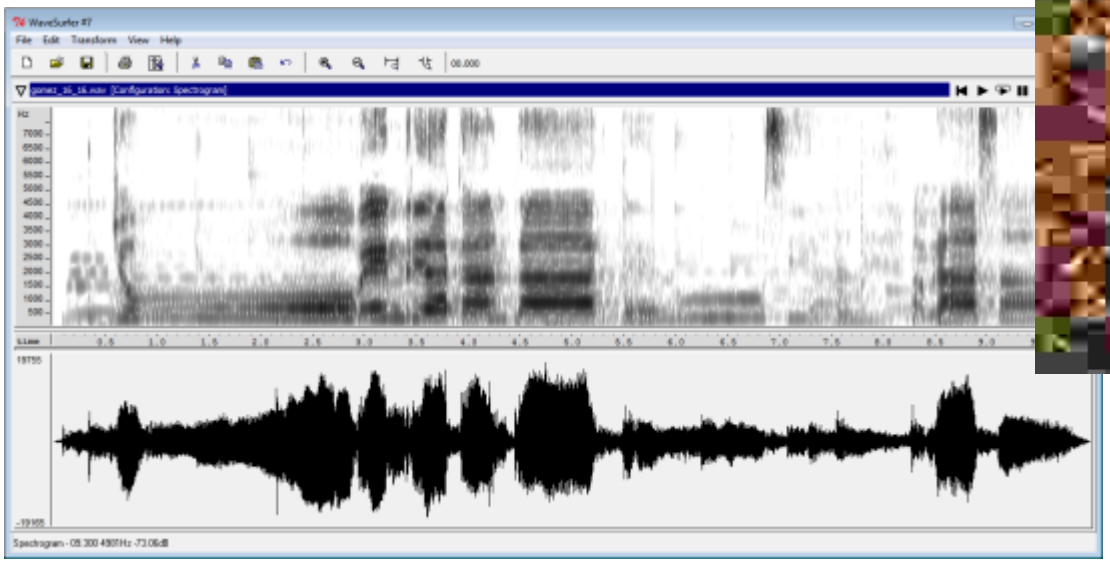
Lossy compression



$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

Overview

- Digital audio
  - Sampling rate
  - Quantization
  - MPEG audio layer 3 (MP3)
- JPEG still images
  - Color space conversion, downsampling
  - Discrete Cosine Transform (DCT)
  - Quantization
  - Encoding

http://www.vectronicsappleworld.com/collection/appleiie.html

Kate Bush 50 Words for Sno

https://www.hdtracks.com/index.php?file=catalogdetail&valbum_code=HD045778718603

HDtracks

High Definition Music / The World's Greatest-Sounding Music Downloads

Artists search

home | my account | logout

You have an order that is not completely downloaded. Click here to go to your account page to download your tracks.

Follow Us!

HDtracks newsletter

Tweet 585

Like 6.3k

EXPERIENCE A FREE HIGH RES DOWNLOAD

Member Login

You are logged in as:
kdv123@gmail.com

logout

Browse Music

The Classic 100

Artists

Genres

Artists | Kate Bush | 50 Words for Snow

Audiophile 96kHz/24bit

Title: 50 Words for Snow
Artist: Kate Bush
Genre: Pop
Label: Anti-/Epitaph
Release Date: 2011

50 Words For Snow is the latest release from British singer-songwriter Kate Bush, an artist who has been cited as a source of inspiration from the likes of Tori Amos and Bjork. With her multi-octave vocals and eloquent piano playing, Bush delivers another boundary defining album full of lyrical landscape. The winter themed album features seven new compositions and collaborations with guest musicians: Elton John, Andy Fairweather Low, Stephen Fry and Steve Gadd.

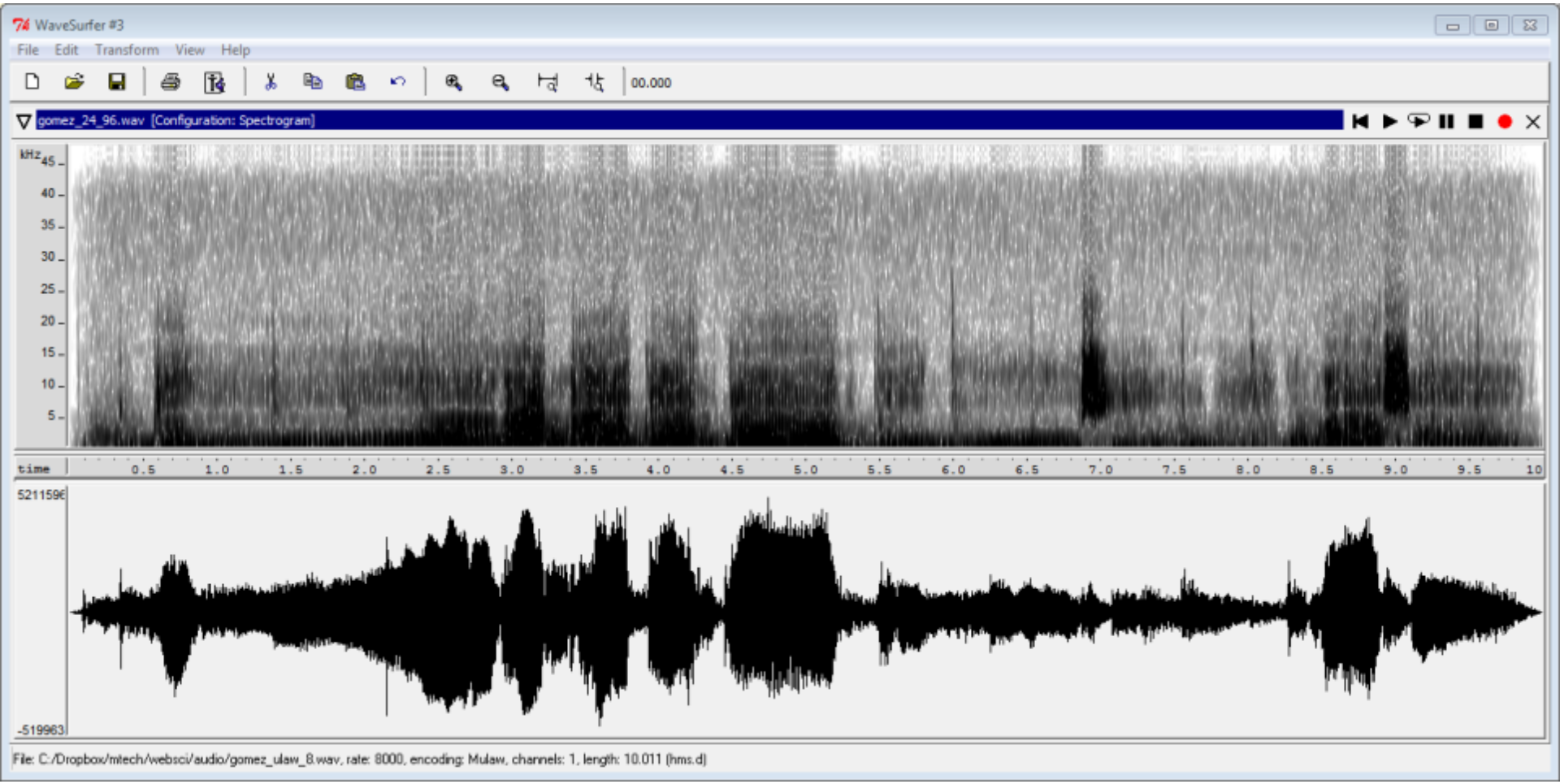
Which Format Should I Download?

Explore Album

Buy Album	Length	Format	Price	Add
50 Words for Snow	1:04:57	flac	$17.98	

Buy Individual Tracks

#	Track Title	Length	Listen	Format	Price	Add
1	Snowflake	9:46			Album only	
2	Lake Tahoe	11;08			Album only	
3	Misty	13:32			Album only	
4	Wild Man	7:16			Album only	
5	Snowed In At Wheeler Street	8:05			Album only	
6	50 Words for Snow	8:30			Album only	
7	Among Angels	6:48			Album only	

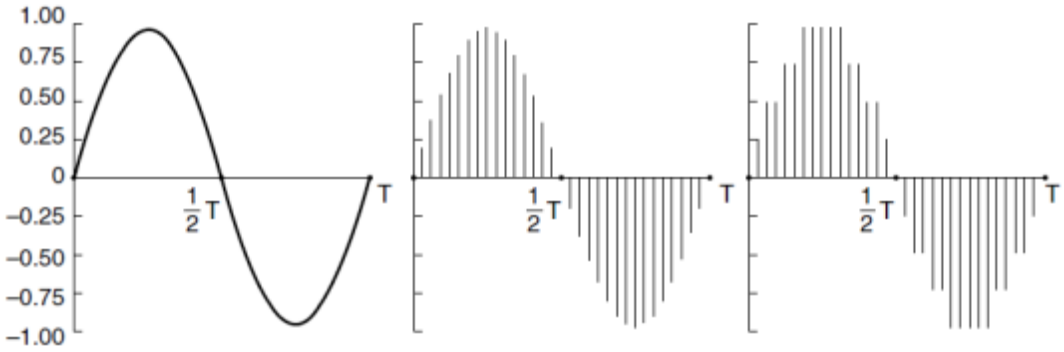


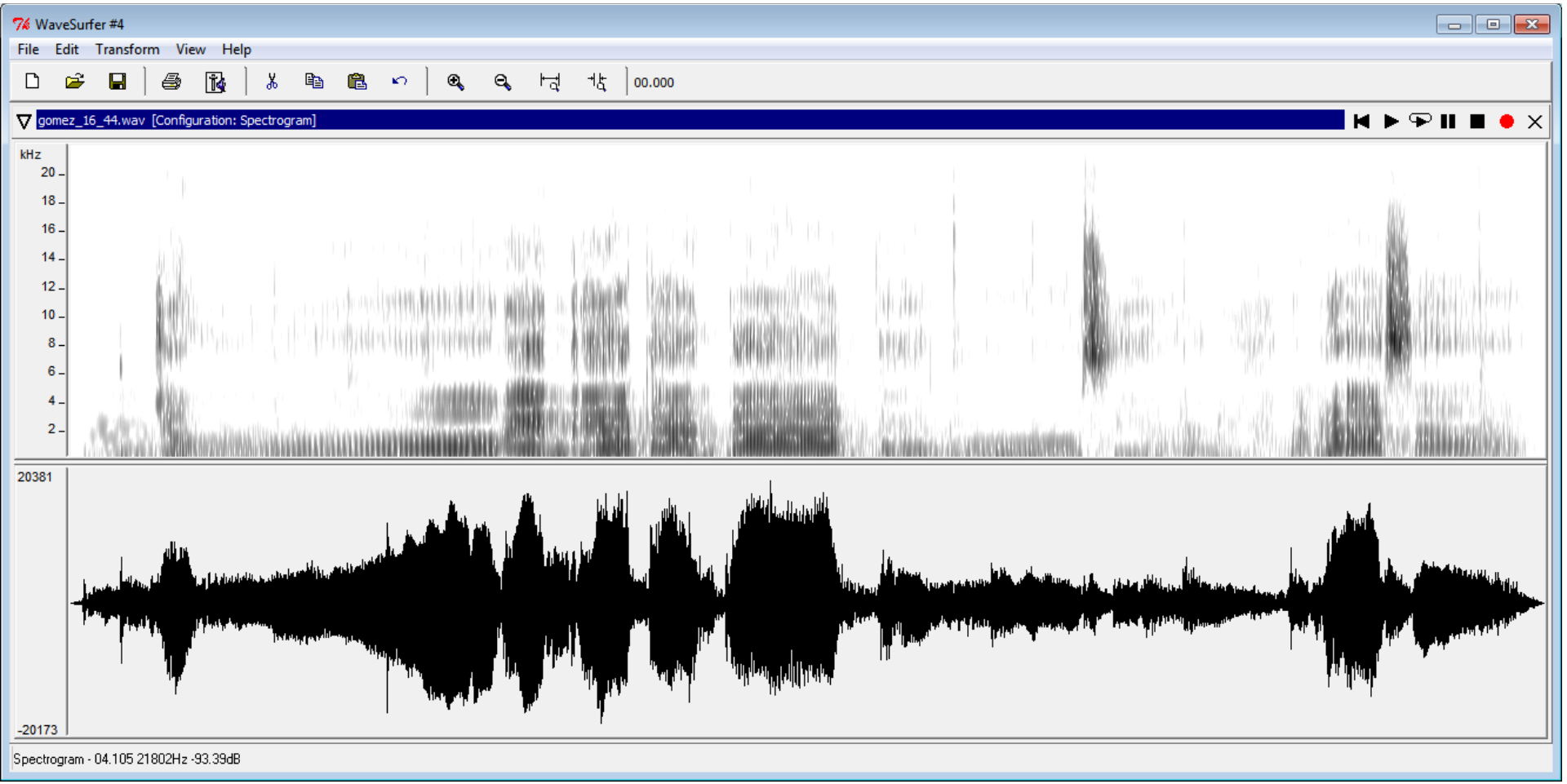
- Studio quality audio
  - 96kHz sampling
  - 24 bits / sample
  - 2 channels
- Stats
  - 10s = 3,411 KB
  - 2729 kbps
  - FLAC loseless, maximum

flac
free lossless audio codec

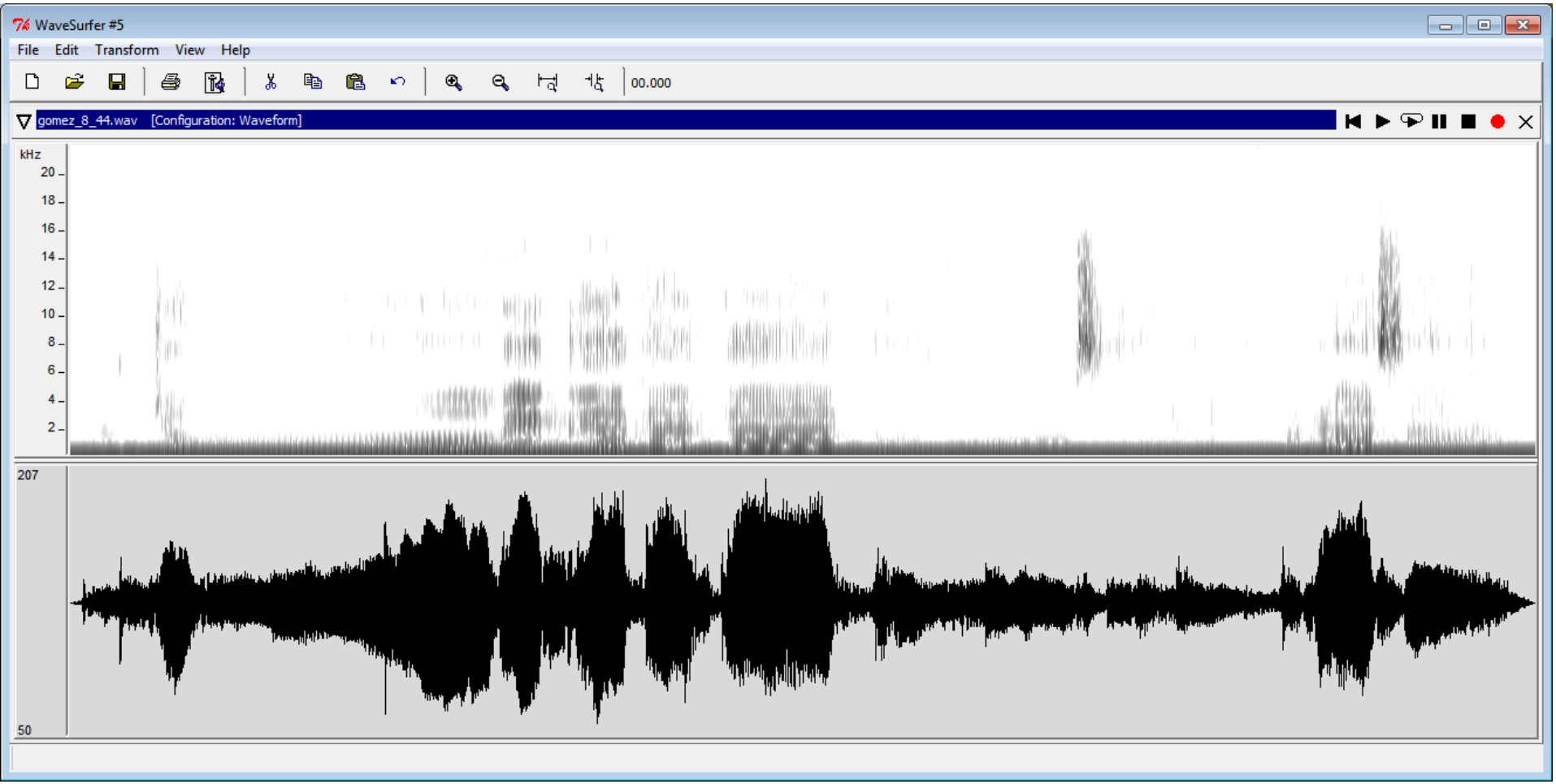
Digital audio

- Analog-to-Digital Converter (ADC)
  - Sampling rate of analog waveform
    - 96 kHz (studio)
    - 44.1 kHz (CD)
    - 8 kHz (telephone)
  - Bits per sample
    - 24 bits (studio)
    - 16 bits (CD)
    - 8 bits (telephone)
  - Number of channels
    - 1 = mono, 2 = stereo

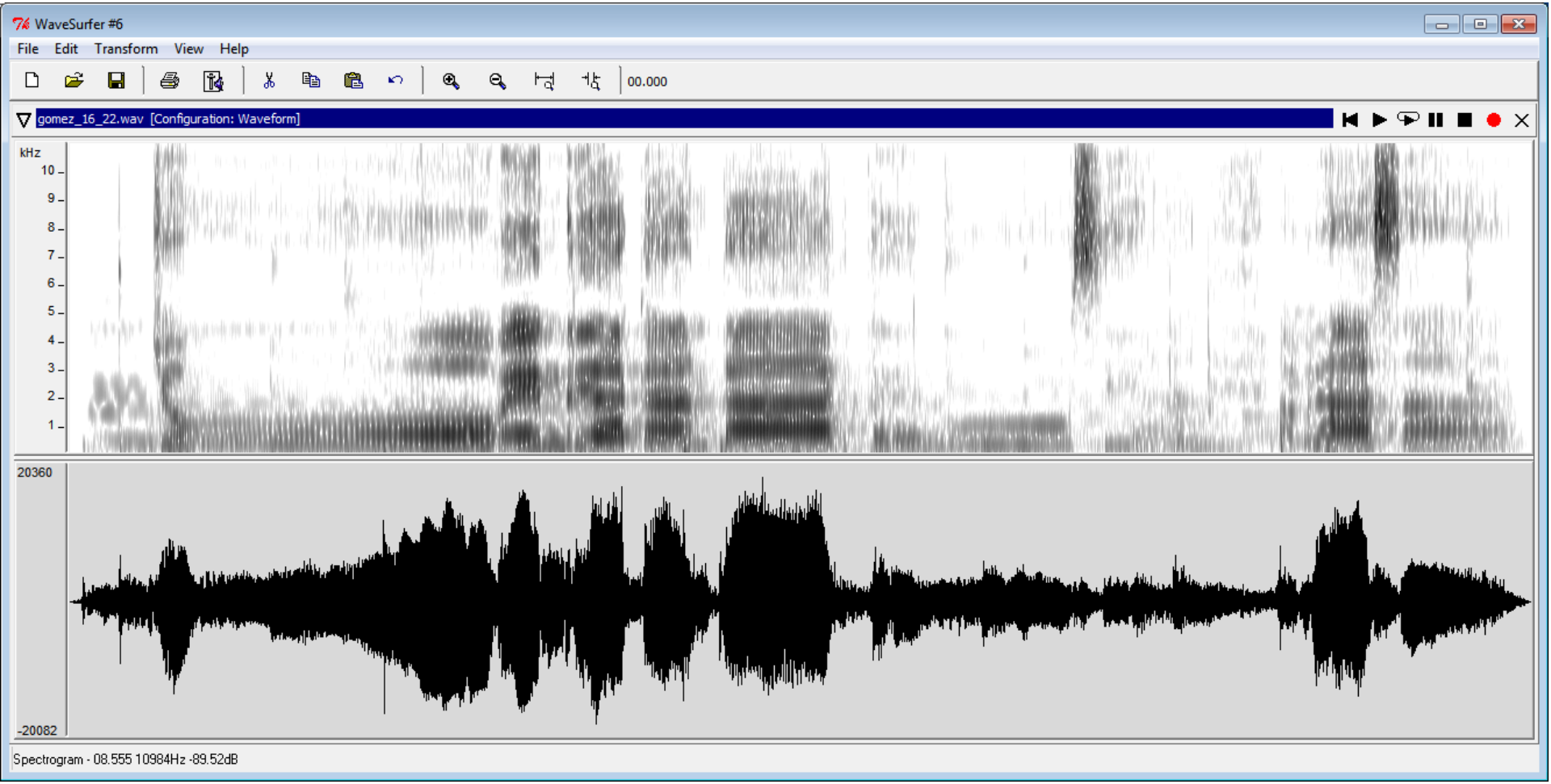




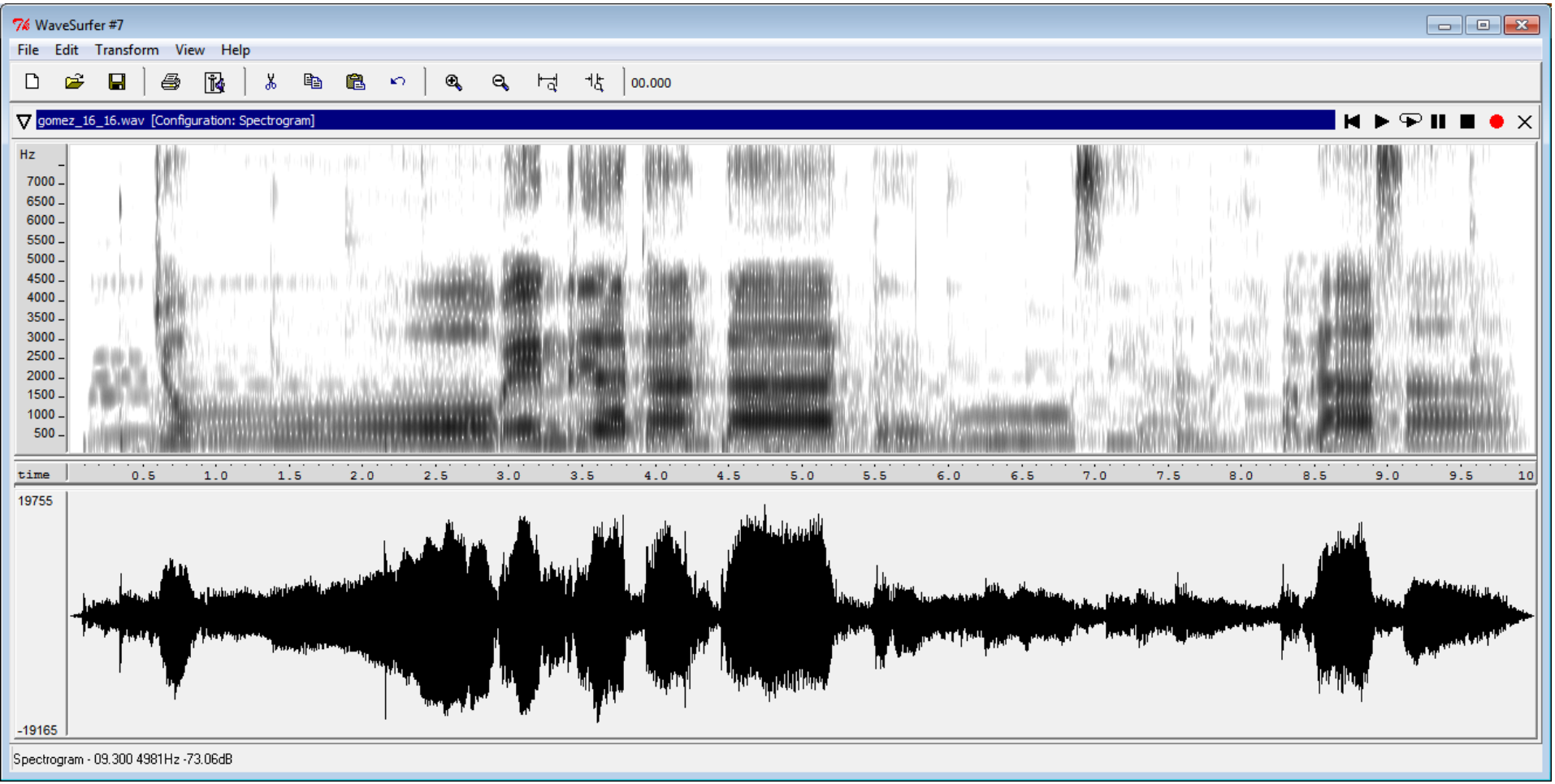
- CD quality audio
  - 44.1 kHz sampling
  - 16 bits / sample
  - 2 channels
- Stats
  - 10s = 831 KB
  - 665 kbps
  - FLAC loseless



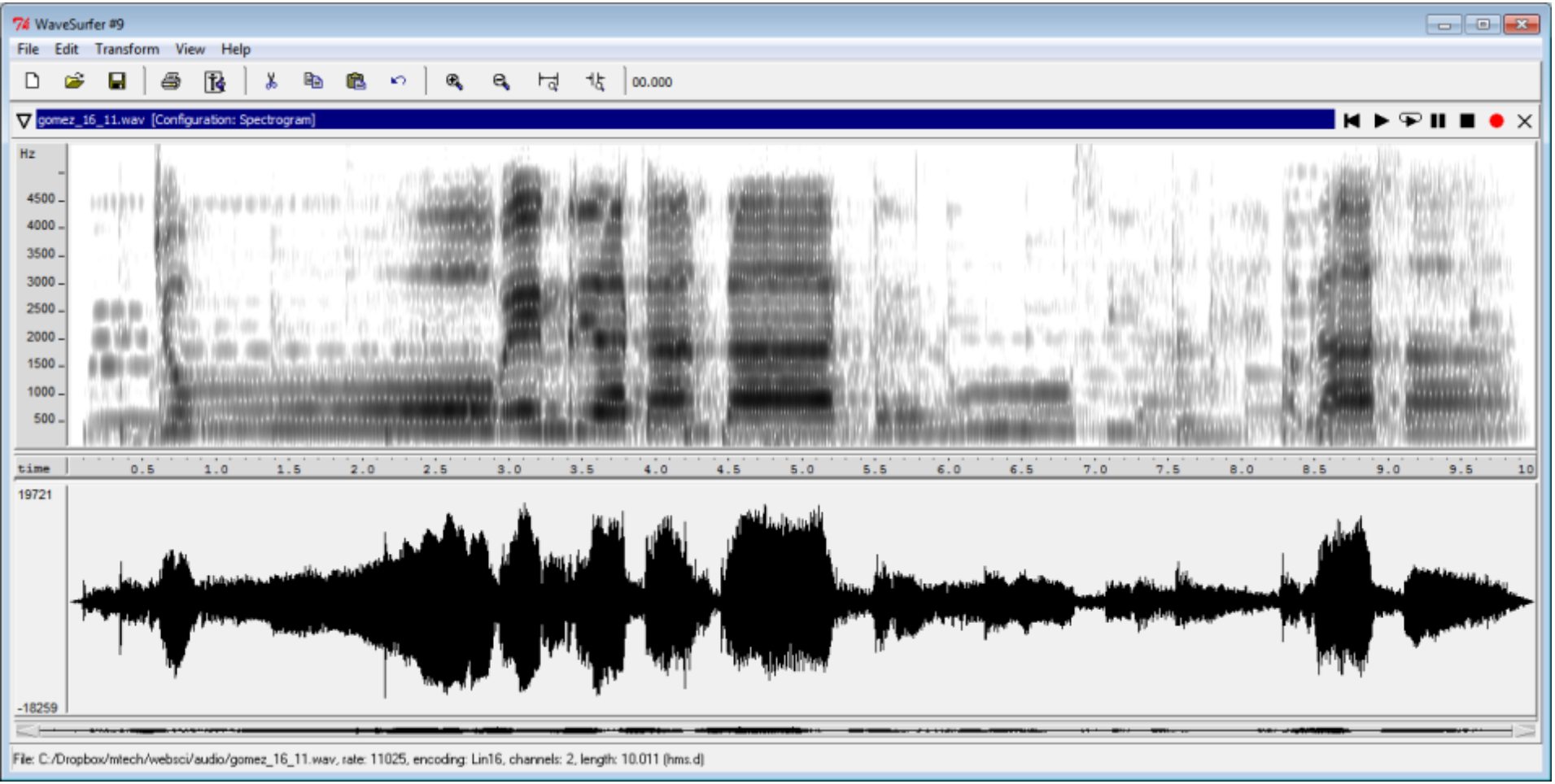
- 8-bit sampling rate
  - 44.1 kHz sampling
  - 8 bits / sample
  - 2 channels
- Stats
  - 10s = 211 KB
  - 169 kbps
  - FLAC loseless



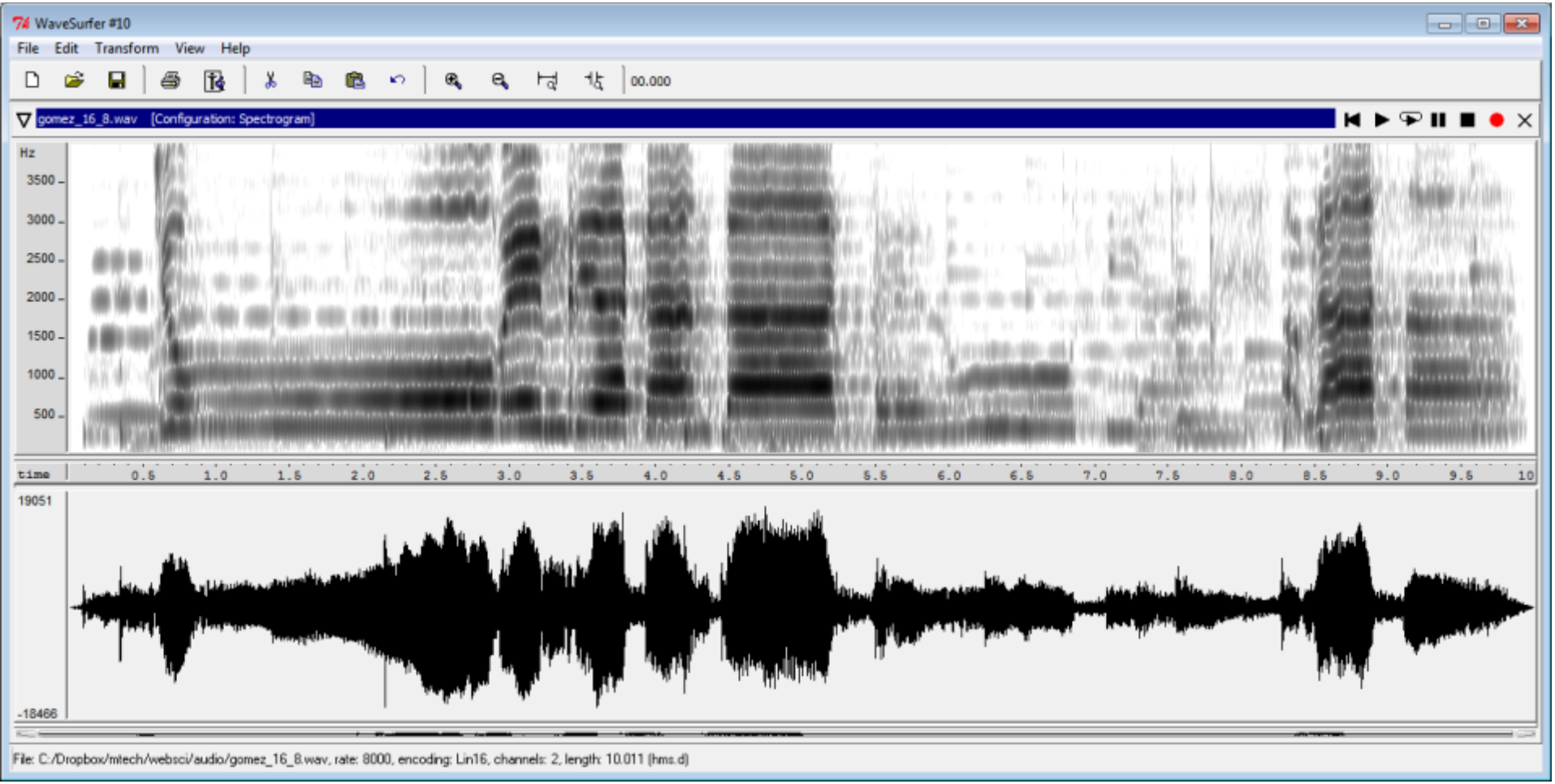
- 22 kHz sampling rate
  - 22.05 kHz sampling
  - 16 bits / sample
  - 2 channels
- Stats
  - 10s = 501 KB
  - 400 kbps
  - FLAC loseless



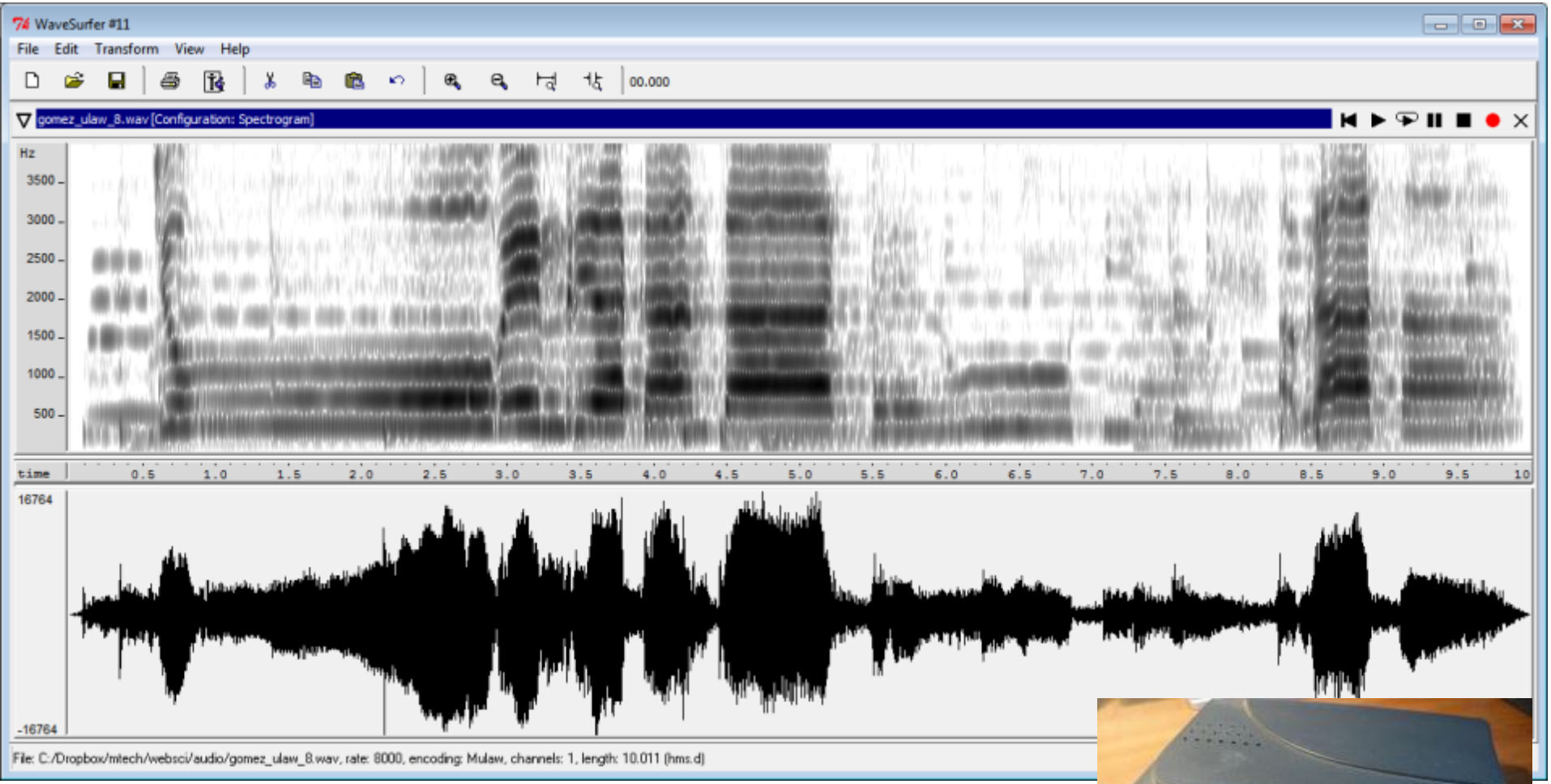
- 16 kHz sampling rate
  - 16 kHz sampling
  - 16 bits / sample
  - 2 channels
- Stats
  - 10s = 378 KB
  - 303 kbps
  - FLAC loseless



- 11 kHz sampling rate
  - 11 kHz sampling
  - 16 bits / sample
  - 2 channels
- Stats
  - 10s = 279 KB
  - 224 kbps
  - FLAC loseless



- 8 kHz sampling rate
  - 8 kHz sampling
  - 16 bits / sample
  - 2 channels
- Stats
  - 10s = 279 KB
  - 171 kbps
  - FLAC loseless



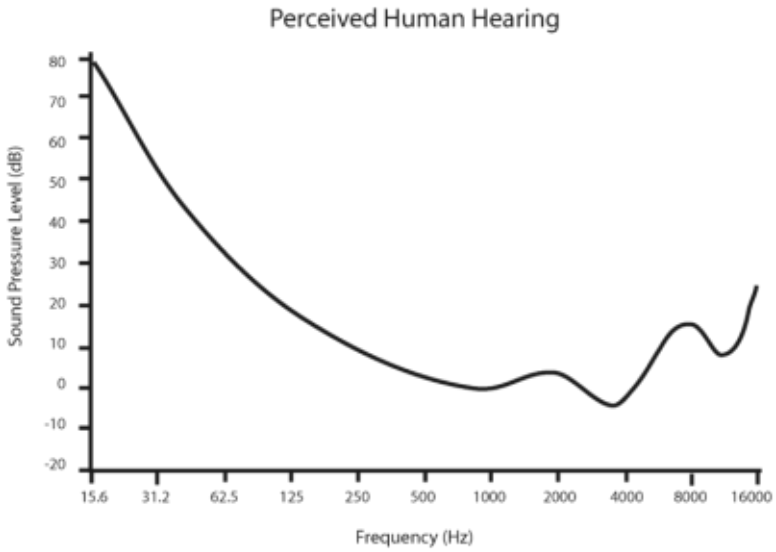
- 8 kHz sampling rate
  - 8 kHz sampling
  - 8 bits / sample
  - mono, µ-Law
- Stats
  - 10s = 80 KB
  - 64 kbps
  - Works on our 56K modem!
  - WAV

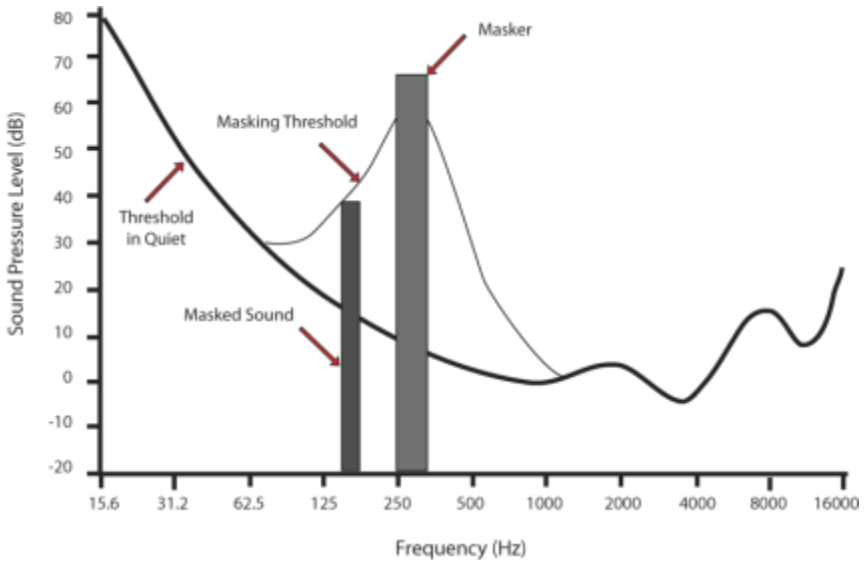
MPEG

- Motion Picture Experts Group (MPEG)
  - Formed to set audio/video compression standards
- MPEG 1
  - First phase started in 1988
  - Compress VHS quality video/audio to 1.5 Mbps
  - Layer 1, 2, 3 of increasing complexity
  - Audio layer 3 = MP3
- MPEG 2
  - Additional extensions
  - Advanced Audio Coding (AAC)
  - Greater sound quality at the same bit rate

MP3

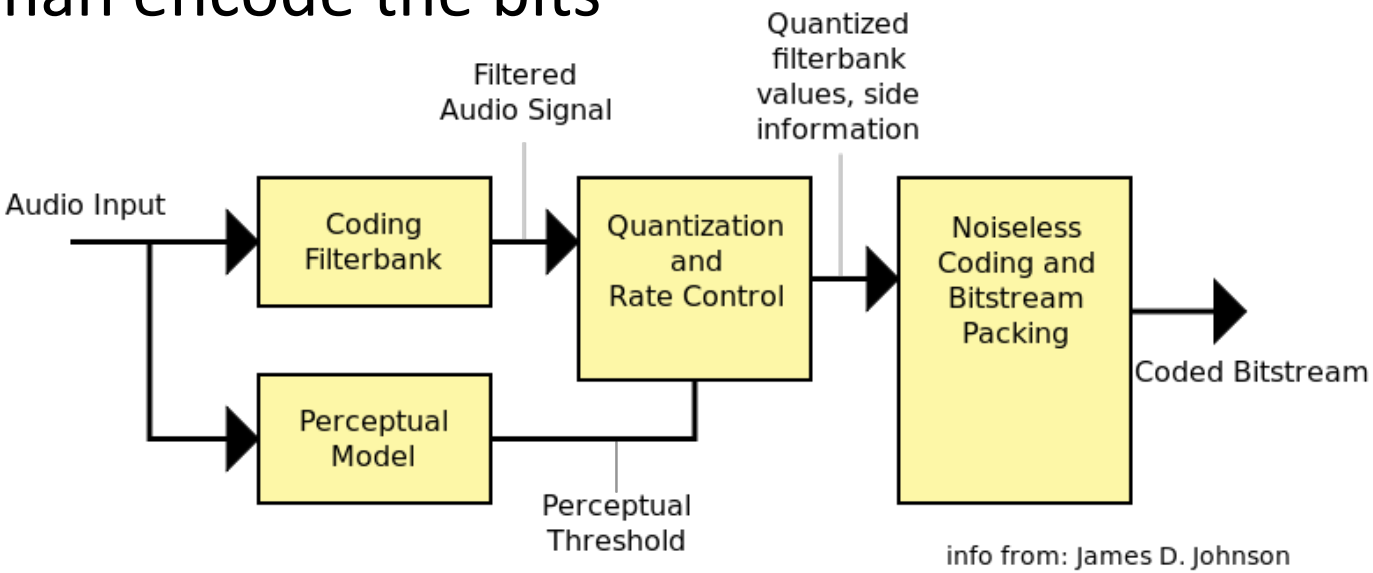
- Perceptual coding
  - Exploit flaws in the human auditory system
  - Psychoacoustics, encode signal so it sounds the same even though you've dropped information
    - Frequency masking, loud sound masks nearby weak one
    - Temporal masking, sudden sound make sounds inaudible slightly before (20 ms) and after (100 ms)



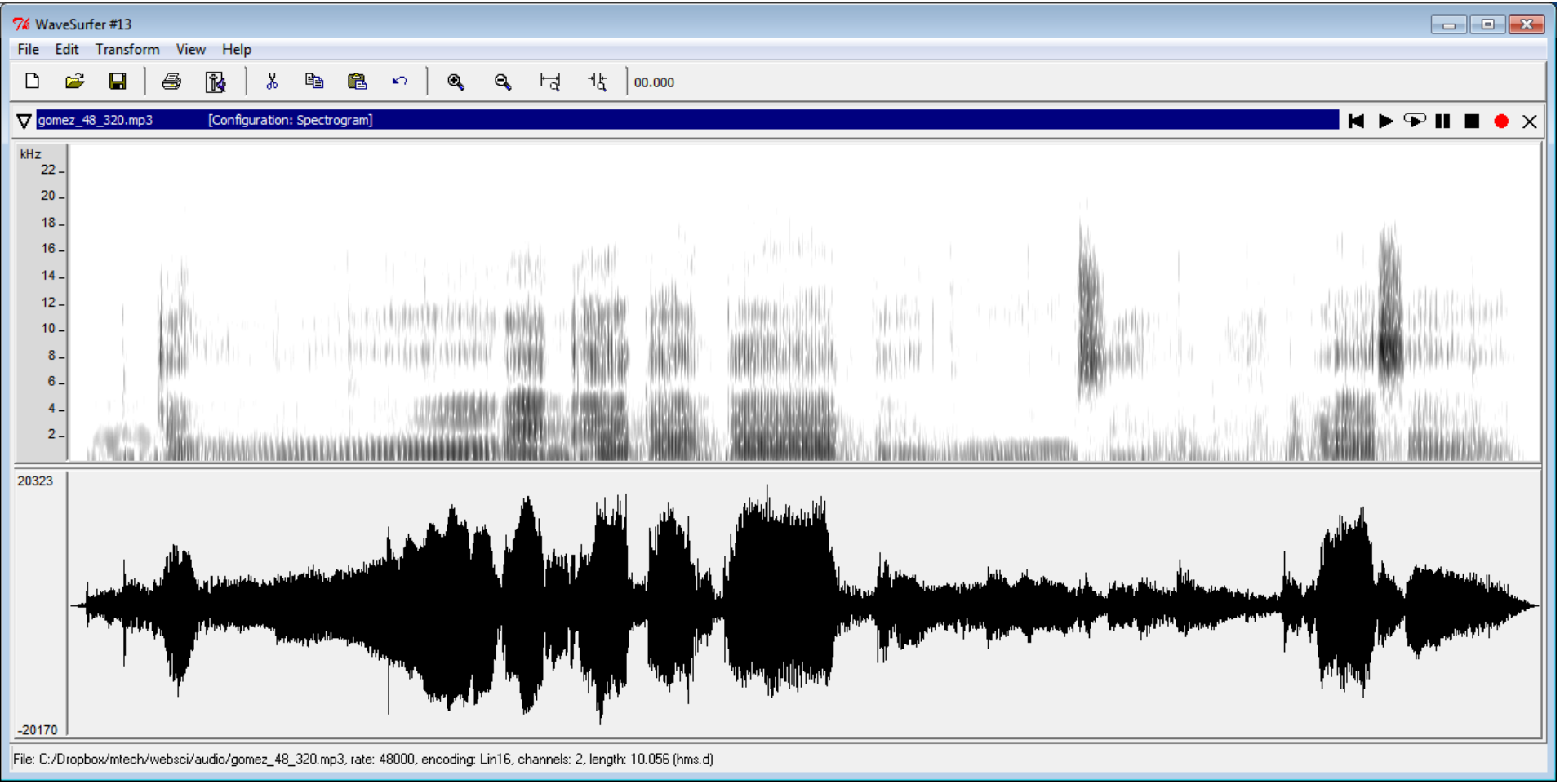


MP3

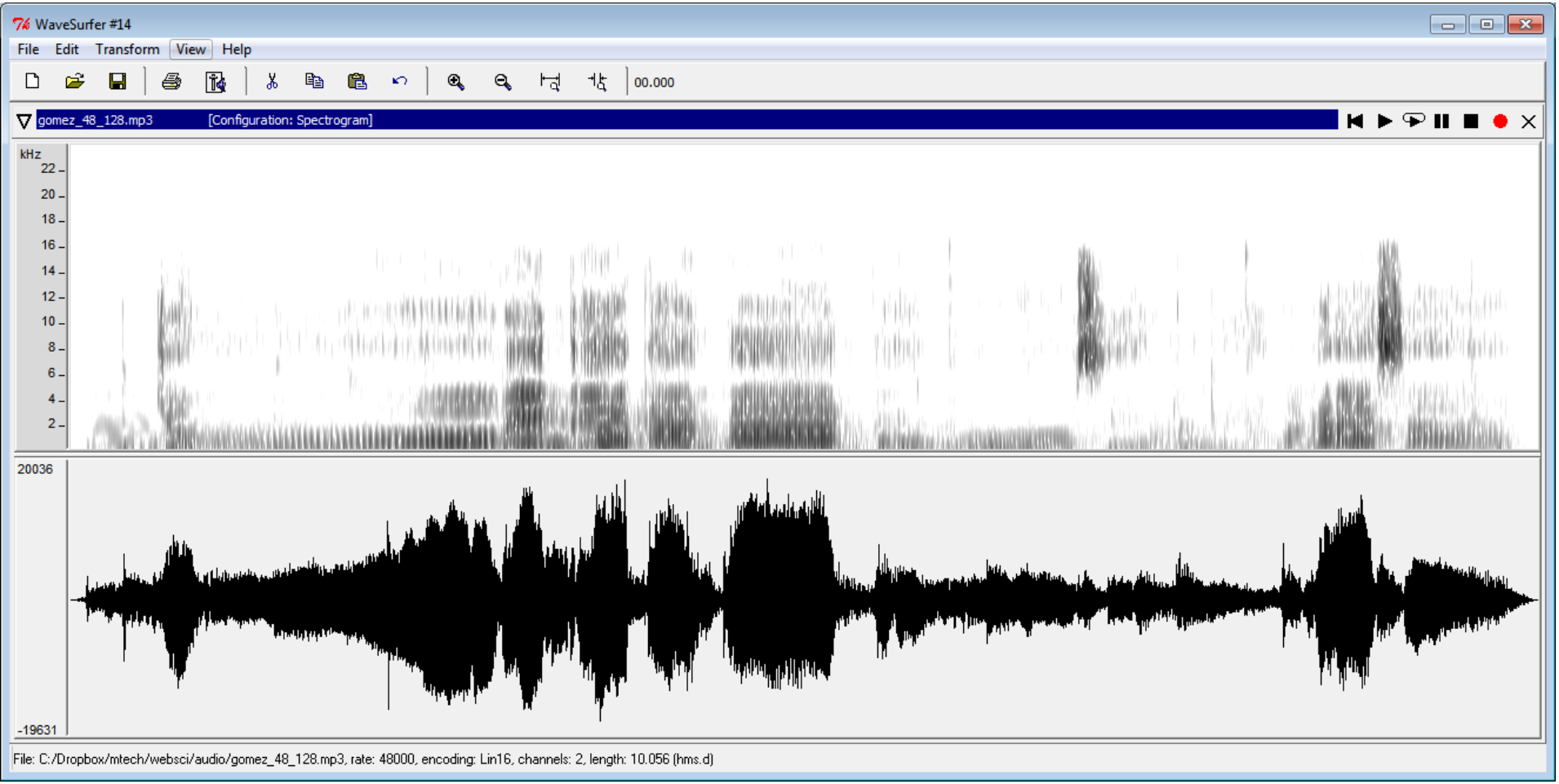
- Perceptual audio coding
  - Process audio in small batches
  - Pass through digital filters to get frequency bands
  - Psychoacoustic model determines masked freqs
  - Available bit budget divides among bands
    - More bits to bands with more spectral power
  - Huffman encode the bits



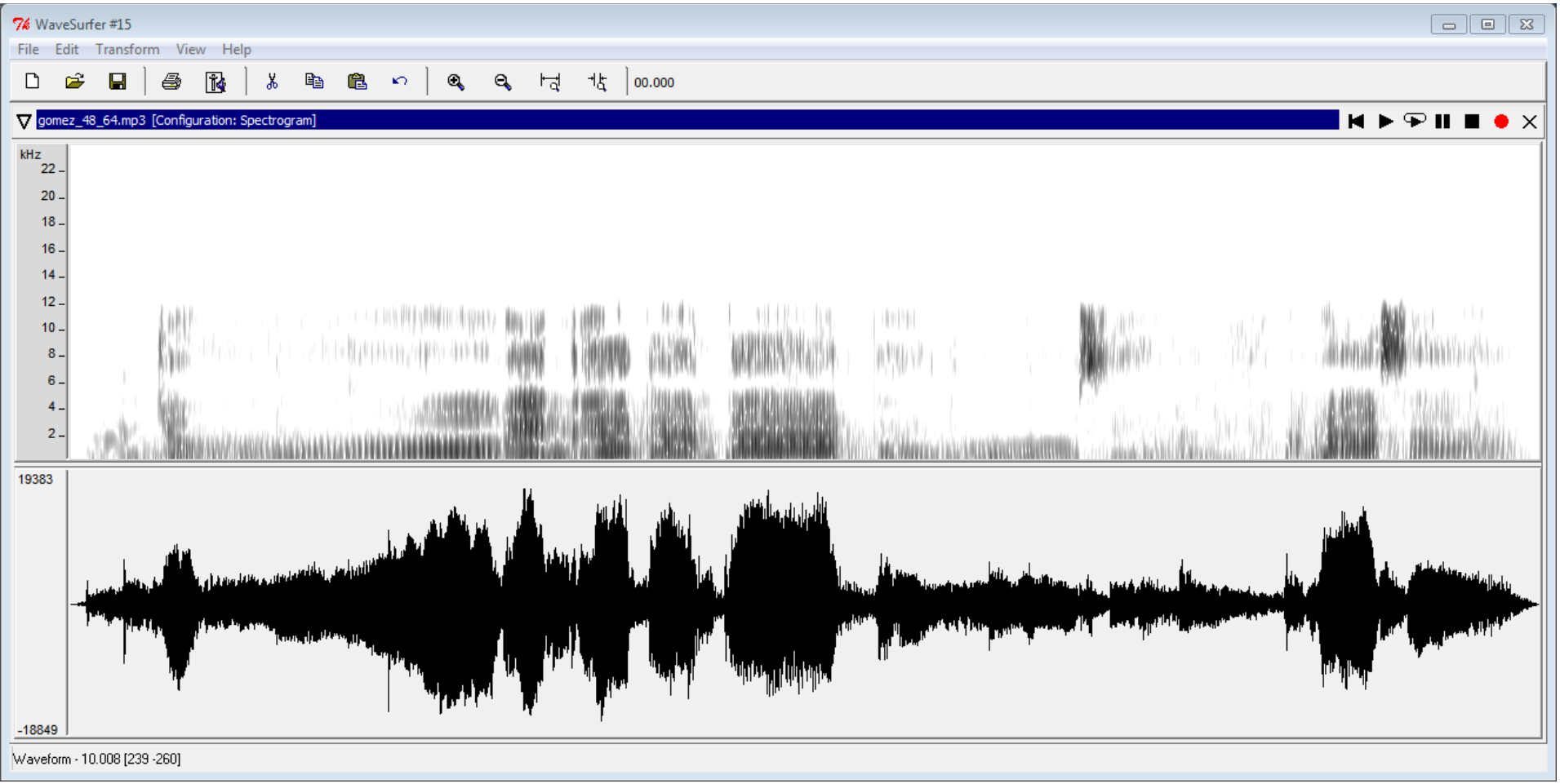
info from: James D. Johnson



- MP3
  - 48 kHz sampling
  - 320 kbps
  - 2 channels
- Stats
  - 10s = 402 KB
  - 322 Kbps
  - MPEG-1 audio layer 3



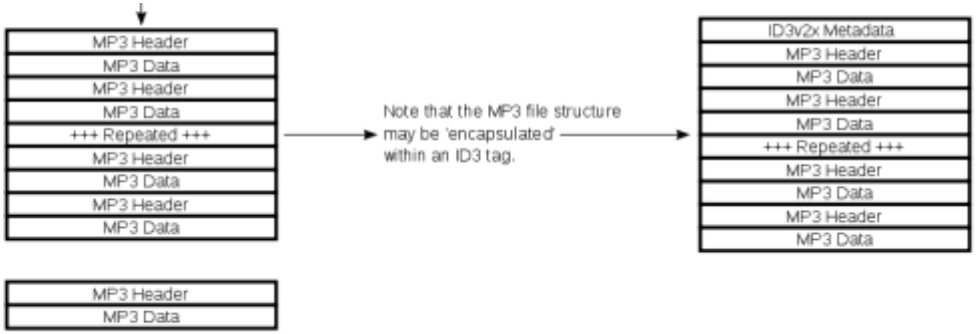
- MP3
  - 48 kHz sampling
  - 128 kbps
  - 2 channels
- Stats
  - 10s = 161 KB
  - 129 Kbps
  - MPEG-1 audio layer 3



- MP3
  - 48 kHz sampling
  - 64 kbps
  - 2 channels
- Stats
  - 10s = 81 KB
  - 65 Kbps
  - MPEG-1 audio layer 3

MP3 format

- Mandatory header with every frame
  - Every 24ms at 48 kHz sampling rate
  - You can drop into a live stream at any point



FFFBA040 Colour-coding shows binary bit mapping to hex values below

Bits	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Binary	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Hex	F			F			F							B			A				0		0				4				0	
Meaning	MP3 Sync Word												Version	Layer		Error Protection	Bit Rate				Frequency		Pad. Bit	Priv. Bit	Mode		Mode Extension (Used With Joint Stereo)		Copy	Original	Emphasis	
Value	Sync Word												1 = MPEG	01 = Layer 3		1 = No	1010 = 160				00 = 44100 Hz		0 = Frame is not padded	Unknown	01 = Joint Stereo		0 = Intensity Stereo Off	0 = MS Stereo Off	0 = Not Copy-righted	0 = Copy Of Original Media	00 = None	

Audio file sizes

```
04/29/2012  04:41 PM         5,766,528 gomez_24_96.wav
04/29/2012  03:48 PM         3,411,865 gomez_24_96.flac

04/29/2012  06:43 PM           402,499 gomez_48_320.mp3
04/29/2012  06:49 PM           241,603 gomez_48_192.mp3
04/29/2012  07:12 PM           161,539 gomez_24_128.mp3
04/29/2012  07:11 PM           161,173 gomez_44_128.mp3
04/29/2012  06:50 PM           161,155 gomez_48_128.mp3
04/29/2012  06:51 PM           120,931 gomez_48_96.mp3
04/29/2012  06:53 PM            80,707 gomez_48_64.mp3
04/29/2012  06:53 PM            40,483 gomez_48_32.mp3
04/29/2012  06:54 PM            10,483 gomez_8_8.mp3
```

JPEG

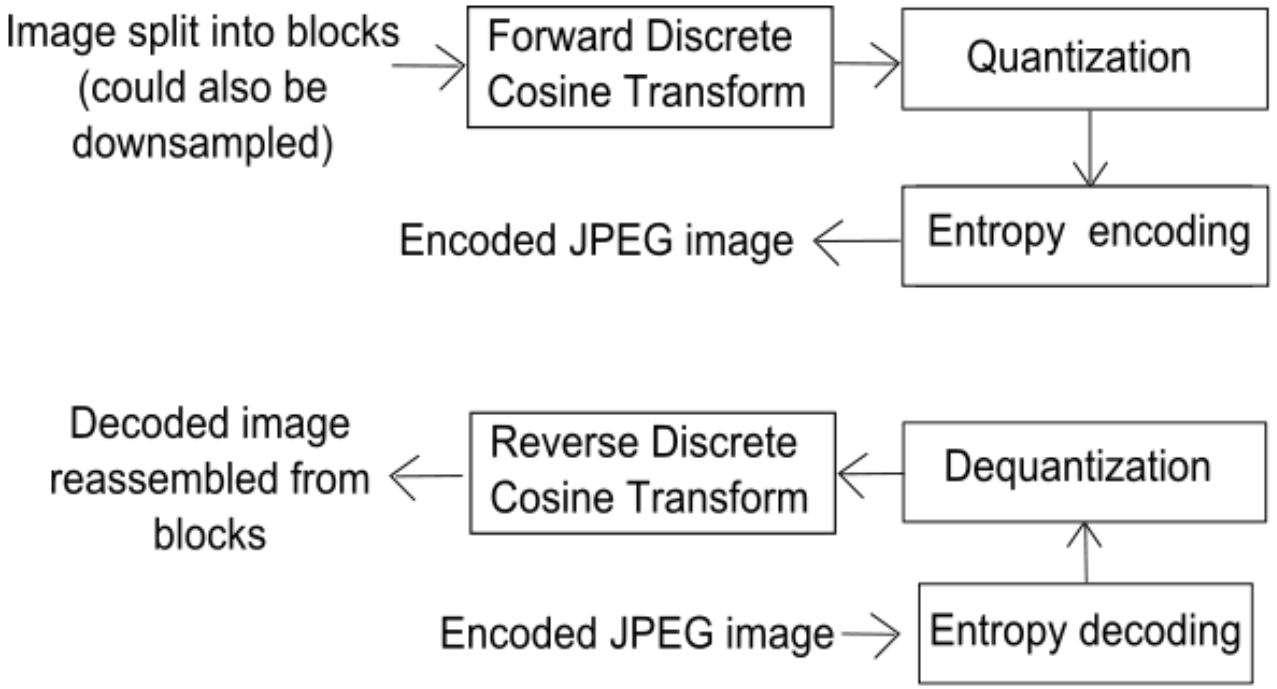
- Joint Photographic Experts Group (JPEG)
  - Compressing continuous-tone still pictures
    - e.g. photos, paintings
  - Not good for sharp changes in color
    - e.g. line drawings, text
  - Not good for repeated edits
  - Inherently a lossy process
    - Quality level 0 (low) to 100 (high)

Lossless image, 1920x1280 = 13.4MB

JPEG file sizes

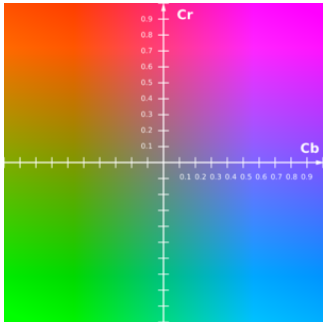
```
04/29/2012  08:09 PM        13,454,982 bike.tif

04/29/2012  08:11 PM         2,617,520 bike_q100.jpg
04/29/2012  08:12 PM           736,695 bike_q90.jpg
04/29/2012  08:12 PM           431,162 bike_q80.jpg
04/29/2012  08:13 PM           321,685 bike_q70.jpg
04/29/2012  08:14 PM           260,027 bike_q60.jpg
04/29/2012  08:14 PM           222,118 bike_q50.jpg
04/29/2012  08:25 PM           196,832 bike_q40.jpg
04/29/2012  08:25 PM           177,614 bike_q30.jpg
04/29/2012  08:26 PM           162,819 bike_q20.jpg
04/29/2012  08:26 PM           151,471 bike_q10.jpg
04/29/2012  08:26 PM           141,793 bike_q0.jpg
```

4-step process

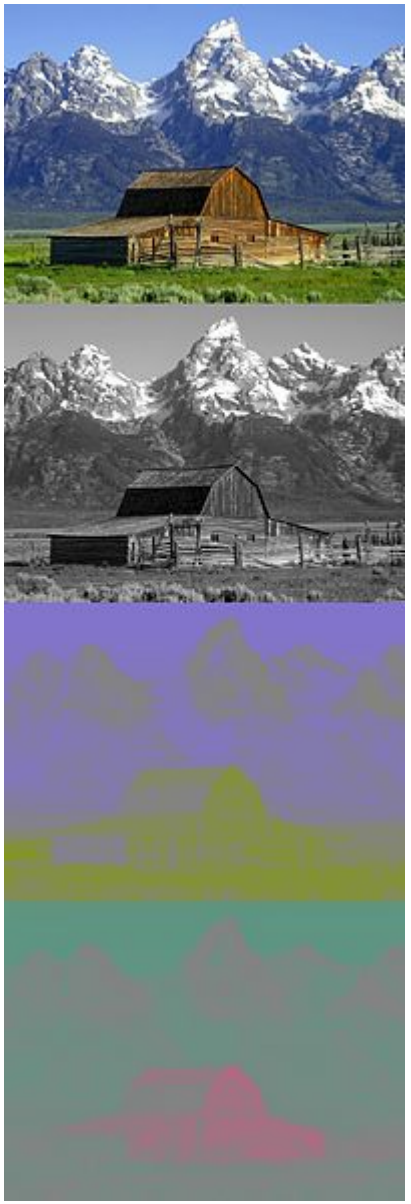


Step 1: JPEG compression

- Convert RGB colors to YCbCr
  - `Y  =  16 + 0.26R + 0.50G + 0.09B`
  - `Cb = 128 + 0.15R - 0.29G - 0.44B`
  - `Cr = 128 + 0.44R - 0.37G + 0.07B`
  - More info in luminance channel (Y)
- Separate channels into matrices
  - Assuming image is 640 x 480:
    - Reduce Cb and Cr to 320 x 240 by averaging square blocks of 4 pixels
    - Split matrices into 8 x 8 blocks
    - Subtract 128 from every element
  - Lossy, but hard to notice



Orig

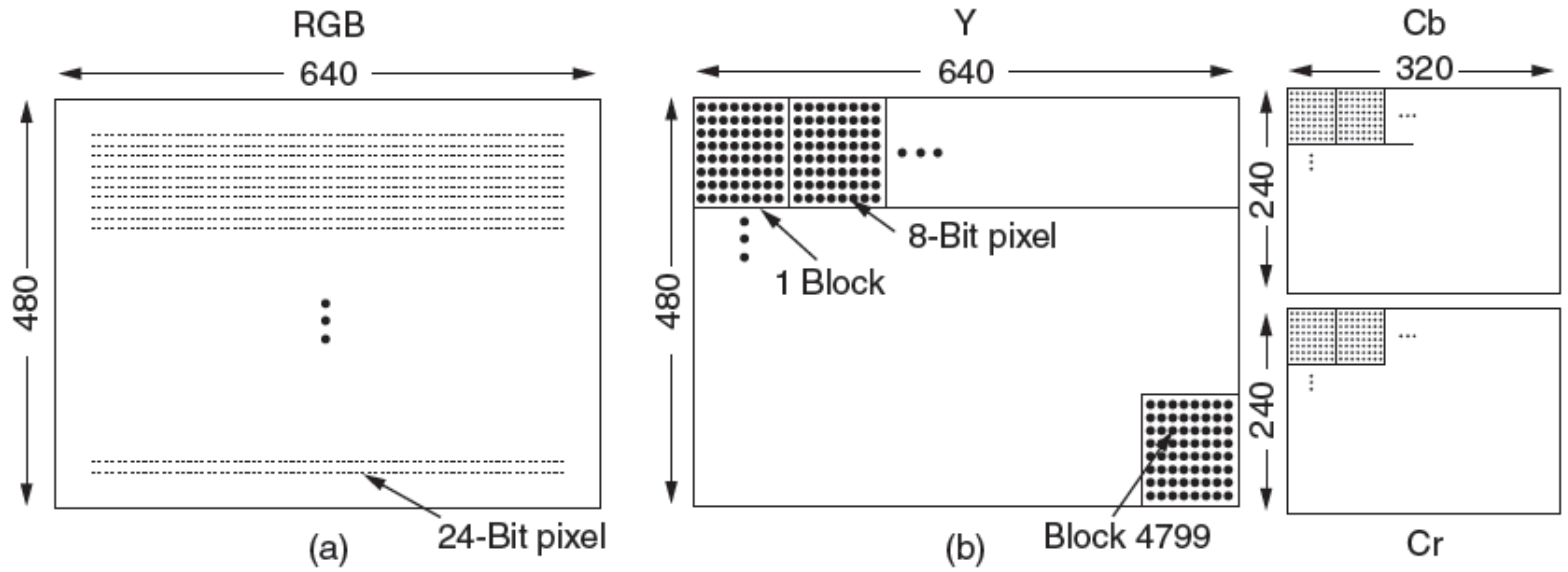


Y

Cb

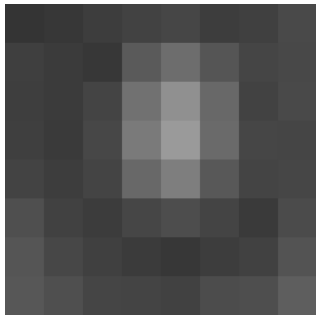
Cr

Step 2: JPEG compression



- Discrete Cosine Transformation (DCT)
  - Applied to each 8 x 8 block separately
  - Output is an 8 x 8 matrix of real numbers
  - DCT (0, 0) is average value of the block
  - Values typically decay rapidly away from (0, 0)
  - DCT not itself lossy (given sufficient precision)

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}.$$



$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x,y)$$
$$\times \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

$$pixel(x,y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j)DCT(i,j)$$
$$\times \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right]$$

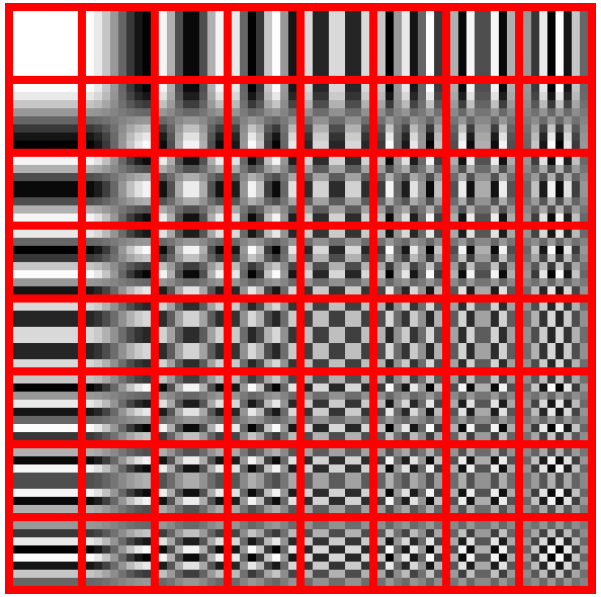
$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

$$g = \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

($x \longrightarrow$, $y \downarrow$)

$$G = \begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.13 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.88 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}$$

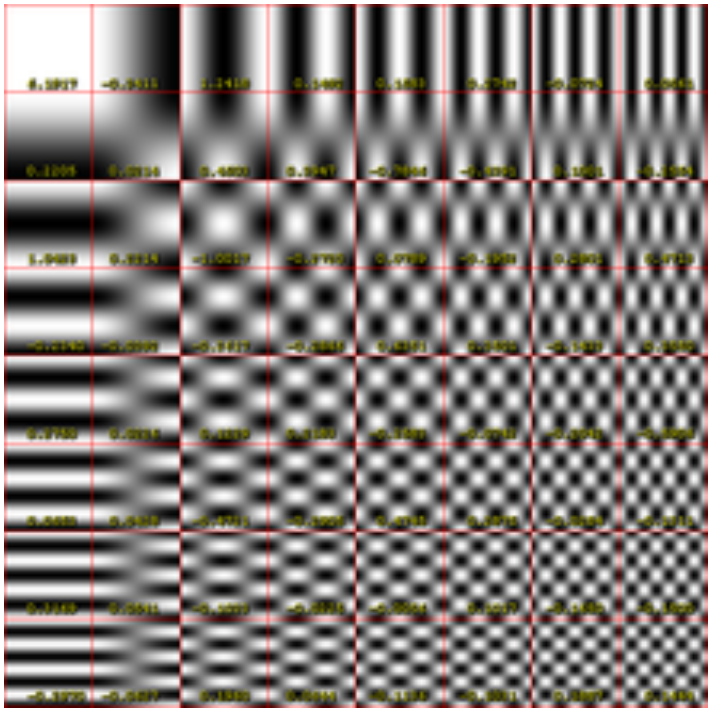
($u \longrightarrow$, $v \downarrow$)



8x8

A

$$\begin{bmatrix} 6.1917 & -0.3411 & 1.2418 & 0.1492 & 0.1583 & 0.2742 & -0.0724 & 0.0561 \\ 0.2205 & 0.0214 & 0.4503 & 0.3947 & -0.7846 & -0.4391 & 0.1001 & -0.2554 \\ 1.0423 & 0.2214 & -1.0017 & -0.2720 & 0.0789 & -0.1952 & 0.2801 & 0.4713 \\ -0.2340 & -0.0392 & -0.2617 & -0.2866 & 0.6351 & 0.3501 & -0.1433 & 0.3550 \\ 0.2750 & 0.0226 & 0.1229 & 0.2183 & -0.2583 & -0.0742 & -0.2042 & -0.5906 \\ 0.0653 & 0.0428 & -0.4721 & -0.2905 & 0.4745 & 0.2875 & -0.0284 & -0.1311 \\ 0.3169 & 0.0541 & -0.1033 & -0.0225 & -0.0056 & 0.1017 & -0.1650 & -0.1500 \\ -0.2970 & -0.0627 & 0.1960 & 0.0644 & -0.1136 & -0.1031 & 0.1887 & 0.1444 \end{bmatrix}$$



Step 3: JPEG compression

- Quantization
  - Divide the DCT coefficients by a table of weights
    - Different tables for different JPEG quality levels
  - Round the result

$$G = \begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.13 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.88 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}$$

($u \rightarrow$, $v \downarrow$)

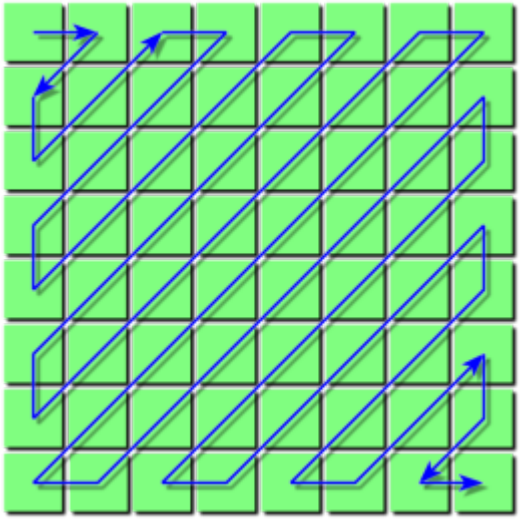
$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 4: JPEG compression

- Encoding the elements
  - Zig-zag pattern to try and get lots of 0's in a row
    - Run-length encoding (RLE)
  - Huffman encode the result

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$



Summary

- MP3
  - Most prevalent lossy audio format
  - Psychoacoustic model
    - Throw away things from audio you can't hear anyway
- JPEG
  - Most prevalent lossy still image format
  - Change colorspace and downsample
  - Lossy:
    - Ignore less important changes in 8 x 8 blocks
    - Controlled quality via quantization tables
  - Lossless compress using RLE and Huffman