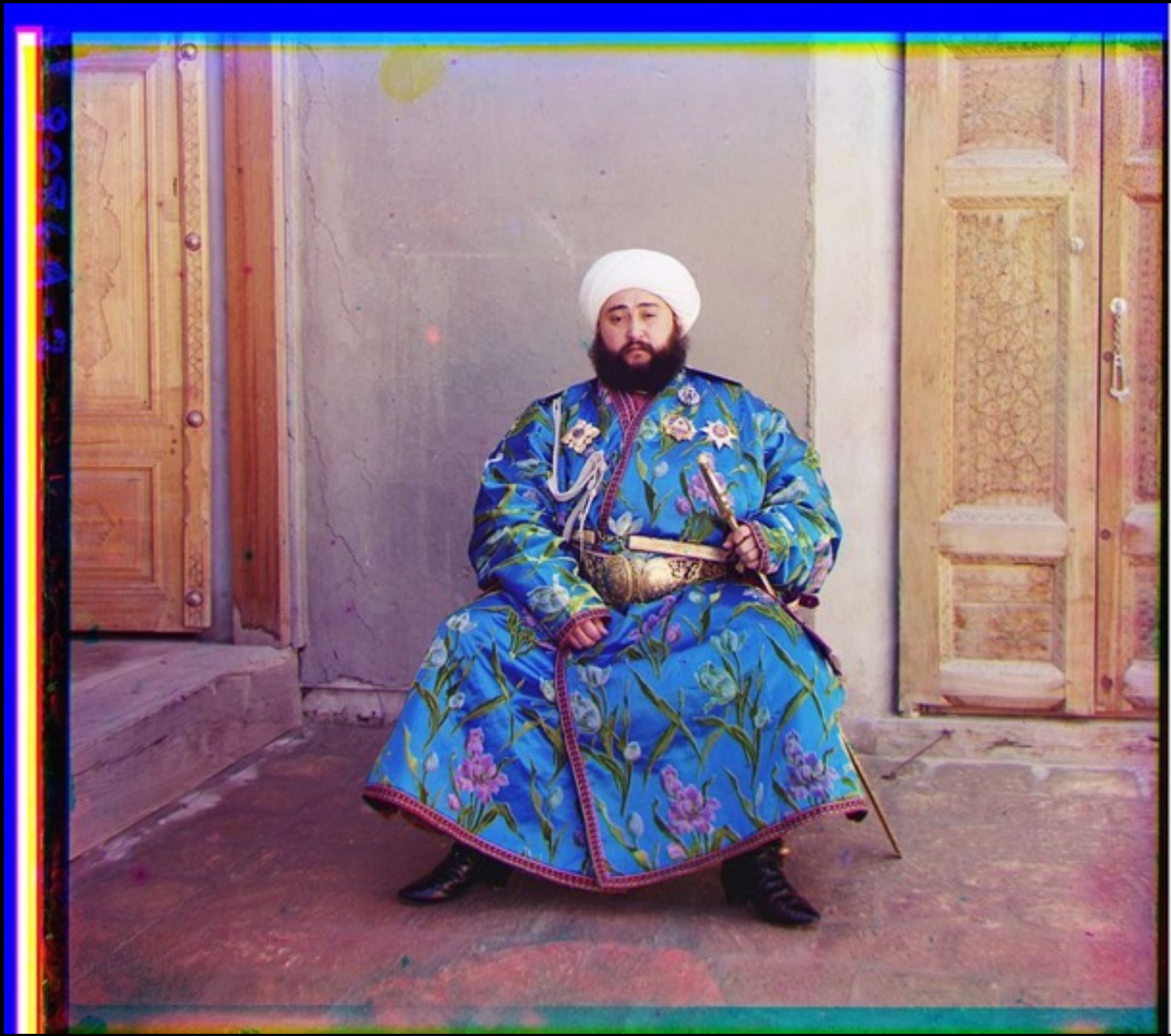
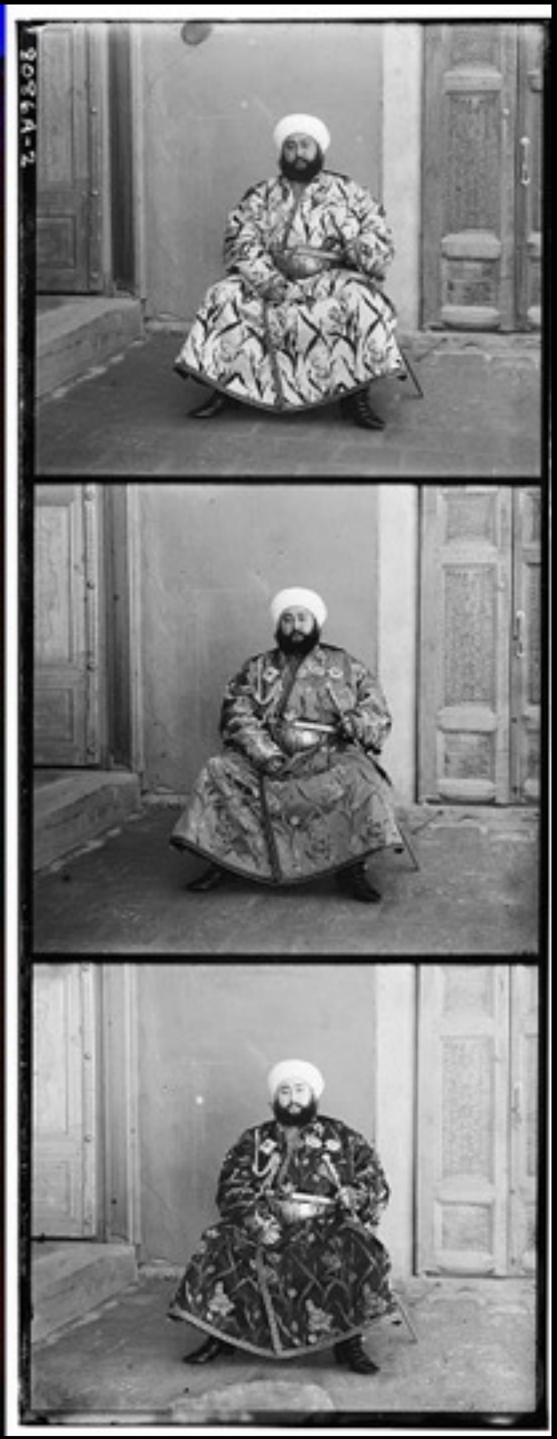
Image compression and recompression

John Graham-Cumming

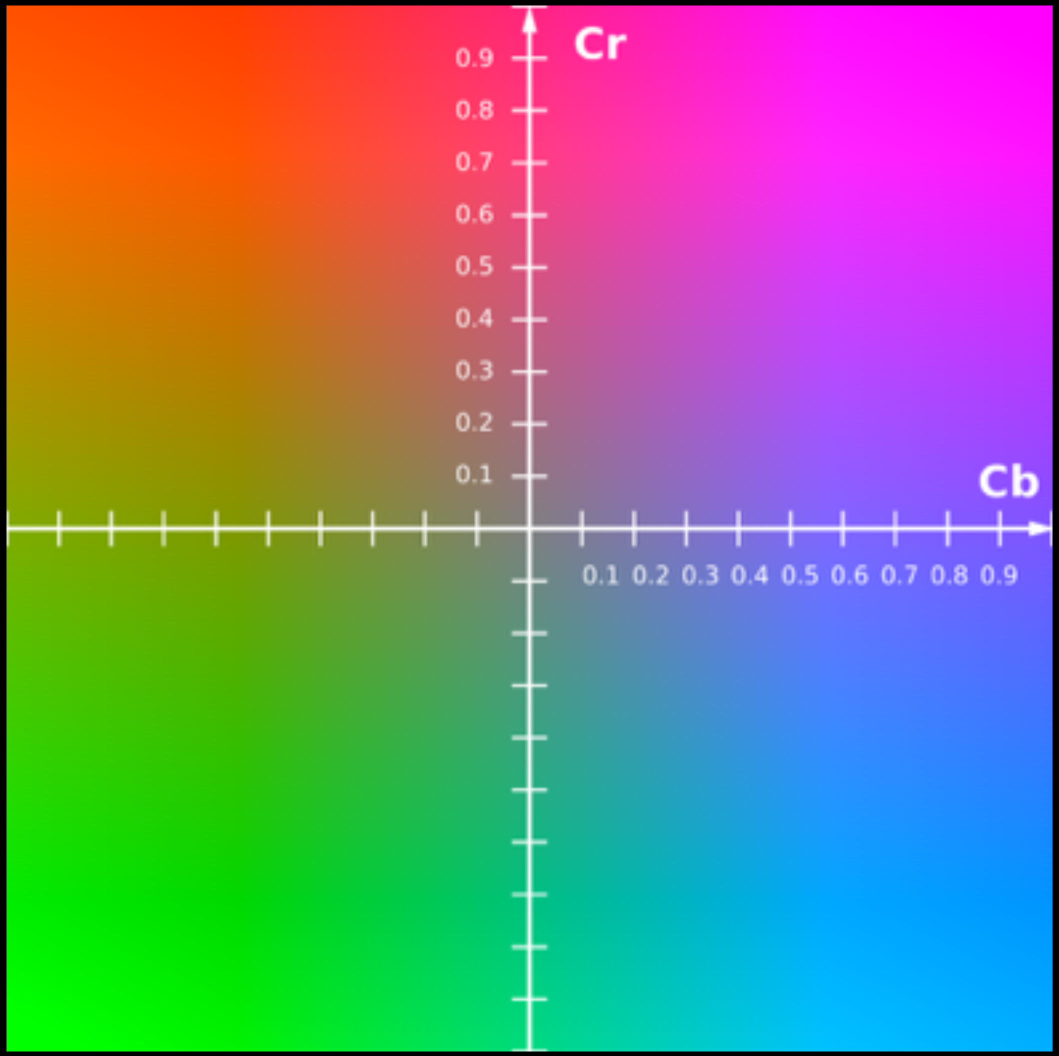
RGB

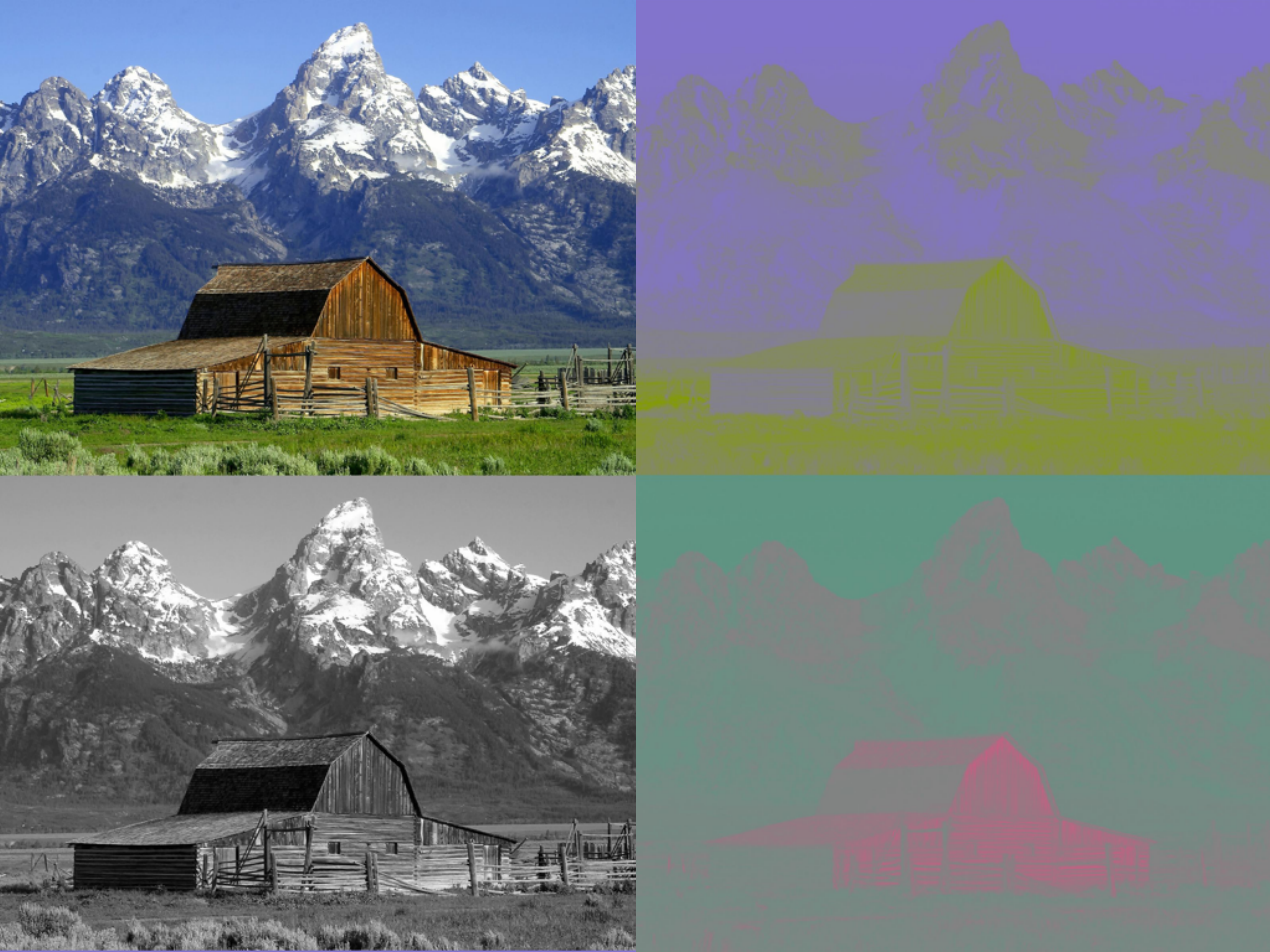




Y’$C_B$$C_R$

- Y’ is luma (brightness of image)
- C_B is blue-difference
- C_R is red-difference



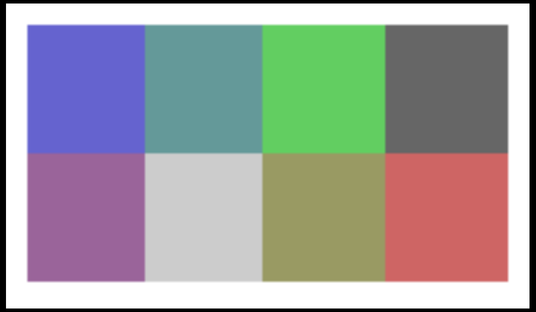


Conversion RGB / Y’C$_B$C$_R$

$$
\begin{aligned}
R &= Y + 1.402 \cdot (C_R - 128) \\
G &= Y - 0.34414 \cdot (C_B - 128) - 0.71414 \cdot (C_R - 128) \\
B &= Y + 1.772 \cdot (C_B - 128)
\end{aligned}
$$

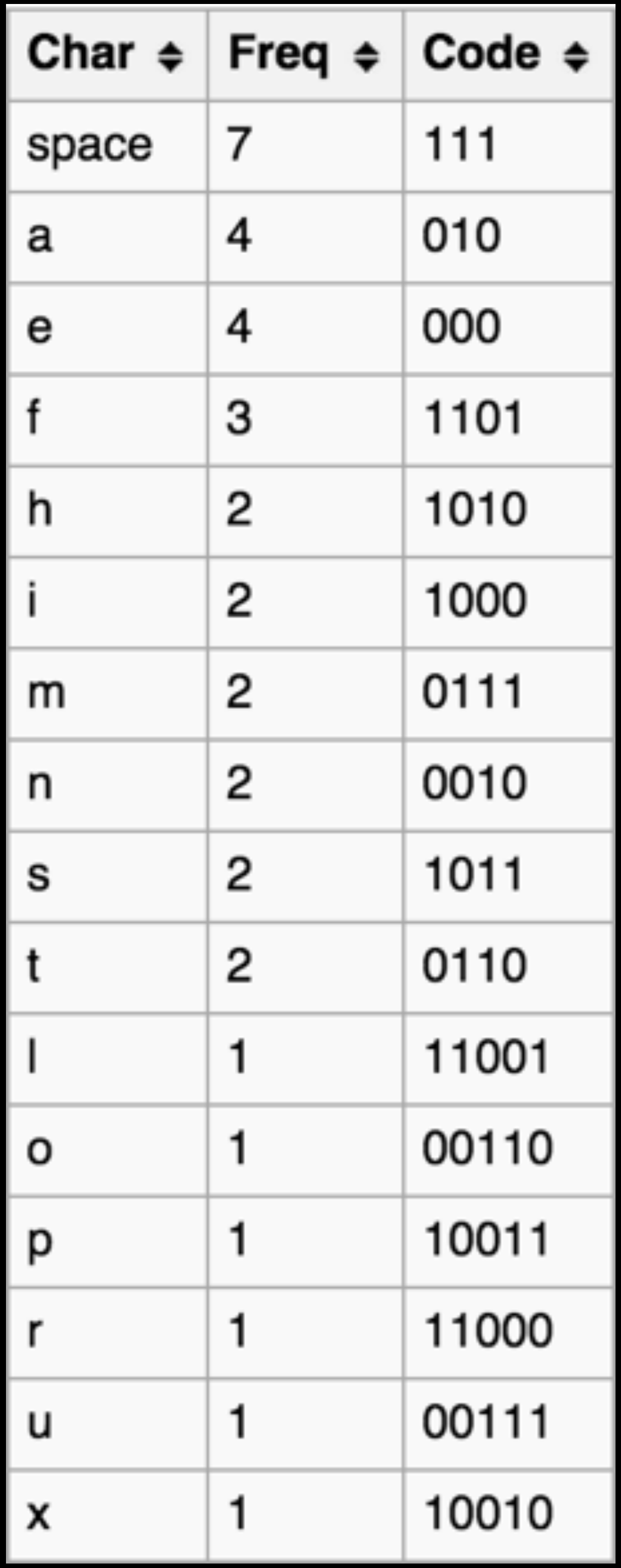
Human Visual System

- Color resolution is poorer than brightness resolution
  - Can compress colors more without upsetting Homo Homo Sapiens
  - Humans are good at brightness, movement, edge and face detection
- Chroma subsampling
  - Sample brightness and colors separately
    - 4:4:4 (brightness and two colors at same rate)
    - 4:2:2 (half sampling for colors)
    - 4:1:1 (horizontal sampling of colors)
    - 4:2:0 (vertical square sampling of colors)



Huffman Coding

- Optimal ‘prefix code’
  - Examples are UTF-8, international dialing codes
  - The prefixes of each code are NOT themselves codes
- Variable length with no need for markers between codes
- Length of code is inversely proportional to frequency of item it encodes



Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010

LZ77

```
We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand
Never gonna give you up<204,13>let you down<204,13>run around<45,5>desert you<20
4,13><184,9>cry<204,13>say goodbye<204,13>tell a lie<45,5>hu<285,7>
We've<30,5>n each<129,7>for so long
Your heart's been aching but
You're too shy to say it
Inside we both<30,6>wha<422,9>going on
We<30,10>game<45,5>we're<210,7>play it
And if you ask me<161,17>Don't tell m<220,5>'re too blind to see
<340,13><217,161><229,12><634,161>(Ooh,<633,12>)<967,24>)<340,13>give,
n<230,11>
give
(G<635,10><1004,57><377,11><389,160><139,239><229,12><634,333>
```

LZ77

We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand
Never gonna give you up<204,13>let you down<204,13>run around<45,5>desert you<20
4,13><184,9>cry<204,13>say goodbye<204,13>tell a lie<45,5>hu<285,7>
We've<30,5>n each<129,7>for so long
Your heart's been aching but
You're too shy to say it
Inside we both<30,6>wha<422,9>going on
We<30,10>game<45,5>we're<210,7>play it
And if you ask me<161,17>Don't tell m<220,5>'re too blind to see
<340,13><217,161><229,12><634,161>(Ooh,<633,12>)<967,24>)<340,13>give,
n<230,11>
give
(G<635,10><1004,57><377,11><389,160><139,239><229,12><634,333>

LZ77

We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand
Never gonna give you up<204,13>let you down<204,13>run around<45,5>desert you<20
4,13><184,9>cry<204,13>say goodbye<204,13>tell a lie<45,5>hu<285,7>
We've<30,5>n each<129,7>for so long
Your heart's been aching but
You're too shy to say it
Inside we both<30,6>wha<422,9>going on
We<30,10>game<45,5>we're<210,7>play it
And if you ask me<161,17>Don't tell m<220,5>'re too blind to see
<340,13><217,161><229,12><634,161>(Ooh,<633,12>)<967,24>)<340,13>give,
n<230,11>
give
(G<635,10><1004,57><377,11><389,160><139,239><229,12><634,333>

LZ77

We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand
Never gonna give you up<204,13>let you down<204,13>run around<45,5>desert you<20
4,13><184,9>cry<204,13>say goodbye<204,13>tell a lie<45,5>hu<285,7>
We've<30,5>n each<129,7>for so long
Your heart's been aching but
You're too shy to say it
Inside we both<30,6>wha<422,9>going on
We<30,10>game<45,5>we're<210,7>play it
And if you ask me<161,17>Don't tell m<220,5>'re too blind to see
<340,13><217,161><229,12><634,161>(Ooh,<633,12>)<967,24>)<340,13>give,
n<230,11>
give
(G<635,10><1004,57><377,11><389,160><139,239><229,12><634,333>

LZ77

We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand
Never gonna give you up<204,13>let you down<204,13>run around<45,5>desert you<20
4,13><184,9>cry<204,13>say goodbye<204,13>tell a lie<45,5>hu<285,7>
We've<30,5>n each<129,7>for so long
Your heart's been aching but
You're too shy to say it
Inside we both<30,6>wha<422,9>going on
We<30,10>game<45,5>we're<210,7>play it
And if you ask me<161,17>Don't tell m<220,5>'re too blind to see
<340,13><217,161><229,12><634,161>(Ooh,<633,12>)<967,24>)<340,13>give,
n<230,11>
give
(G<635,10><1004,57><377,11><389,160><139,239><229,12><634,333>

LZ77

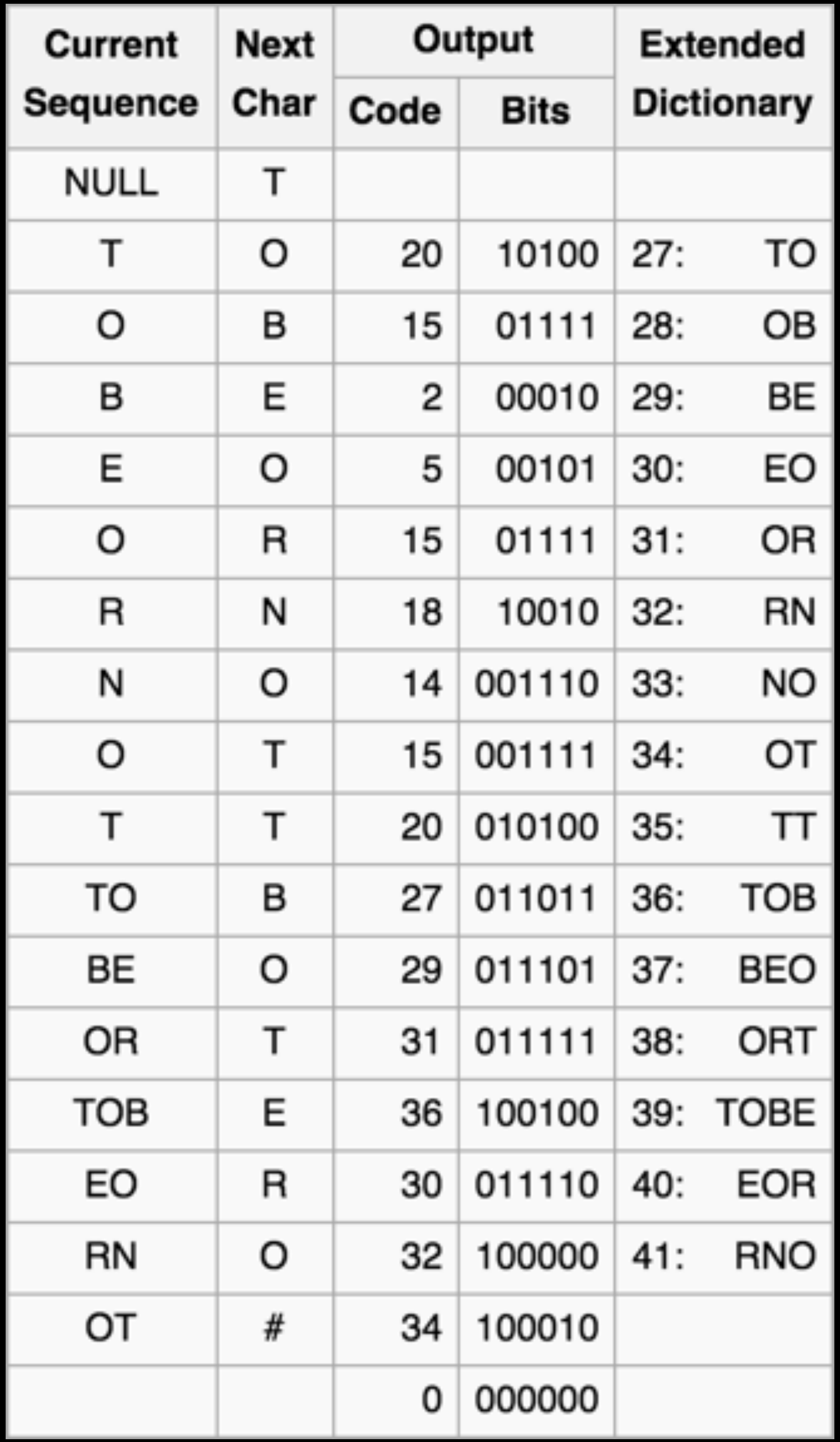
We're no strangers to love
You know the rules and so do I
A full commitment's what I'm thinking of
You wouldn't get this from any other guy
I just wanna tell you how I'm feeling
Gotta make you understand
Never gonna give you up<204,13>let you down<204,13>run around<45,5>desert you<20
4,13><184,9>cry<204,13>say goodbye<204,13>tell a lie<45,5>hu<285,7>
We've<30,5>n each<129,7>for so long
Your heart's been aching but
You're too shy to say it
Inside we both<30,6>wha<422,9>going on
We<30,10>game<45,5>we're<210,7>play it
And if you ask me<161,17>Don't tell m<220,5>'re too blind to see
<340,13><217,161><229,12><634,161>(Ooh,<633,12>)<967,24>)<340,13>give,
n<230,11>
give
(G<635,10><1004,57><377,11><389,160><139,239><229,12><634,333>

DEFLATE

- Two stages:
    - LZ77
    - Huffman Code
- Used in PKZIP, zlib/gzip, PDF, and PNG

LZW

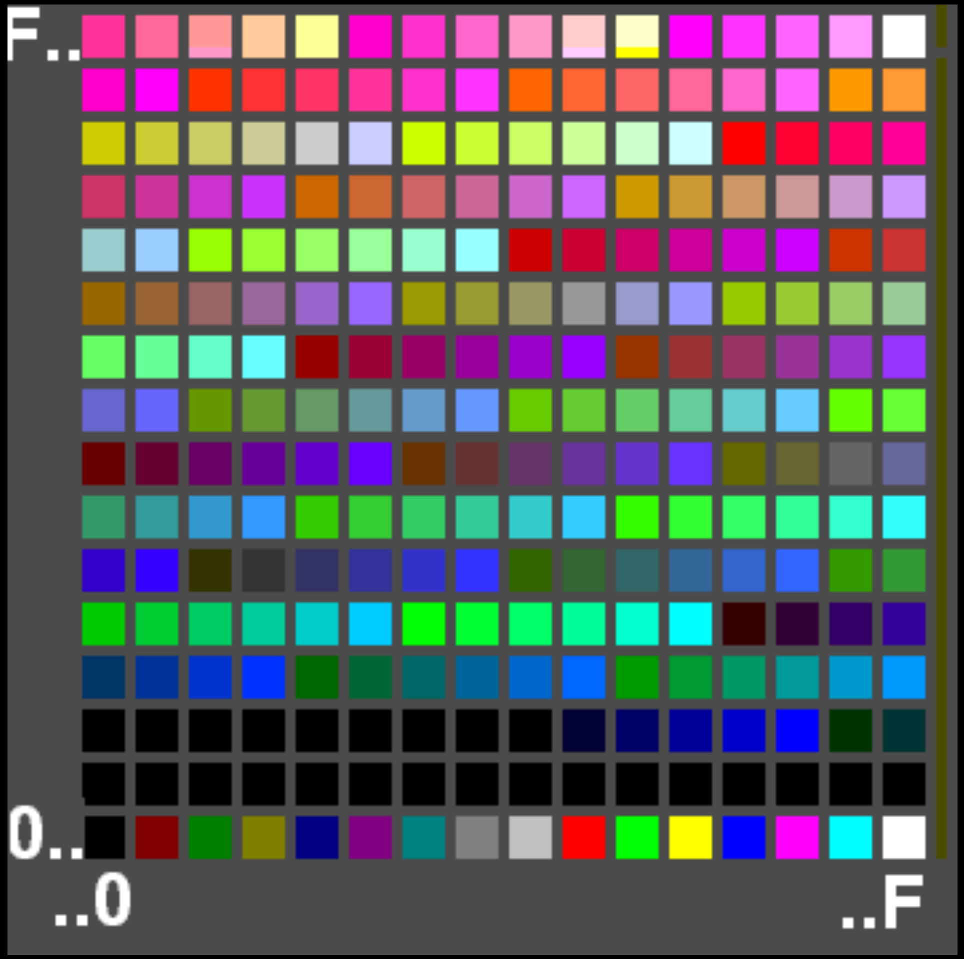
- Works on a known ‘alphabet’
- Adds longer and longer strings to a dictionary
- 12 bit codes for strings from alphabet



Current Sequence	Next Char	Output		Extended Dictionary
		Code	Bits	
NULL	T			
T	O	20	10100	27: TO
O	B	15	01111	28: OB
B	E	2	00010	29: BE
E	O	5	00101	30: EO
O	R	15	01111	31: OR
R	N	18	10010	32: RN
N	O	14	001110	33: NO
O	T	15	001111	34: OT
T	T	20	010100	35: TT
TO	B	27	011011	36: TOB
BE	O	29	011101	37: BEO
OR	T	31	011111	38: ORT
TOB	E	36	100100	39: TOBE
EO	R	30	011110	40: EOR
RN	O	32	100000	41: RNO
OT	#	34	100010	
		0	000000	

GIF

- Uses a variable sized color palette: 2^n bytes
- Palette codes are ‘alphabet’ for LZW encoding
- LZW code sizes determined by alphabet/palette size



GIF Optimization

- gifsicle
  http://www.lcdf.org/gifsicle/

  Good at optimizing animated GIFs

- lossygif
  https://pornel.net/lossygif

  Makes LZW do ‘fuzzy matching’ to lose information

- Main available ‘knob’ in GIF optimization is “reduce size of palette”

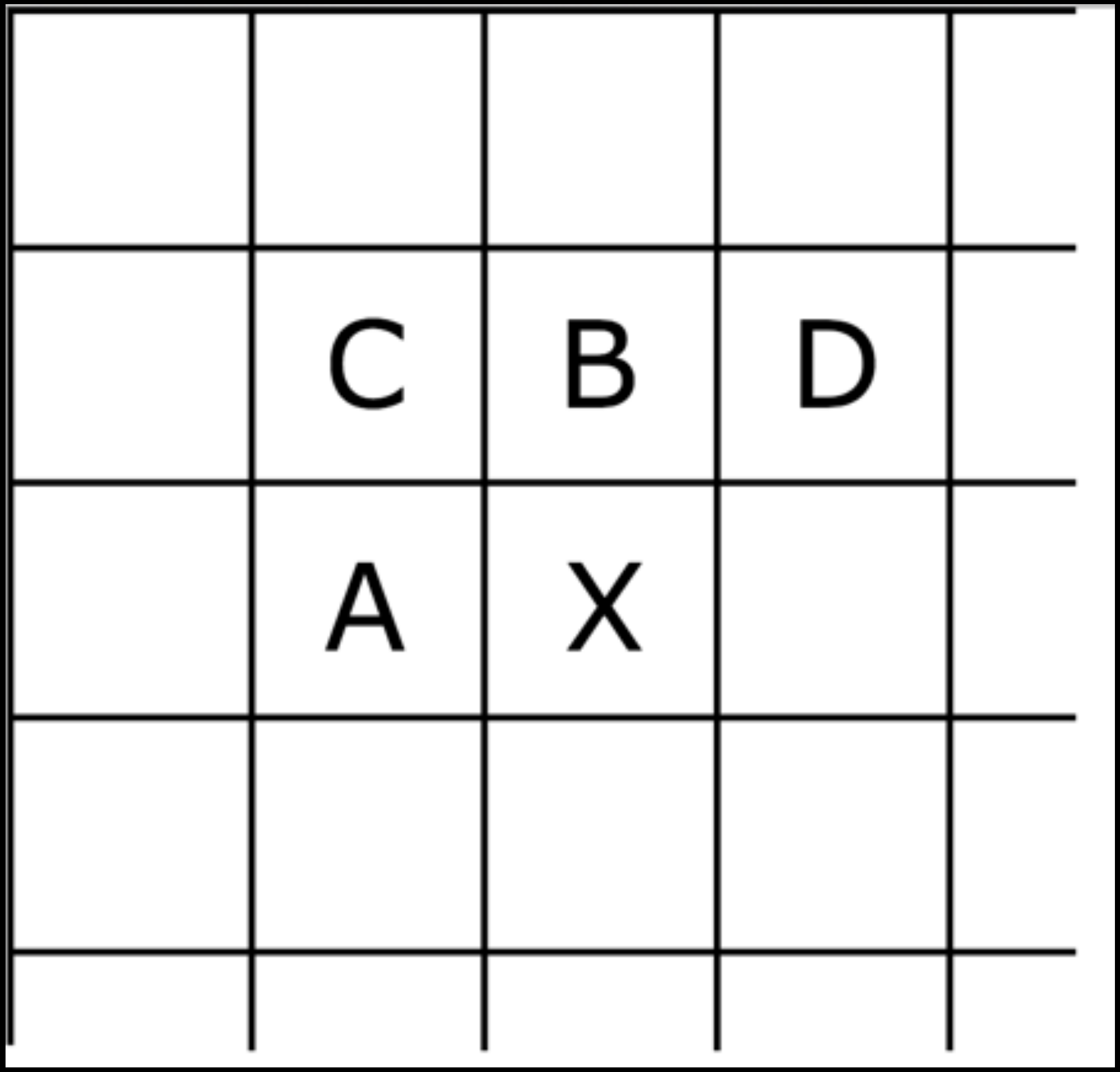
- Best GIF optimization technique for non-animated images is “convert to a PNG”

PNG

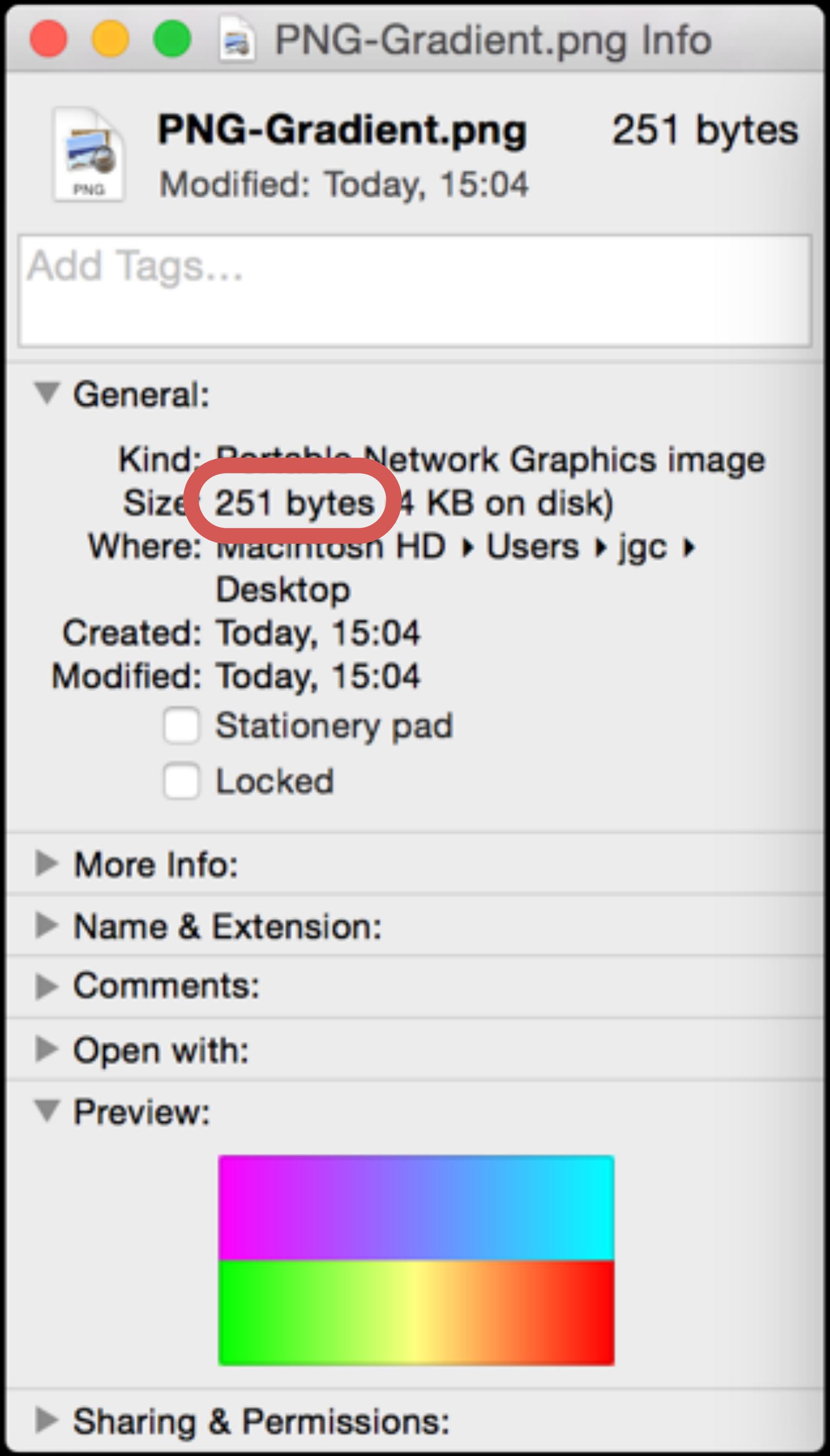
- Has palettized, grayscale or true color modes and transparency
- Two phase compression
  - Filtering
  - DEFLATE
- Compression is done in chunks (IDAT) and can be line by line

PNG Filtering

- Designed to exploit similarities between rows of pixels
- X encoded as difference between
  - Left: X and A
  - Up: X and B
  - Average: X and mean(A, B)
  - Paeth: X and whichever of A, B, C is closest to A + B - C



Extreme filter example



PNG-Gradient.png Info

PNG-Gradient.png 251 bytes
Modified: Today, 15:04

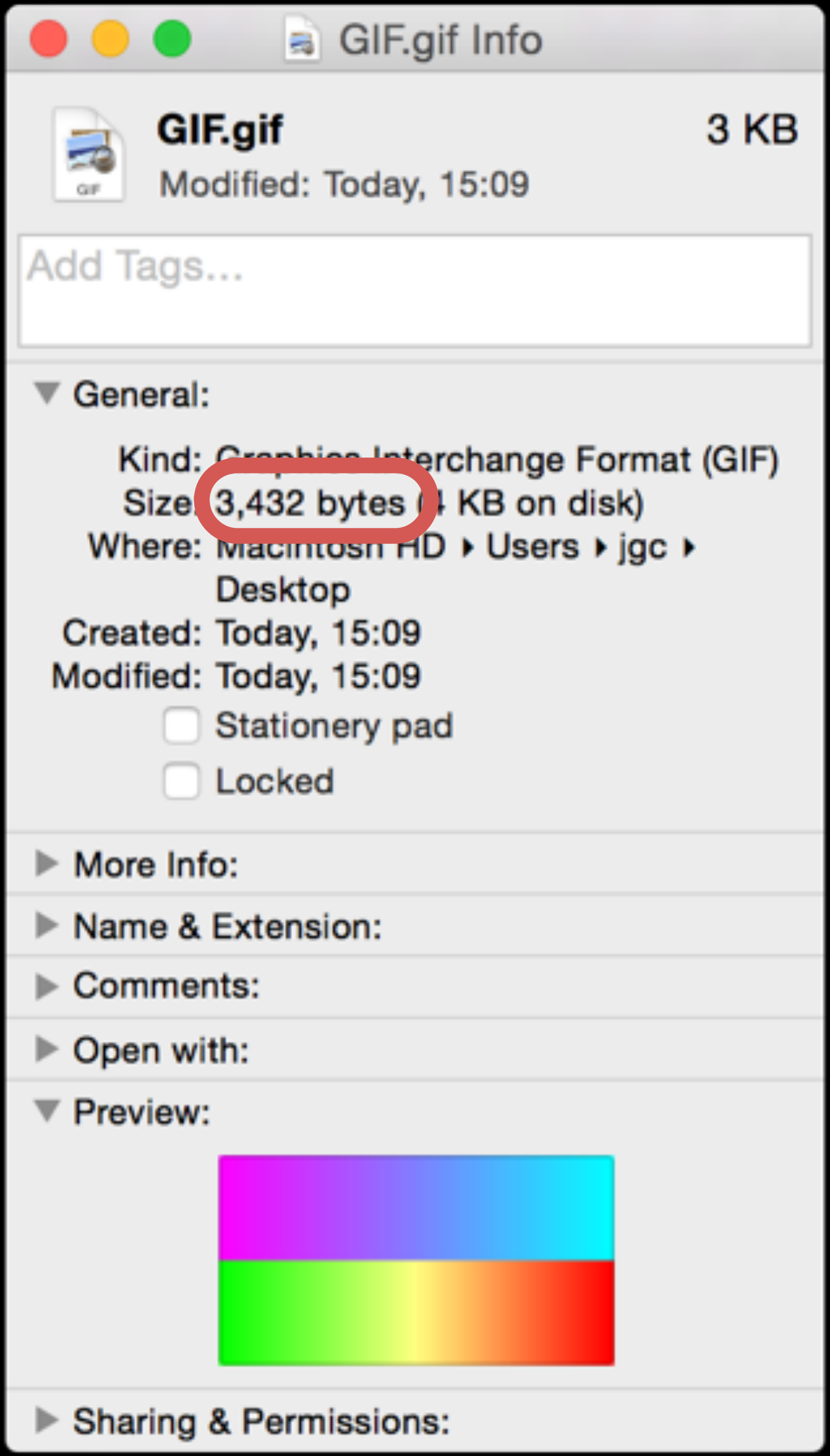
Add Tags...

▼ General:

Kind: Portable Network Graphics image
Size: 251 bytes (4 KB on disk)
Where: Macintosh HD ▸ Users ▸ jgc ▸ Desktop
Created: Today, 15:04
Modified: Today, 15:04
☐ Stationery pad
☐ Locked

▶ More Info:
▶ Name & Extension:
▶ Comments:
▶ Open with:
▼ Preview:

▶ Sharing & Permissions:



GIF.gif Info

GIF.gif 3 KB
Modified: Today, 15:09

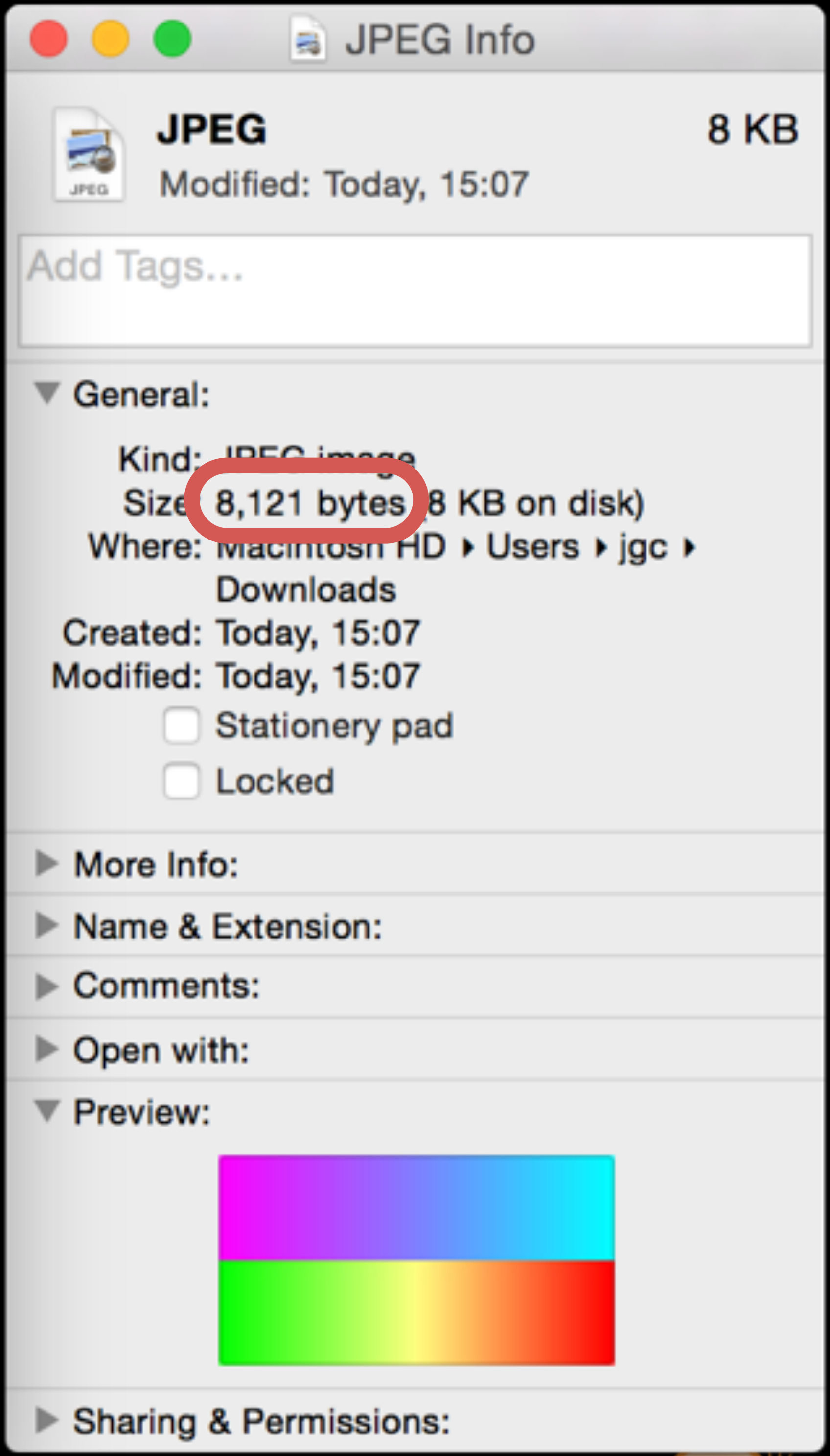
Add Tags...

▼ General:

Kind: Graphics Interchange Format (GIF)
Size: 3,432 bytes (4 KB on disk)
Where: Macintosh HD ▸ Users ▸ jgc ▸ Desktop
Created: Today, 15:09
Modified: Today, 15:09
☐ Stationery pad
☐ Locked

▶ More Info:
▶ Name & Extension:
▶ Comments:
▶ Open with:
▼ Preview:

▶ Sharing & Permissions:



JPEG Info

JPEG 8 KB
Modified: Today, 15:07

Add Tags...

▼ General:

Kind: JPEG image
Size: 8,121 bytes (8 KB on disk)
Where: Macintosh HD ▸ Users ▸ jgc ▸ Downloads
Created: Today, 15:07
Modified: Today, 15:07
☐ Stationery pad
☐ Locked

▶ More Info:
▶ Name & Extension:
▶ Comments:
▶ Open with:
▼ Preview:

▶ Sharing & Permissions:

PNG Optimization

- pngcrush
  http://pmt.sourceforge.net/pngcrush/

  Tries out combinations of filters and DEFLATE parameters *line by line*

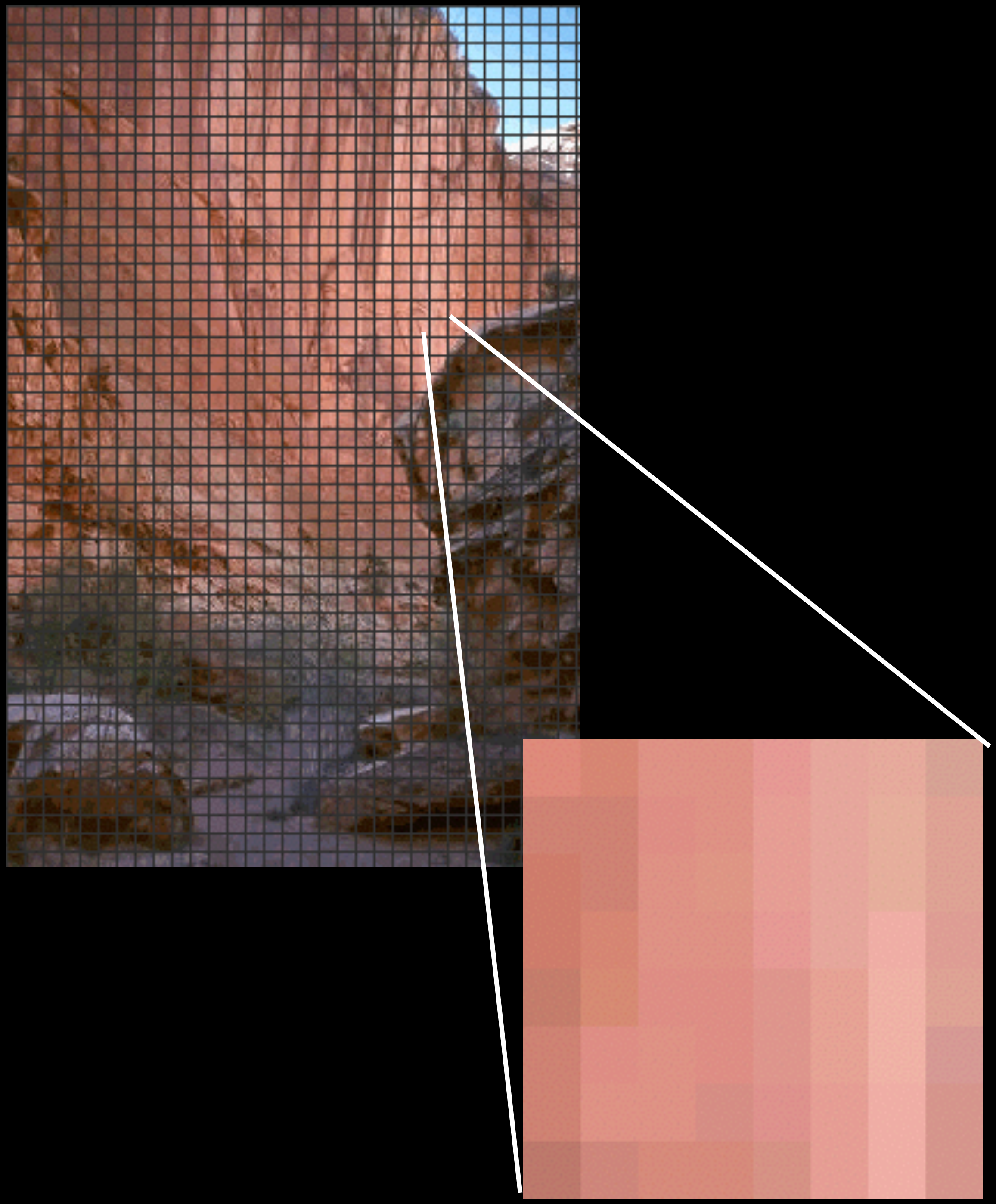
JPEG

- Trivial really 😃 It’s just

1. Blocking
2. Color space conversion to $Y'C_BC_R$
3. Chroma subsampling
4. Conversion to frequency domain via discrete cosine transformation
5. Quantization
6. Entropy coding

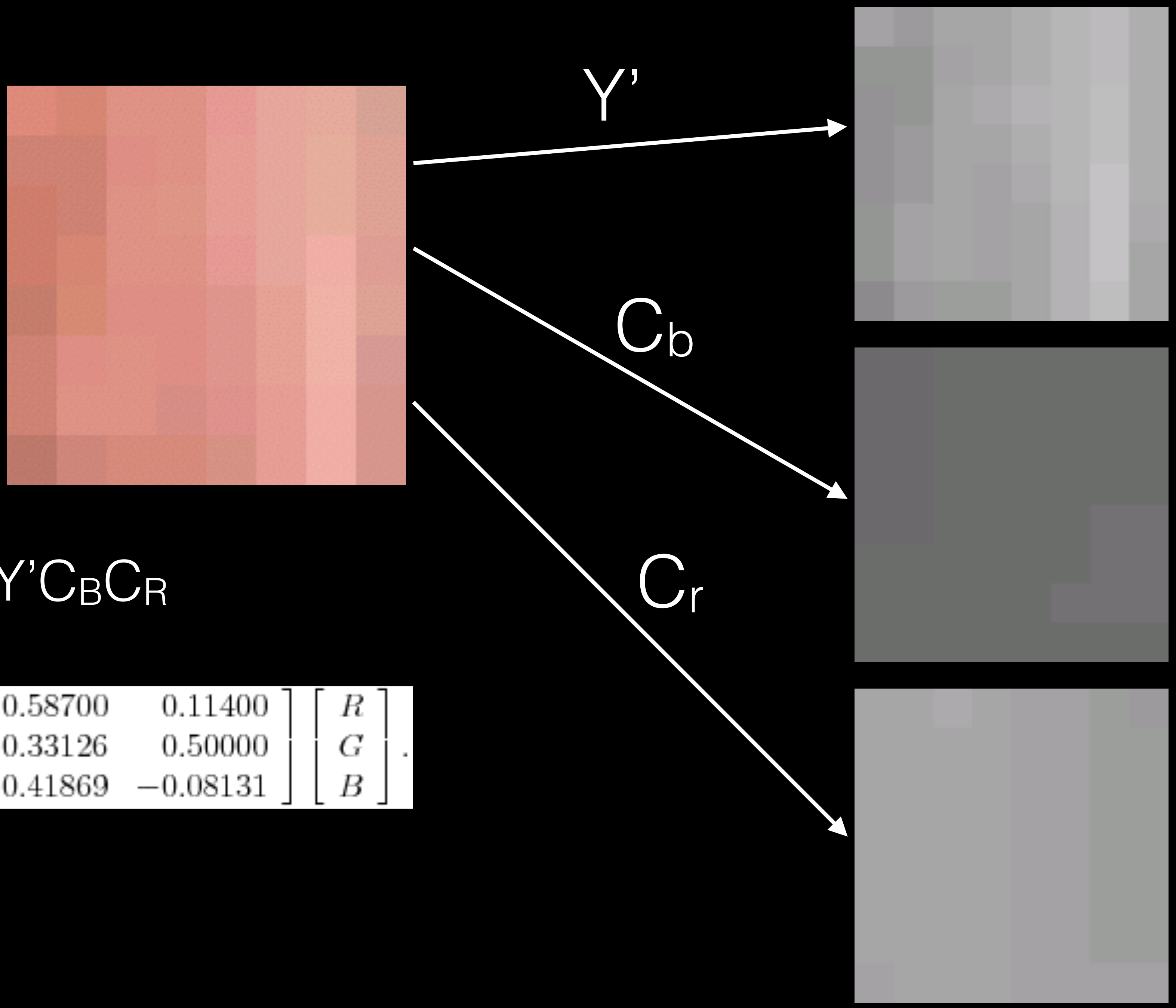
JPEG Blocking

- Divide image up into 8x8 pixel blocks
- Separate blocks for $Y'C_BC_R$
- Rest of algorithm works on these blocks



http://www.ams.org/samplings/feature-column/fcarc-image-compression

Color space conversion



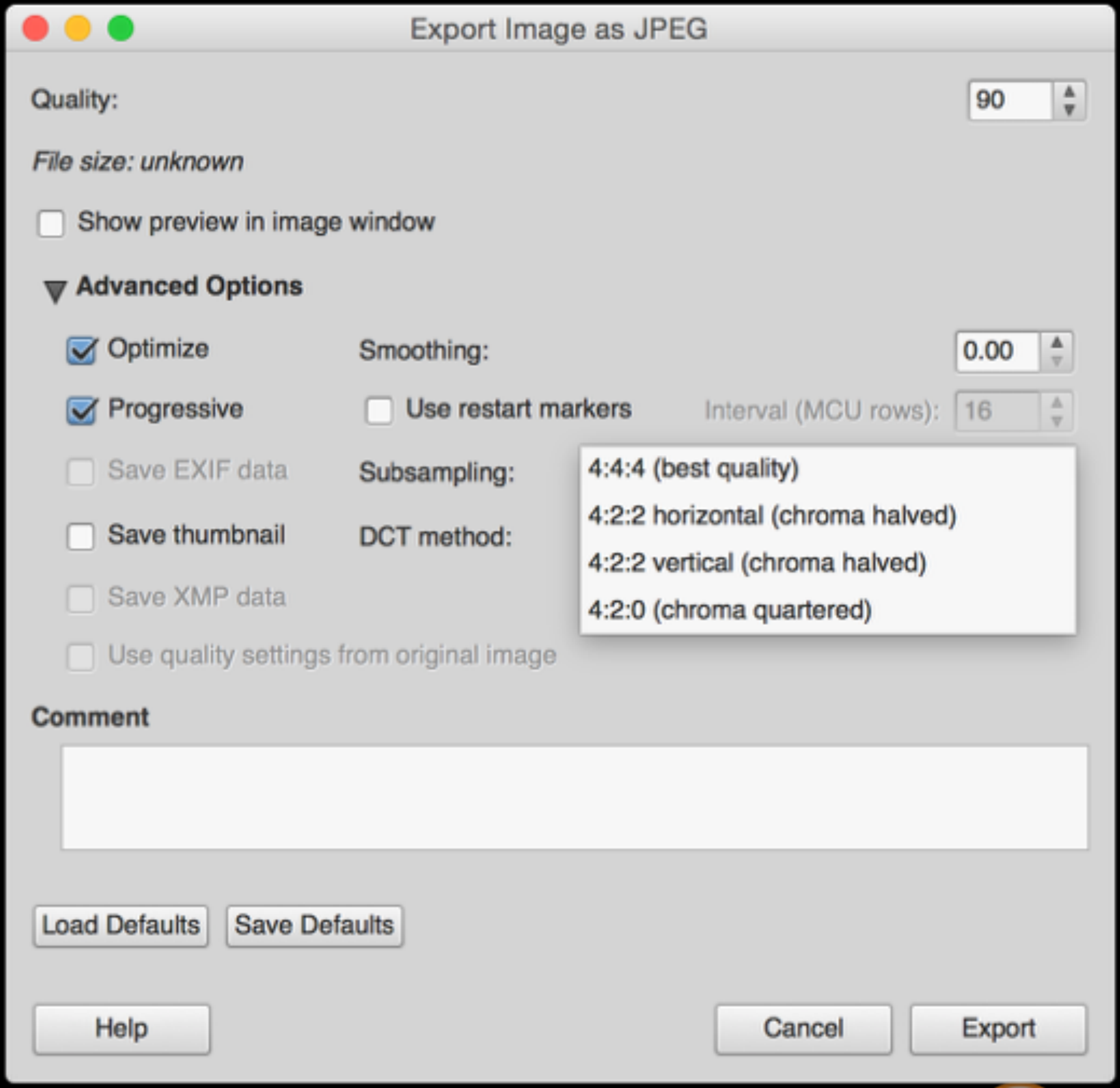
- Switch to Y’CBCR

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.29900 & 0.58700 & 0.11400 \\ -0.16874 & -0.33126 & 0.50000 \\ 0.50000 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

http://www.ams.org/samplings/feature-column/fcarc-image-compression

Chroma subsampling

- Possible to subsample each 8x8 block
- Huge saving in terms of chroma information



Export Image as JPEG

Quality: 90

File size: unknown

□ Show preview in image window

▼ **Advanced Options**

☑ Optimize — Smoothing: 0.00

☑ Progressive — □ Use restart markers — Interval (MCU rows): 16

□ Save EXIF data — Subsampling:

- 4:4:4 (best quality)
- 4:2:2 horizontal (chroma halved)
- 4:2:2 vertical (chroma halved)
- 4:2:0 (chroma quartered)

□ Save thumbnail — DCT method:

□ Save XMP data

□ Use quality settings from original image

Comment

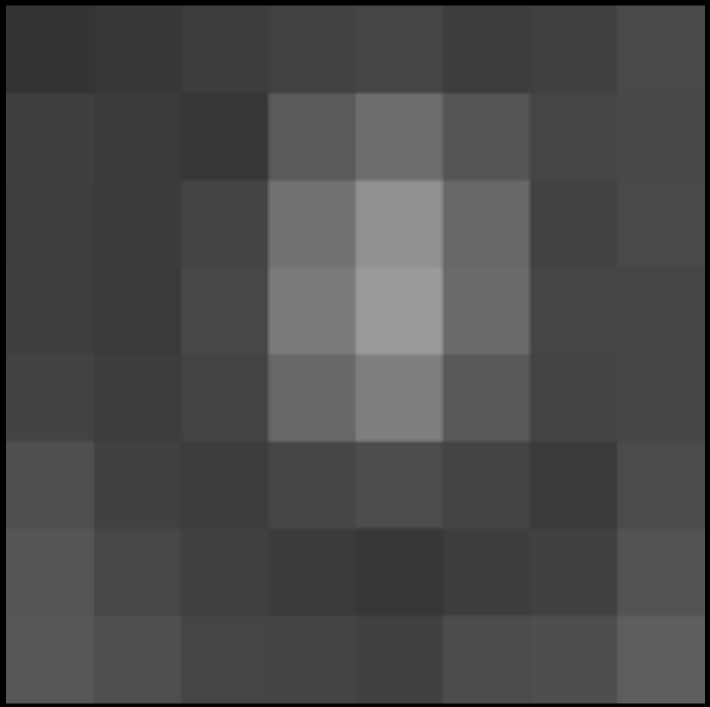
Discrete Cosine Transformation

- Humans are bad at detecting high-frequency changes in color/intensity information

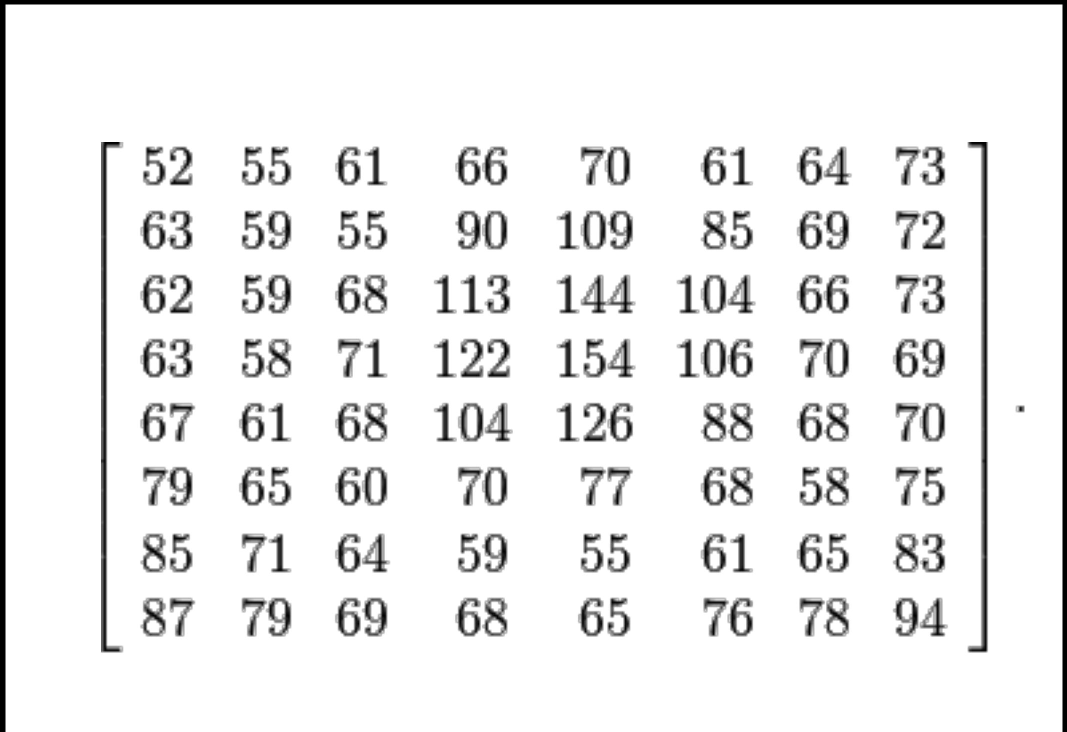
- But we are good at low frequency changes

- So… extract the frequencies from a block of 8x8 pixels and discard/reduce high frequency information

DCT



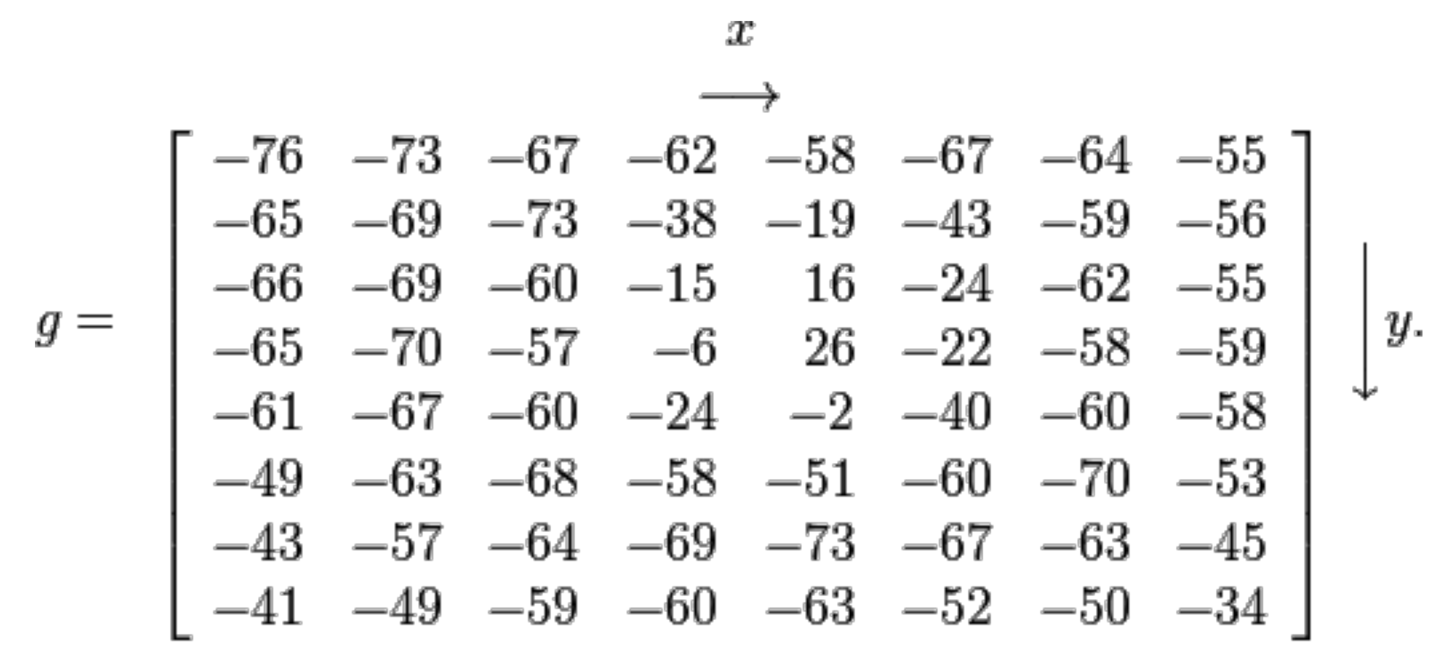
=



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

Centered on 0 by subtracting 128

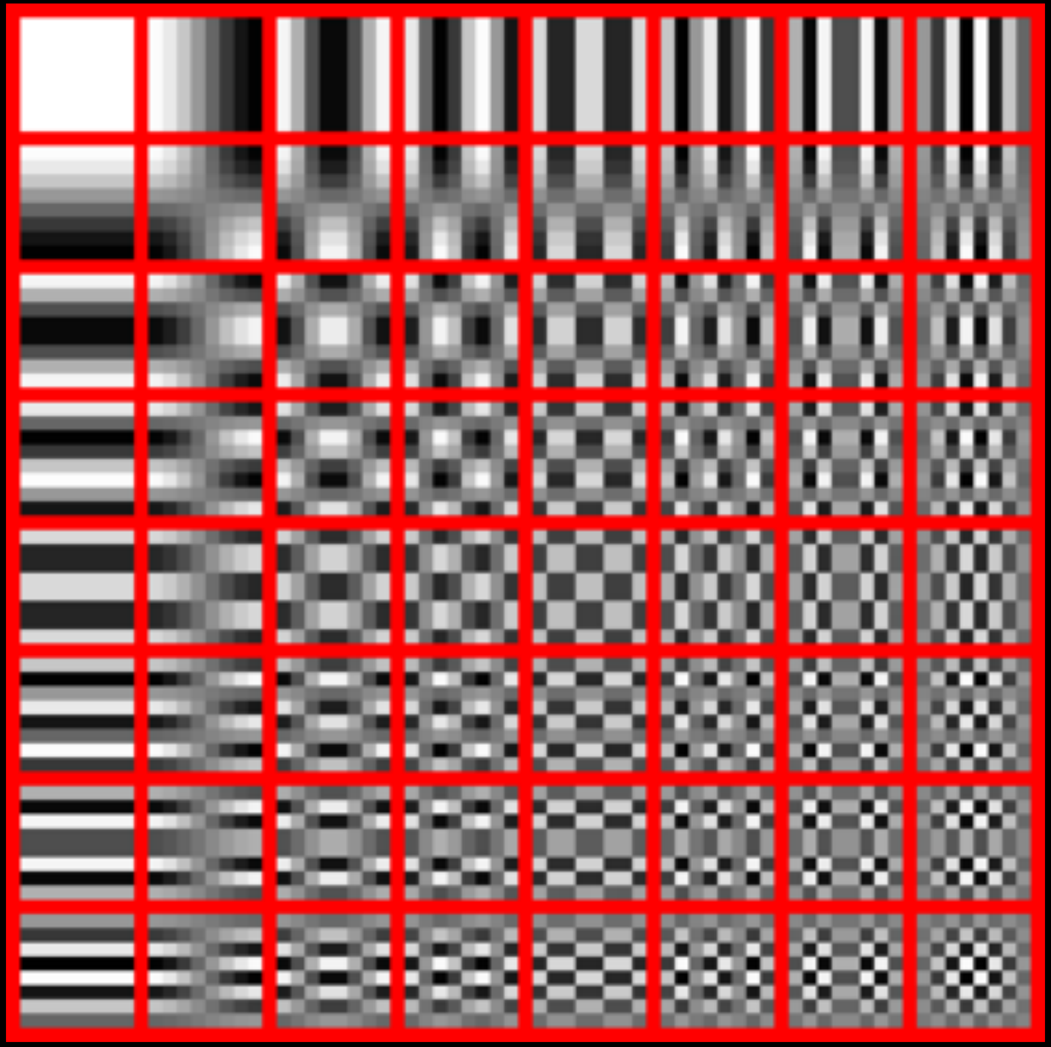
=



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

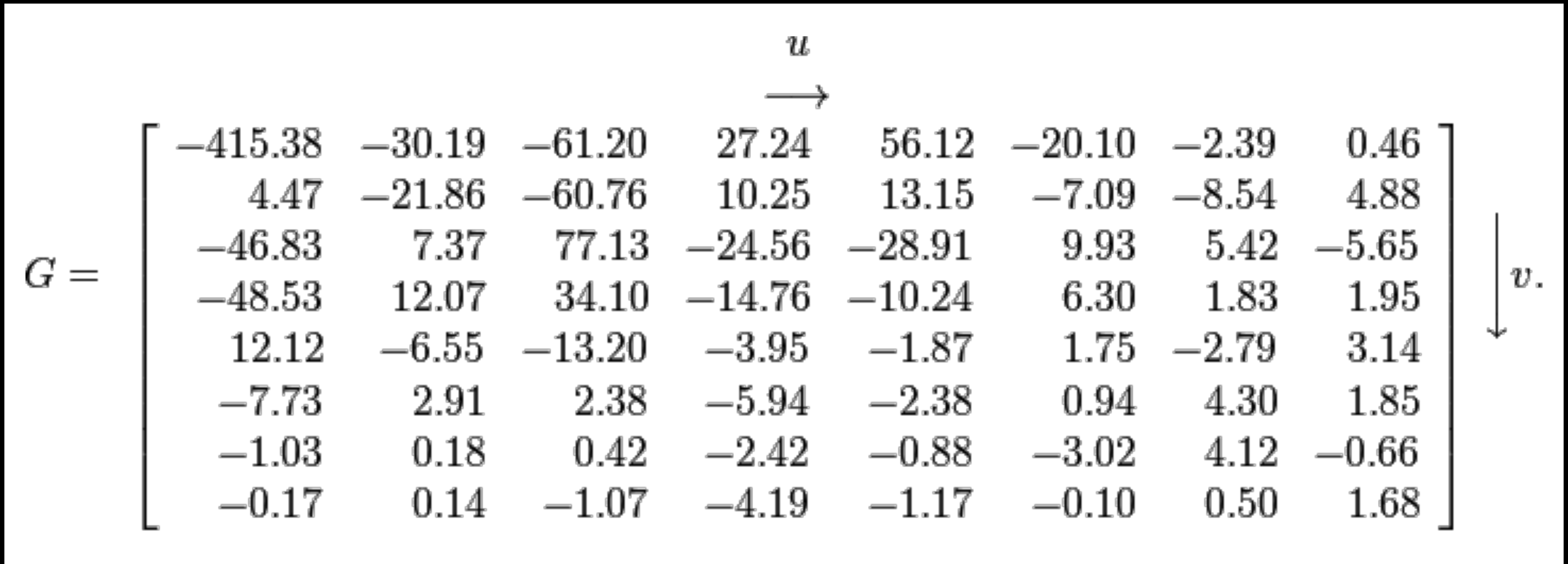
DCT

- Convert to 8x8 block of values based on summation of cosines of different frequencies



$$G_{u,v} = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y}\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

DCT

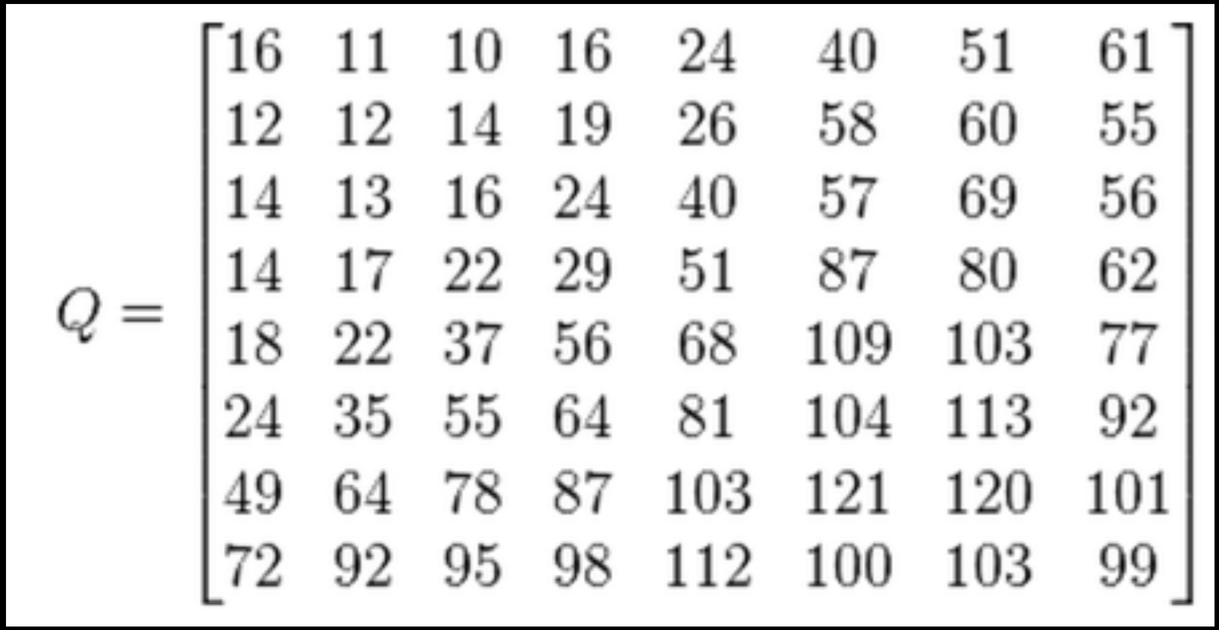


$$G = \begin{array}{c} \overset{u}{\longrightarrow} \\ \begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix} \end{array} \Big\downarrow v.$$

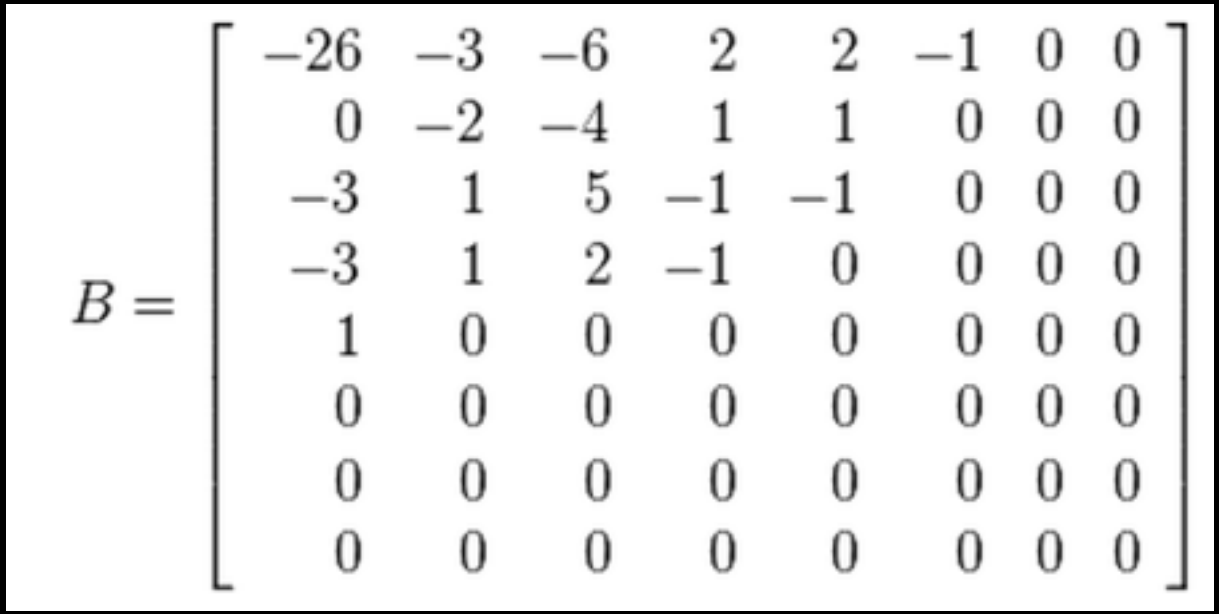
Low frequency components in top left corner

Quantization

- Throw away information (particularly at high frequencies)
- Divide each element of DCT matrix by value from a ‘quantization matrix’
- When you change JPEG quality it’s this step that changes

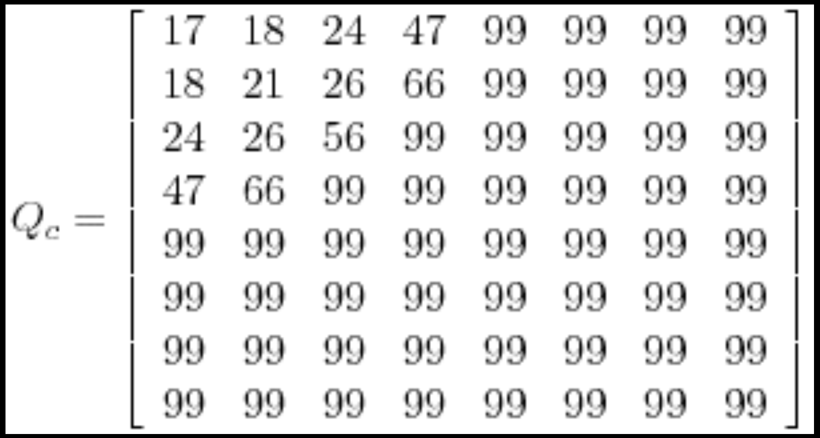


$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121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

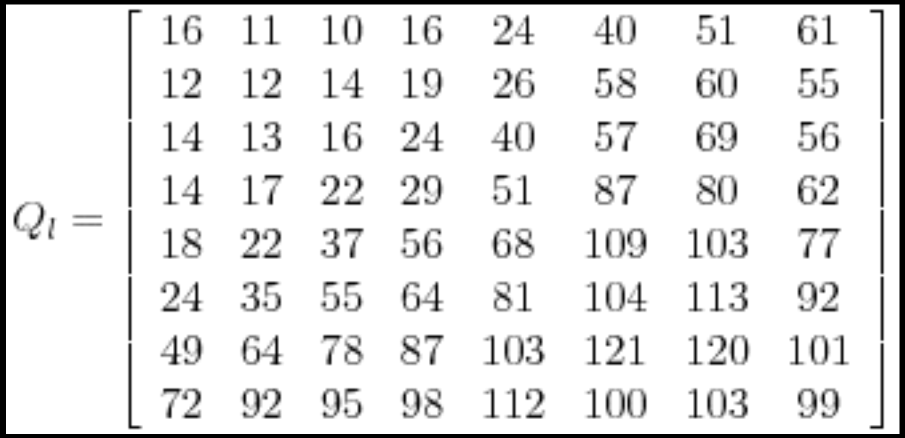


$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Different tables for luminance and chrominance

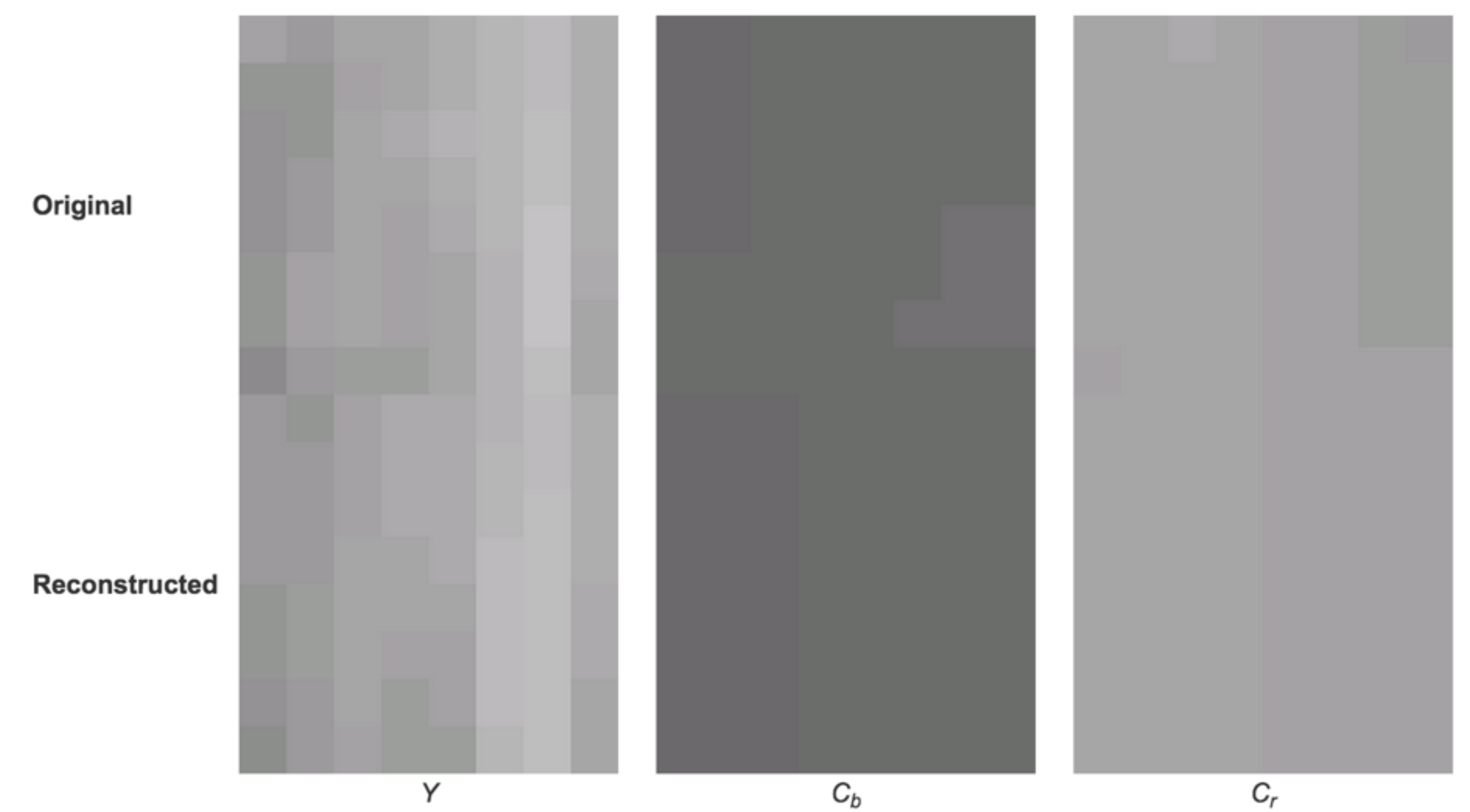


$$Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$



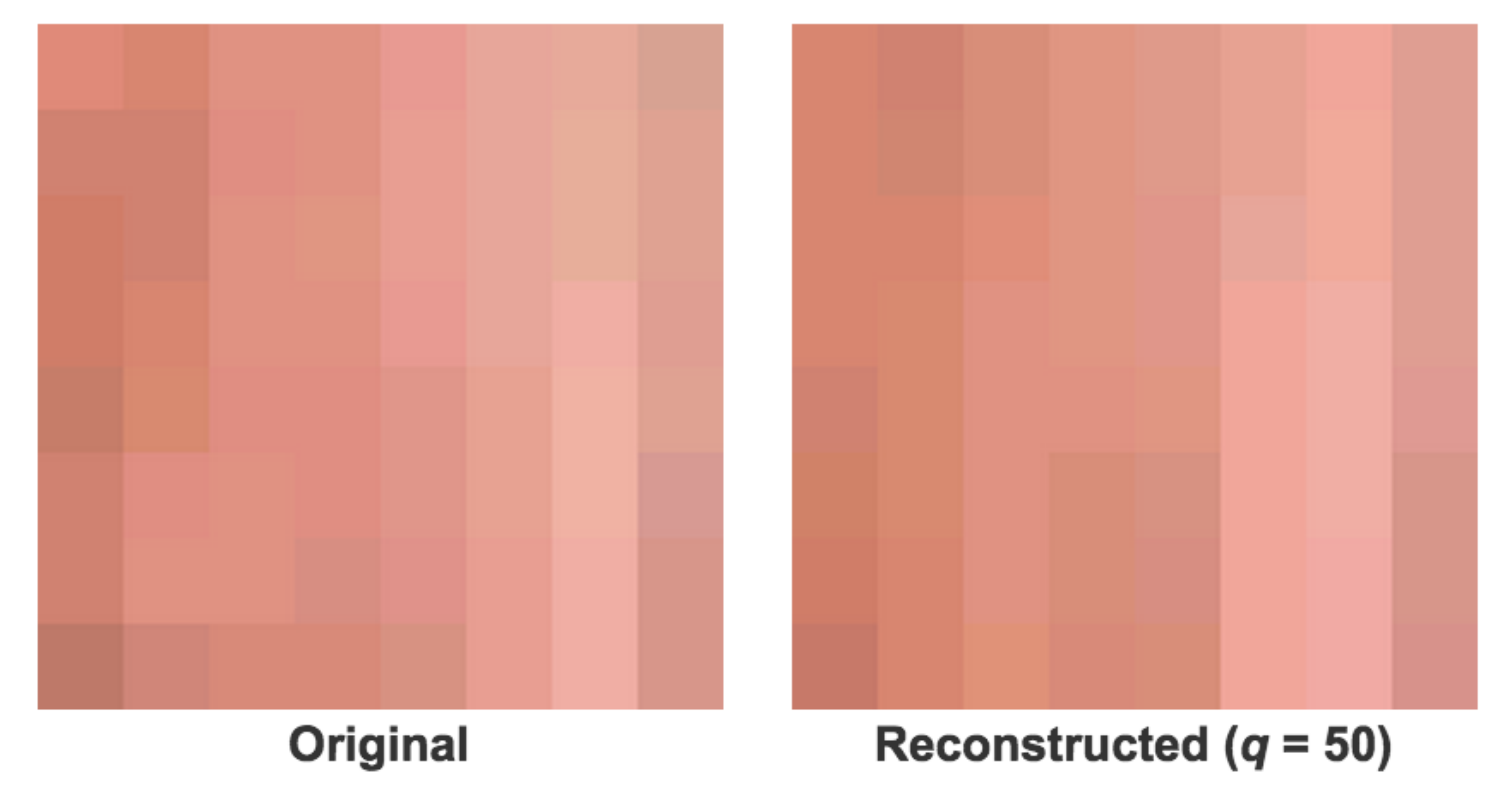
$$Q_l = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121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

Example



http://www.ams.org/samplings/feature-column/fcarc-image-compression

Example



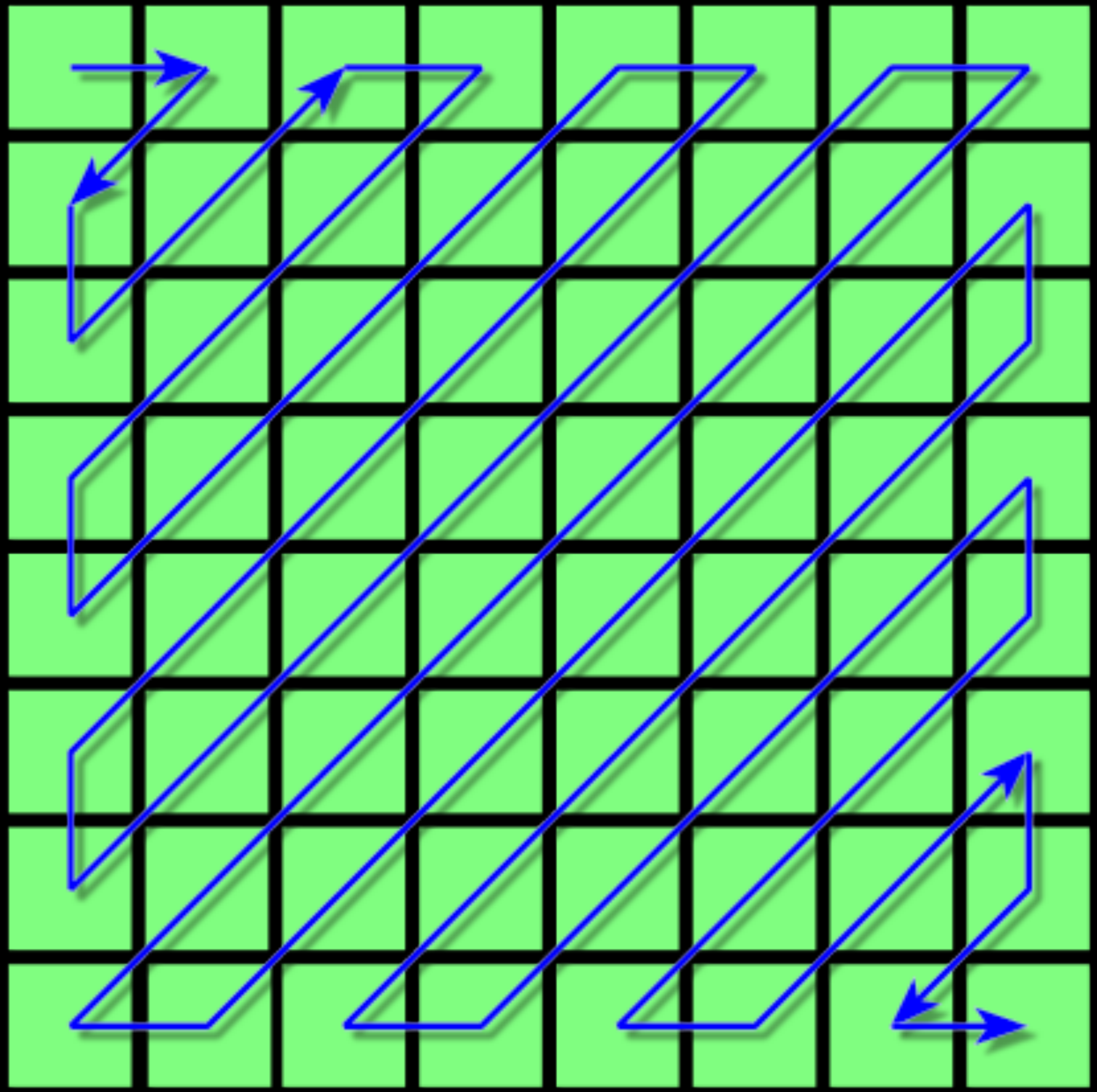
Original

Reconstructed (q = 50)

http://www.ams.org/samplings/feature-column/fcarc-image-compression

Entropy coding

- Read off the values from the resulting matrix in ‘zigzag’ order
- Essentially orders values by frequency
- Then run-length encode the result
- Then Huffman code
  - Can use standard Huffman tables or generate on the fly



JPEG Optimization

- Lots of opportunity here
  - How the image is chroma subsampled
  - How the image is quantized and by how much
  - The Huffman table used

JPEG Optimization

- jpegtran
  http://jpegclub.org/jpegtran/

  Optimizes the Huffman tree (-optimize)
  Can change the DCT coefficients

- mozjpeg
  https://github.com/mozilla/mozjpeg

  Uses progressive encoding for better Huffman performance
  Experimenting with Trellis quantization

Hackers wanted!

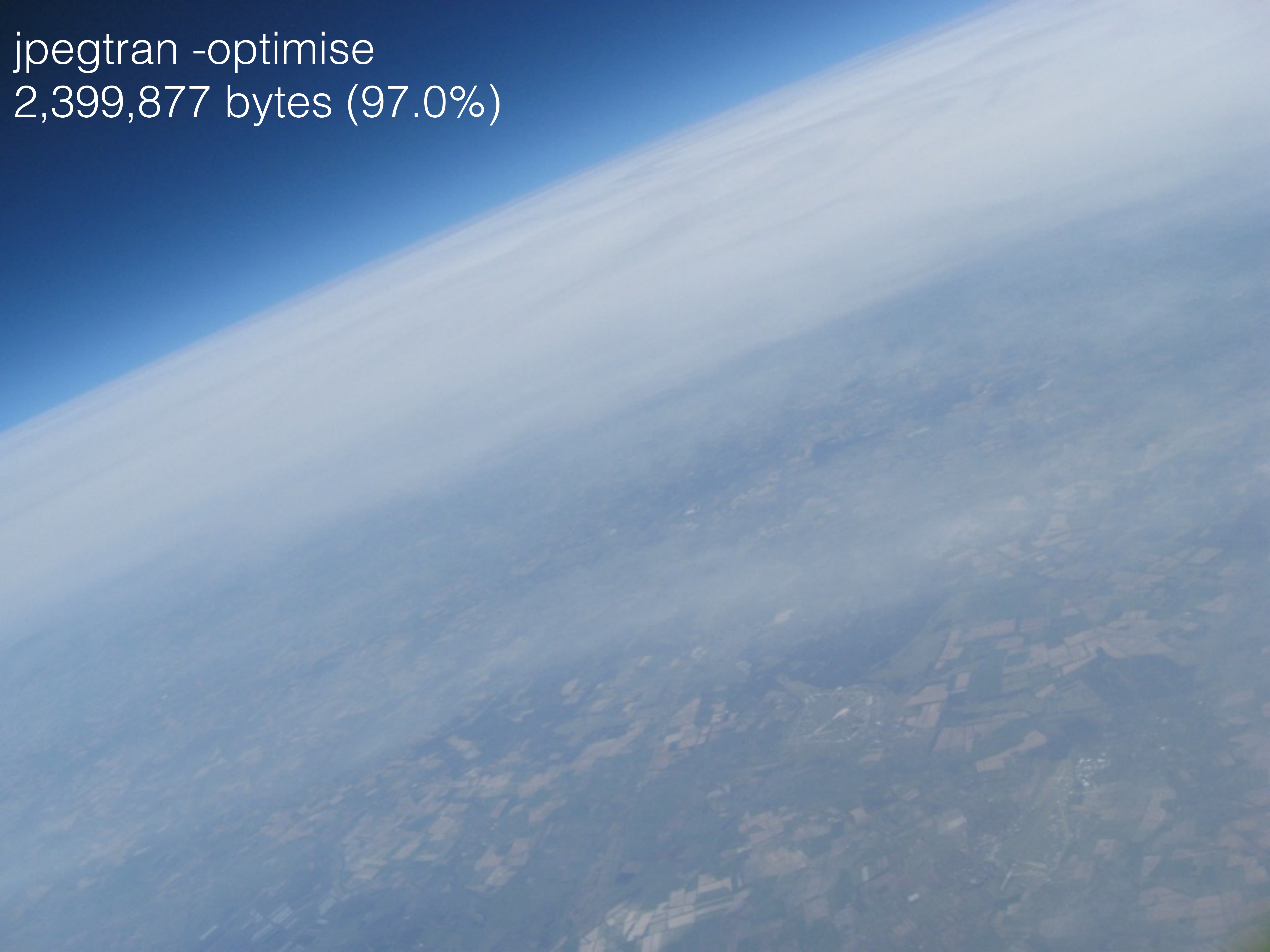
- Nice thing about JPEG format is… everything is in the JPEG file
  - So… invent your own quantization table
  - Come up with a better Huffman table
- Decoders should be able to decode

GIF vs. PNG vs. JPEG

- Only use GIF for animated images
- Use JPEG for photographs
- Use PNG for drawings with straight lines

Examples

2,472,447 bytes

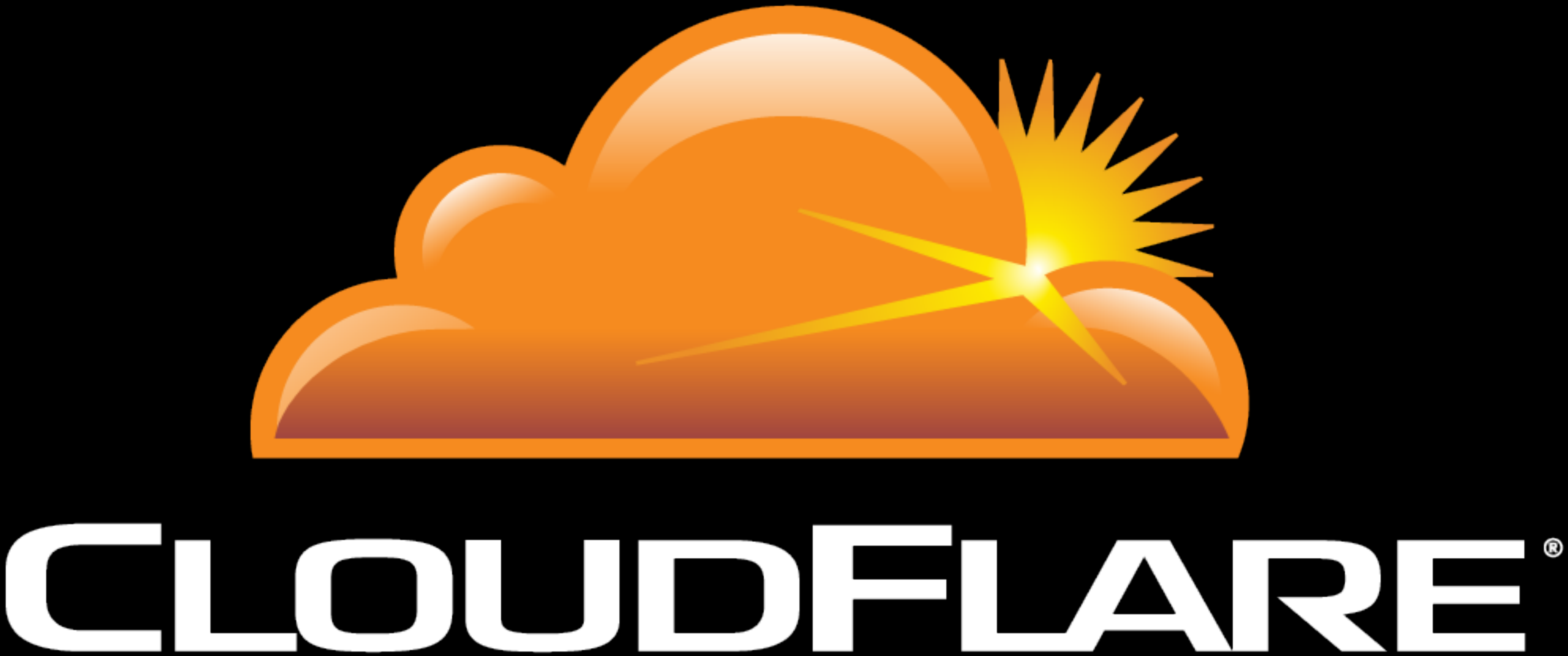


jpegtran -optimise
2,399,877 bytes (97.0%)

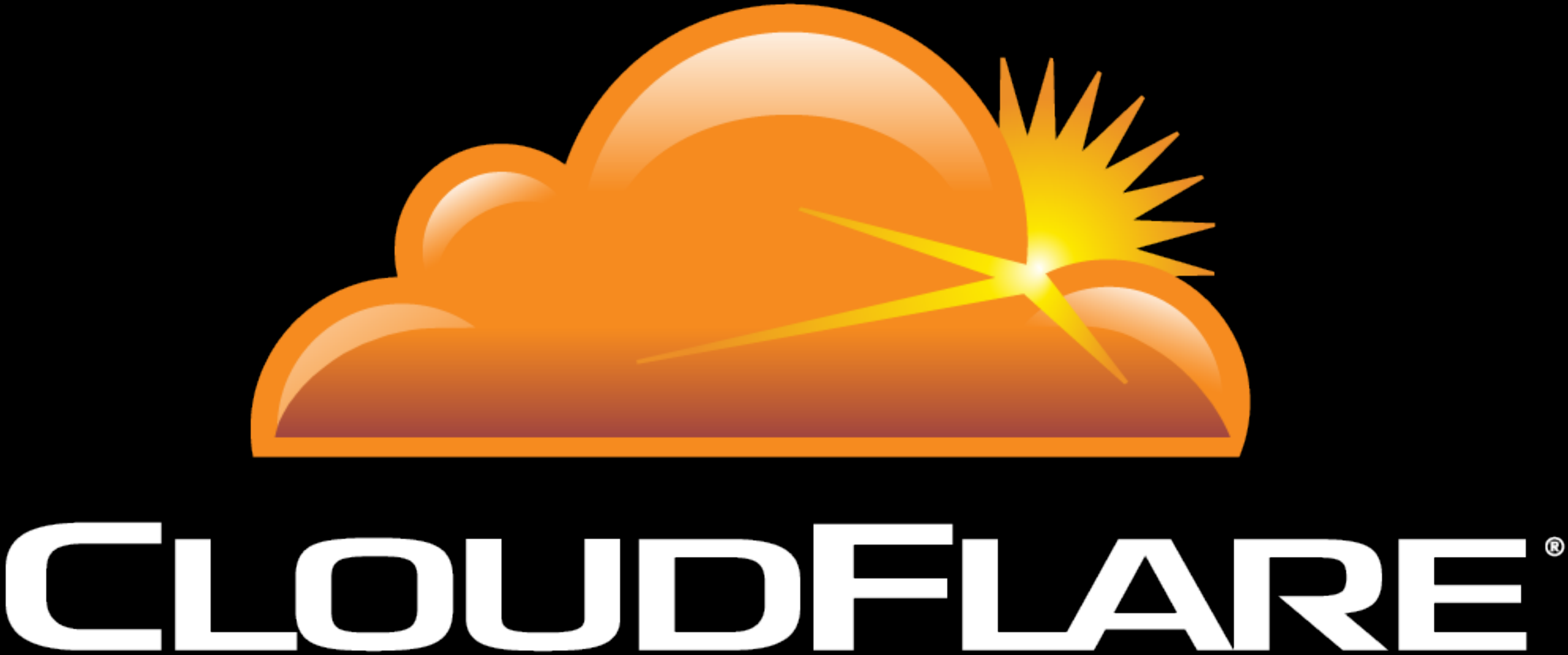
jpegtran -optimise -progressive

2,326,612 bytes (94.1%)

36,485 bytes

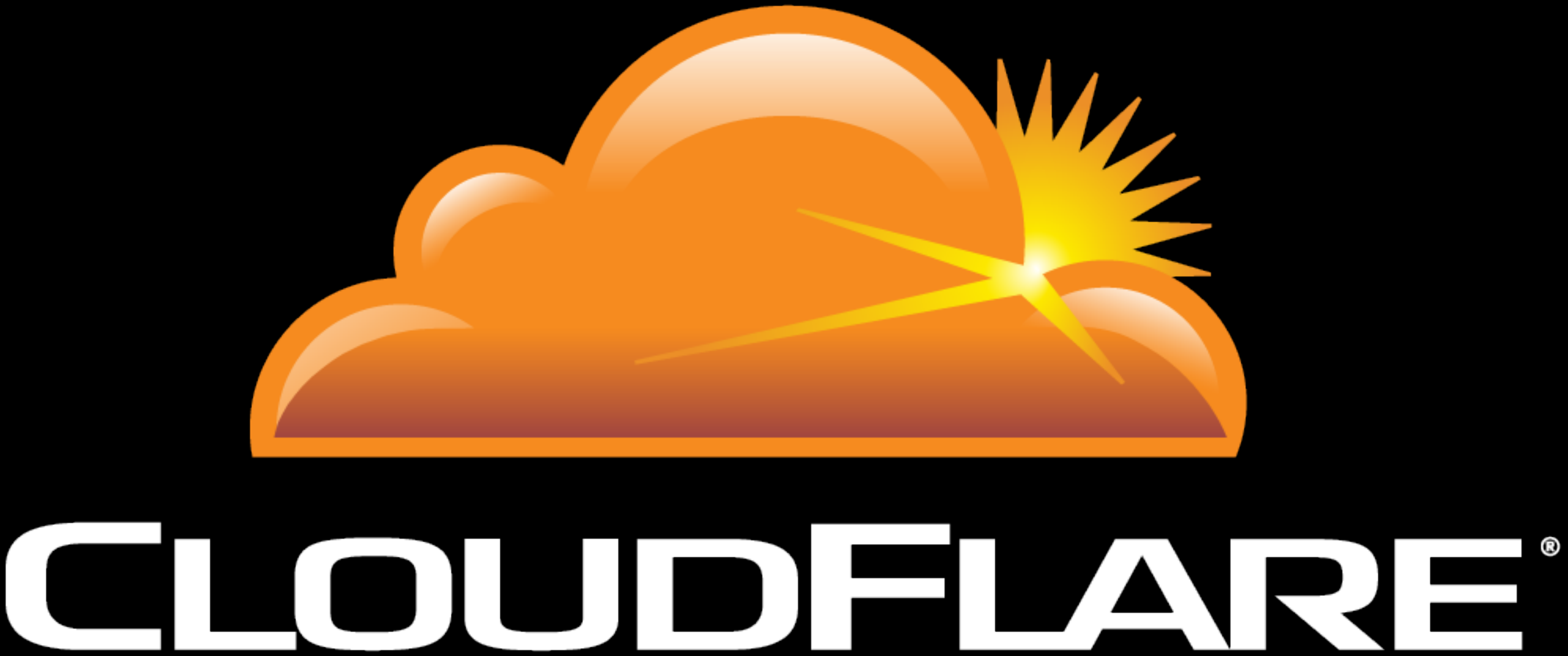


pngcrush -reduce
36,300 bytes (99.5%)



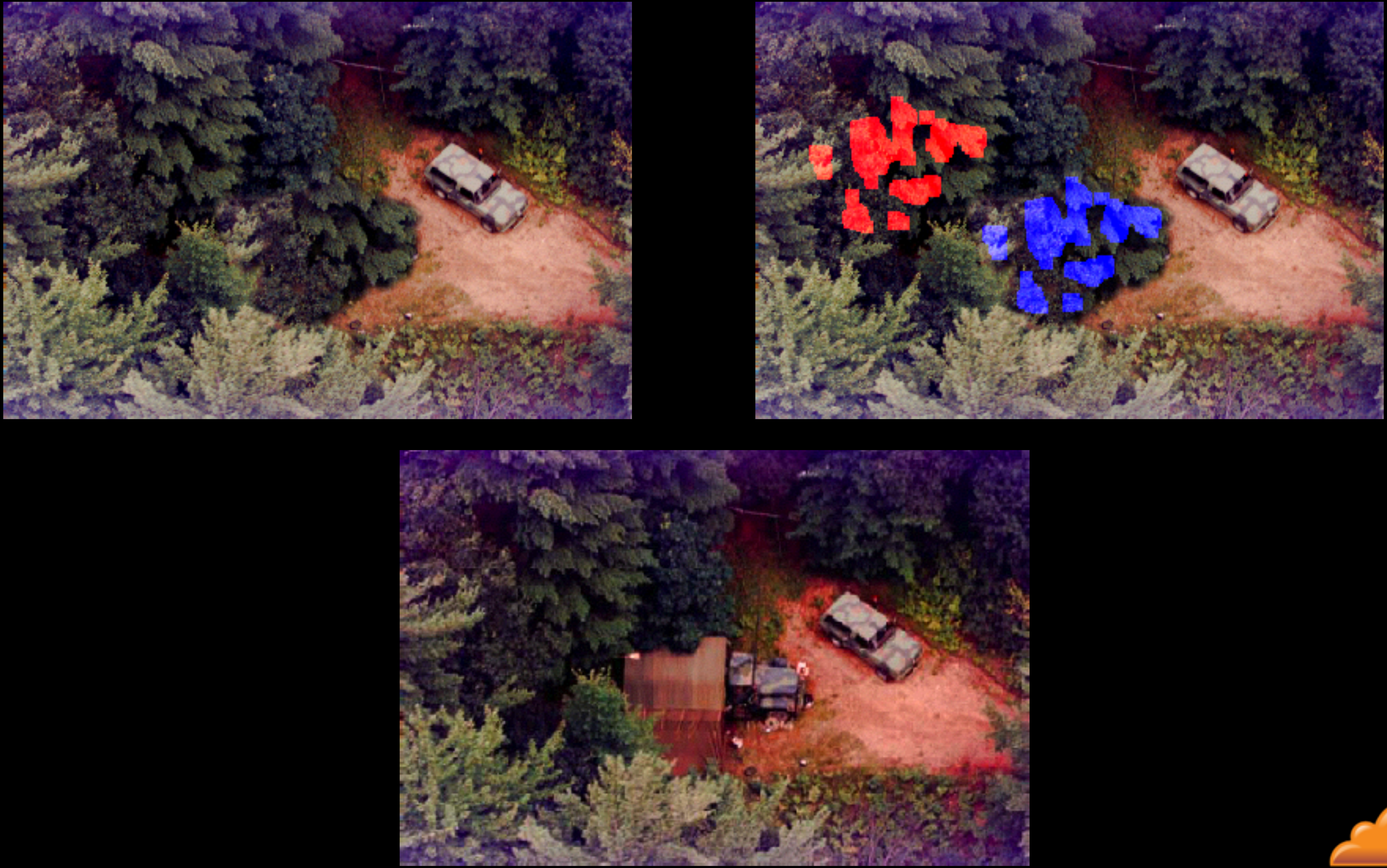
pngcrush -brute

35,010 bytes (96.0%)



Fun with JPEG Quantization

Automatically detecting copy/ paste image alteration



Used DCT and quantization to fuzzily find similar blocks

Details: http://blog.jgc.org/2008/02/tonight-im-going-to-write-myself-aston.html