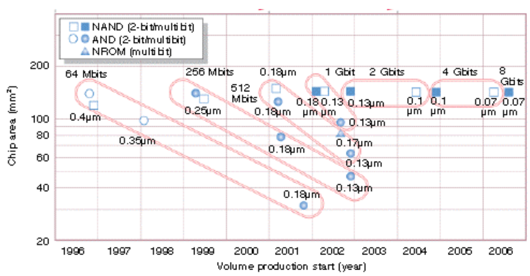
First MP3 Player

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

2

ESE150 Spring 2023

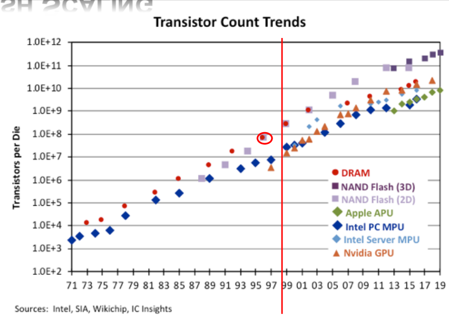
Flash Scaling



3

ESE150 Spring 2023

Flash Scaling

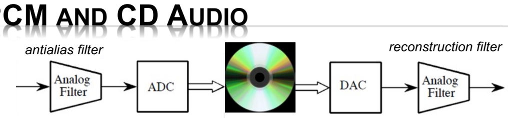


https://www.icinsights.com/news/bulletins/Transistor-Count-Trends-Continue-To-Track-With-Moores-Law/

4

ESE150 Spring 2023

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!)

5

ESE150 Spring 2023

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

6

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

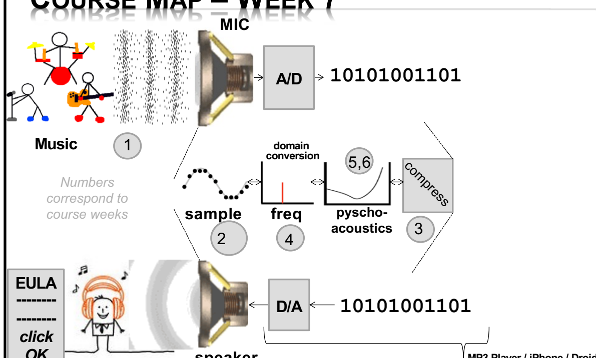
7

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 (Monday 3/13)
- Next Lab
- References

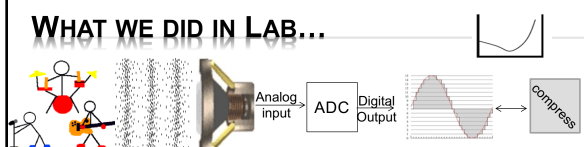
8

Course Map – Week 7



9

What we did in Lab…



- Week 1: Converted analog sound to Digital
  - Sample: Break up independent variable, take discrete 'samples'
  - Quantize: Break up dependent variable into n-levels (need 2^n bits to digitize)
- Week 2: Reconstructed analog sound from Digital
- 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

10

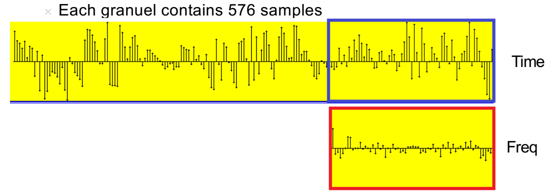
Compare to Lab

- 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 (this week and after spring break)
  - 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

11

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



12

PRECLASS

13

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?**

14

PRECLASS 2

- 16b amplitude
- 1704b budget
- Quantize frequency
- **How many quantized frequencies?**
- **Conclude:** cannot keep all frequencies and hit budget

15

PRECLASS 3

- 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

16

PRECLASS 4

- 576 frequencies
- 1704b budget
- (frequency,amplitude) pairs to represent
- 24 human critical bands
- **How many bits per band? (evenly divided)**
- **Quantization required to keep 5 frequencies per band?**
- **Conclude:** can tradeoff frequencies and amplitude quantization

17

TRICKS FOR COMPRESSION

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

18

BACKGROUND

19

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)*

20

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!

21

BIG PICTURE: WHY CARE

- **MP3 shows**
  - We can reduce bits needed to adequately 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
- **2023**
  - With 256GB phones, maybe not essential for music on iPhone, but…

26

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, Disney+, AppleTV, … all enabled by compression

27

MIDTERM – WEDNESDAY IN LECTURE

- **Closed book, notes**
- **Calculators allowed**
- **50 minutes**
  - Same as last year
  - Shorter than previous years
- **5% of grade**
  - prepare for final
- **Last 5 year's midterm and answers**
  - On 2022 syllabus
    - like this year
    - (50 minutes, closed book)
  - on 2018, 2019, 2020 syllabus
    - Were all in-person, closed book (75 minutes)
  - 2021 was online, open book, also on syllabus
    - (120 minutes)

Topics

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

28

Penn Engineering ESE

Part 2

Optimizing Encoding

29

Goal

- **Fit in bit budget**
- **Maximize quality of sound**

30

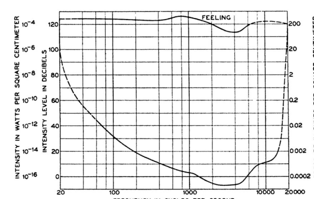
Knobs we can turn

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

31

Bands vary in importance

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



32

Knobs we can turn

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

33

Quantifying Bit cost

- **Fixed frequencies per Band:**
  - Bands × (Frequencies/Band) × Bits/freq

34

Preclass 5

- **Bits if only have 1 non-masked, non-zero frequency in a critical band?**
- **Conclude:** given fixed band allocation, some bands won't use all their bits

Opportunities/Challenges

- **Trying to hit fixed bit rate, what challenge does lossless compression impose?**
  - Encounter many common frequencies, amplitudes?
  - Encounter many uncommon frequencies, amplitudes?
  - Hint: what did we see in Lab 3 for time samples?

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)?**
  - 128K/s * 0.026 s = 3408 bits per 26ms frame
  - 3408/2 = 1704b per stereo channel

Optimization Problem

- **How fit in the resource constraints (128Kb/s) while maximizing goodness (sound quality)?**
  - 128K/s * 0.026 s = 3408 bits per 26ms frame
  - 3408/2 = 1704b per stereo channel
- **Optimization problems central to engineering**

Optimization Problem

- **How fit in the resource constraints (128Kb/s = 3408 bits/frame) while maximizing goodness (sound quality)?**
- **Quantify bits used**

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

- **Quantify quality**

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

Quantifying Bit cost

- **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**

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

41

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

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

42

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**

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

43

Problem Formulation

- **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**

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

44

Formulation vs. Solution

- **Formulation – goal – formally what want to achieve**
  - Today's focus
- **Solution – how go about solving it**
  - How achieve (minimize) constraints (costs) in formulation
  - More lecture 12 (Monday 3/13)

45

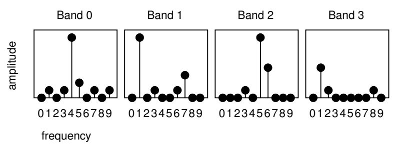
(as time permits)

Approach

46

Example

- Simplified example with 4 critical bands



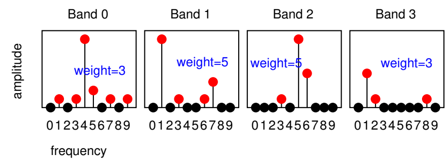
47

Approach

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

- Start with nothing
- Start with bitbudget
- Incremental moves to improve

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



48

Moves

- Select a frequency to encode
  - Main example today
- Assign more bits to a band for quantization

49

Example Weight Function W(f)

- W(f)=CBWeight*Mask
- Mask = 0 if MaxAmp-FreqAmp>3
  - 1 otherwise

50

Greedy Incremental Frequency Selection

- Start with nothing encoded
- Start with bits=0
- While (bits<bitbudget)
  - Identify frequency with maximum current error
  - Identify bits required to encode frequency (encode_bits)
  - If (bits+encode_bits<bitbudget)
    - Add frequency to encoding
    - Update bits (bits+=encode_bits)
  - Else break

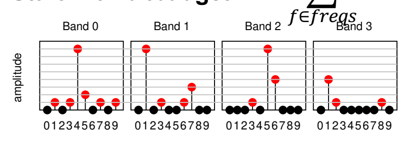
51

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

	Band 0	Band 1	Band 2	Band 3
	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

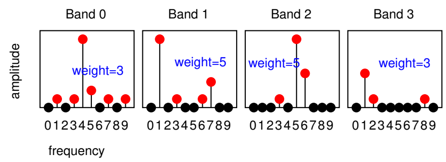
52

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?



Bits = 0
Error = 190

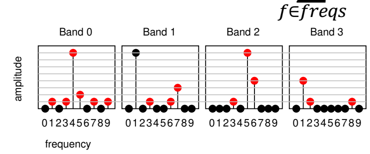
53

Approach

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

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

- Start with nothing
- Start with bitbudget



Assume 8 levels

Bits = 3+6=9
Error = 125

	Band 0	Band 1	Band 2	Band 3
	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

54

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?

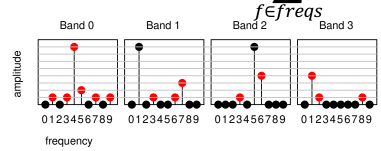
55

Approach

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

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

- Start with nothing
- Start with bitbudget



Bits = 18
Error = 60

	Band 0	Band 1	Band 2	Band 3
	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

56

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?

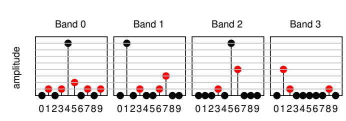
57

Approach

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

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

- Start with nothing
- Start with bitbudget



Bits = 27
Error = 18

	Band 0	Band 1	Band 2	Band 3
		0 0 0 0	0 00	41 1
sum	0			6
weight	3	5	5	3
weighted	0	0	0	18

58

Next Time (Monday 3/13)

- **Continue to refine into algorithm / solution**

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 including Lab**
- **Lab 5 writeup today**
- **Lab 6 start on today**
  - On syllabus
  - Start of 2 week lab
    - Simplified version compression algorithm
- **Midterm Wednesday**

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.