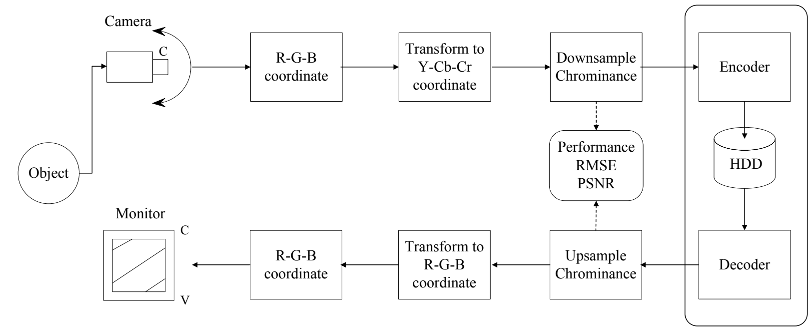
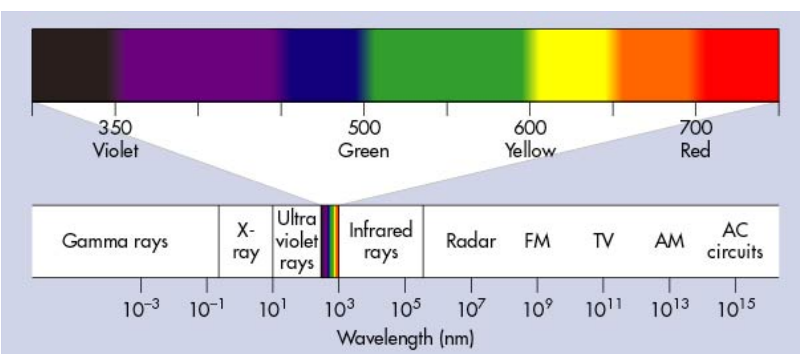
Color and Image Compression

General Image Storage System

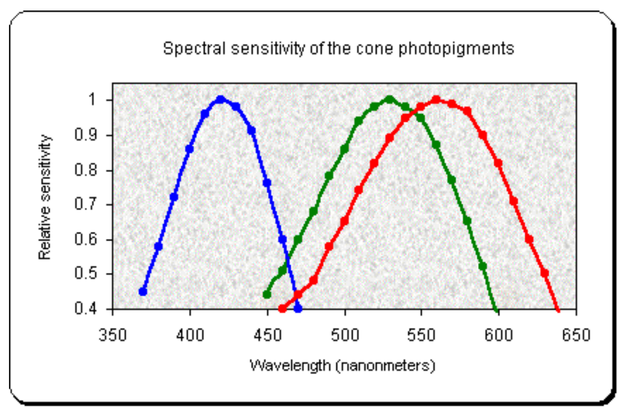


A beam of light separated into its Wavelengths



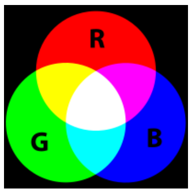
Human Color Perception

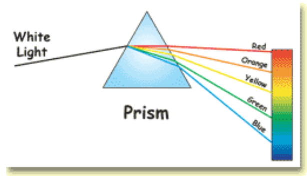
- Human reception is similar to RGB



Color

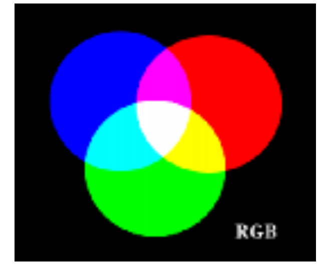
- Human eye has receptors for brightness (in low light) and separate receptors for red, green, and blue
- Can make any color we can see by mixing red, green and blue light in different intensities
- A color space is the set of all colors possible by mixing the basic colors.





RGB

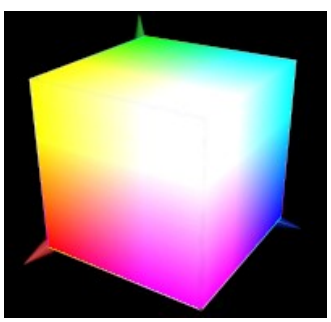
- Red-Green-Blue
  - Additive color model
  - Also known as the light model
  - Used in computer monitors

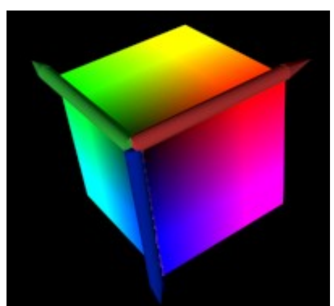


  - Can you see CMYK in the figure?

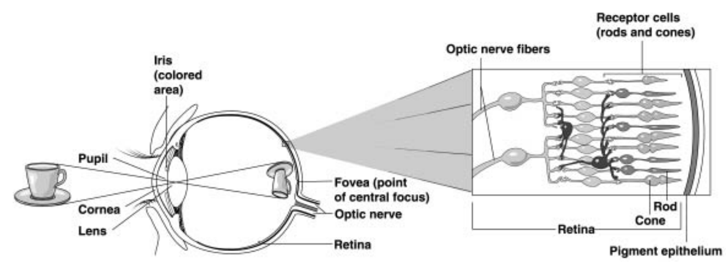
RGB Cube

- RGB values actually define a cube



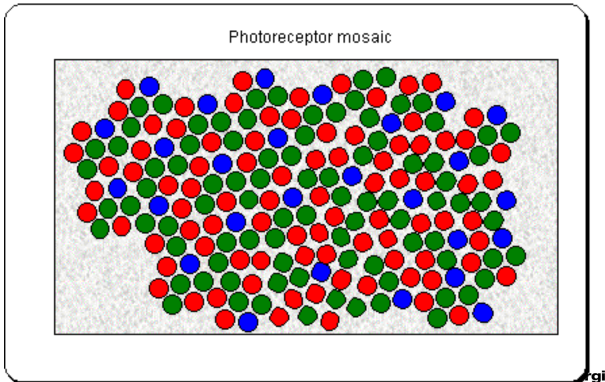


Eye cross-section

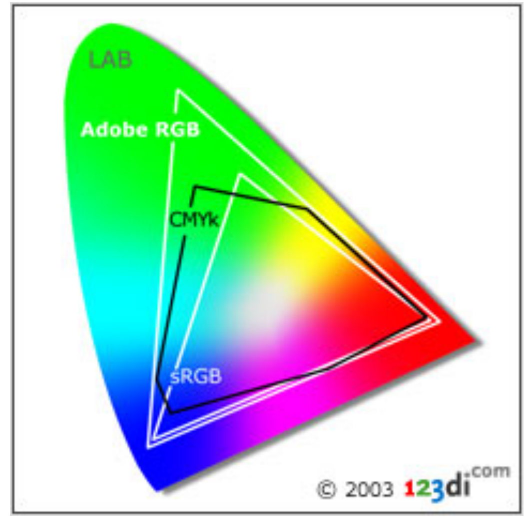


Cones

- Humans are most sensitive to Green, then Red.
  - A result of the distribution of cones



Color Spaces



HSV

- The Hue/Saturation/Value model was created by A. R. Smith in 1978.
  - It is based on such intuitive color characteristics as tint, shade and tone (or family, purety and intensity).
  - The coordinate system is cylindrical, and the colors are defined inside a hexcone.
    - The hue value H runs from 0 to 360º.
    - The saturation S is the degree of strength or purity and is from 0 to 1. Purity is how much white is added to the color, so S=1 makes the purest color (no white).
    - Brightness V also ranges from 0 to 1, where 0 is the black.
- See **http://www.cs.rit.edu/~ncs/color/t_convert.html** for conversion algorithm

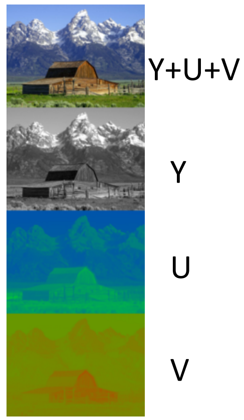
Color Specification

- **Luminance**
  - Received brightness of the light, which is proportional to the total energy in the visible band.
- **Chrominance**
  - Describe the perceived color tone of a light, which depends on the wavelength composition of light
  - Chrominance is in turn characterized by two attributes
    - **Hue**
      - Specify the color tone, which depends on the peak wavelength of the light
    - **Saturation**
      - Describe how pure the color is, which depends on the spread or bandwidth of the light spectrum

YUV Color Space

- The YUV color space describes color in terms of luminance and chrominance separately. This is often more efficient for processing and transmitting color signals
  - Y is the components of luminance
  - Cb and Cr are the components of chrominance
  - The values in the YUV coordinate are related to the values in the RGB coordinate by

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.334 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

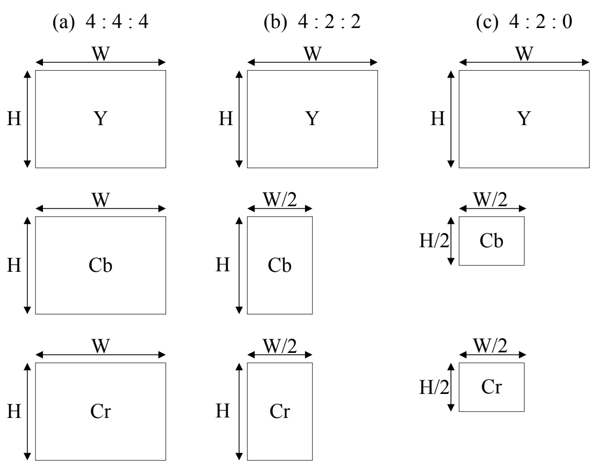


YUV Formats

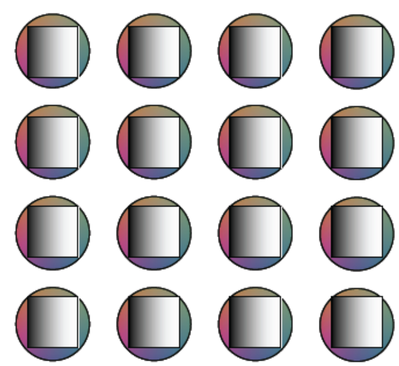
- YUV 4:4:4
  - 8 bits per Y,U,V channel (no chroma downsampling)
- YUV 4:2:2
  - 4Y pixels for every 2U and 2V
  - 2:1 horizontal downsampling, no vertical downsampling
- YUV 4:2:0
  - 2:1 horizontal downsampling
  - 2:1 vertical downsampling
- YUV 4:1:1
  - 4Y pixels for every 1 U and 1 V
  - 4:1 horizontal downsampling, no vertical downsampling

Spatial Sampling of Color Component

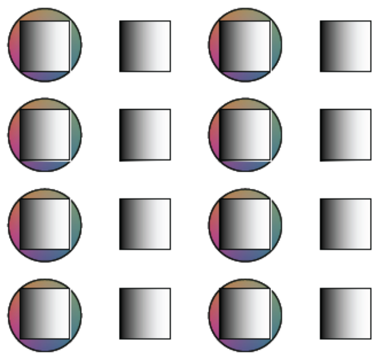
The three different chrominance down-sampling formats



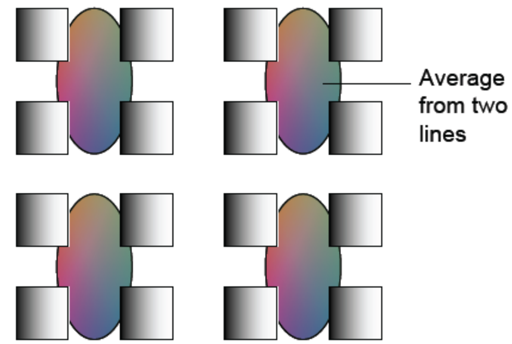
YUV 4:4:4 Sample Positions



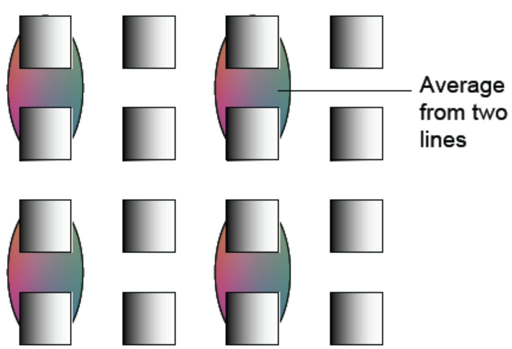
YUV 4:2:2 Sample Positions



YUV 4:2:0 (MPEG1/H.261/H.263/MPEG-4 AVC MainProfile)

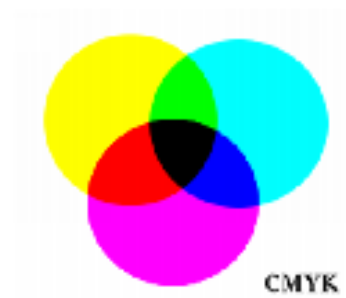


YUV 4:2:0 (MPEG-2, Progressive Source)



CMYK

- Cyan-Magenta-Yellow-(Black)
  - Subtractive color model
  - Black only needed in non-ideal situations
  - Used for print
  - Can you find a simple transformation for CMYK<->RGB?



RGB <-> CMYK

- Since ideally CMY is the complement of RGB, the following linear equations (known also as masking equations) were initially used to convert between RGB and CMY:

- C = 1 - R   R = 1 - C
- M = 1 - G   G = 1 - M
- Y = 1 - B   B = 1 - Y

- In reality, non-ideal displays, ink, & paper change this…

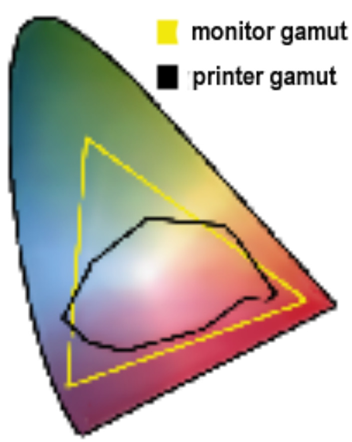
RGB <-> CMYK cont.

- Yellow ink typically provides a relatively pure yellow (it absorbs most of the blue light and reflects practically all the red and green light).
- Magenta ink typically does a good job of absorbing green light, but absorbs too much of the blue light (visually, this makes it too reddish). The extra redness in the magenta ink may be compensated for by a reduction of yellow in areas that contain magenta.
- Cyan ink absorbs most of the red light (as it should) but also much of the green light (which it should reflect, making cyan more bluish than it should be). The extra blue in cyan ink may be compensated for by a reduction of magenta in areas that contain cyan.
- All of these simple color adjustments, as well as the linear conversion from RGB to CMY, may be done using a 3 x 3 matrix multiplication:

- |C| |m1 m2 m3| |1 - R|
- |M| = |m4 m5 m6| |1 - G|
- |Y| |m7 m8 m9| |1 - B|

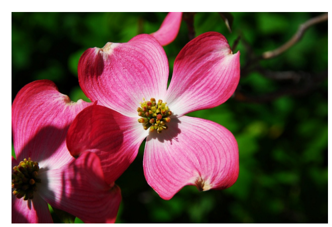
Color Gamut

- Not all colors can be represented using *positive values* in RGB or CMYK
- The printer gamut curve is dependent on many factors including
  - Inks
  - Paper
  - Print methods
- Special mapping is used



Black and White Conversion

- Our eye is about 70% sensitive to green, 20% sensitive to red, and 10% sensitive to blue (might be 60 30 for green and red).
- Picture below used for an example



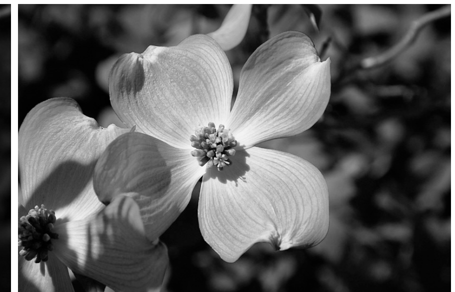
Black and White Conversion

.33*R + .33*G + .33* B

.2*R + .7*G + .1*B

Color Space Conversion

- There is a difference between Y in the YUV (or YCbCr) (left) and Lab for lightness (right).



Grey-level Image and Waveform

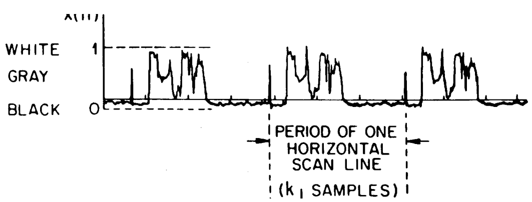


Image Compression

- The steps of image encoding
  1. Reduce the correlation between pixels
     - (We want to avoid sending essentially the same information over and over…)
  2. Quantization
  3. Source Coding

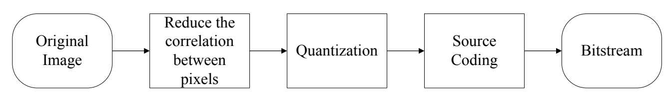


Image Compression

- Measures to evaluate the performance of image compression
  - Root Mean square error: $RMSE = \sqrt{\frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}\left[f(x,y)-f'(x,y)\right]^2}{WH}}$
  - Peak signal to noise ratio: $PSNR = 20\log_{10}\frac{255}{MSE}$
  - Compression Ratio: $Cr = \frac{n1}{n2}$

  Where n1 is the data rate of original image and n2 is that of the encoded bit-stream

Georgia Institute of Technology

Reduce the Correlation between Pixels

- **Orthogonal Transform Coding**
  - KLT (Karhunen-Loeve Transform)
    - Maximal Decorrelation Process
  - DCT (Discrete Cosine Transform)
    - JPEG is a DCT-based image compression standard, which is a lossy coding method and may result in some loss of details and unrecoverable distortion.
- **Subband Coding**
  - DWT (Discrete Wavelet Transform)
    - Various wavelets match images well.
    - JPEG 2000 is a 2-dimension DWT based image compression standard.
- **Predictive Coding**
  - DPCM
    - To remove mutual redundancy between successive pixels and encode only the new information

Georgia Institute of Technology

Karhunen-Loeve Transform

- KLT is the optimum transform coder
  - defined as the one that minimizes the mean square distortion of the reproduced data for a given number of total bits
- Unfortunately, the KLT is data-dependent!
  - New transform coefficients must be found for each data source
- For images, the DCT is close to the typical KLT

Georgia Institute of Technology

Discrete Cosine Transform

- Why DCT is more appropriate for image compression than DFT?
  - The DCT can concentrate the energy of the transformed signal in low frequency, whereas the DFT can not (it is more similar to the KLT)
  - For image compression, the DCT reduces blocking effects more than the DFT

Forward 2-D DCT

$$F(u,v) = \frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

for $u = 0,\ldots,N-1$ and $v = 0,\ldots,N-1$

where $N = 8$ and $C(k) = \begin{cases} 1/\sqrt{2} & \text{for } k = 0 \\ 1 & \text{otherwise} \end{cases}$

Inverse 2-D DCT

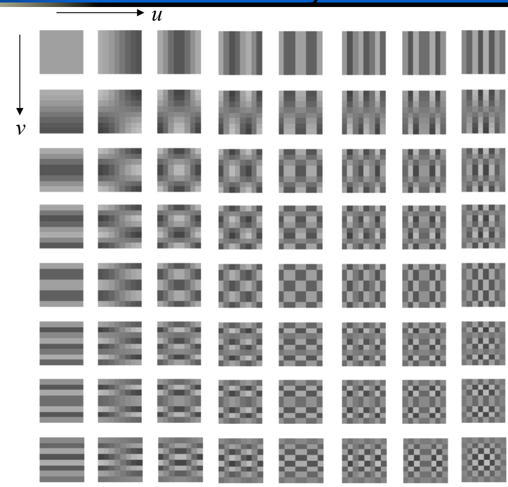
$$f(x,y) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

for $x = 0,\ldots,N-1$ and $y = 0,\ldots,N-1$ where $N = 8$

Georgia Institute of Technology

Discrete Cosine Transform

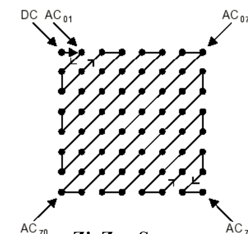
The 8-by-8 DCT basis



Differential Coding - JPEG

- Transform Coefficients
  - DC coefficient
  - AC coefficients
- Because there is usually strong correlation between the DC coefficients of adjacent 8×8 blocks, the quantized DC coefficient is encoded as the difference from the DC term of the previous block
- The other 63 entries are the AC components. They are treated separately from the DC coefficients in the entropy coding process

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

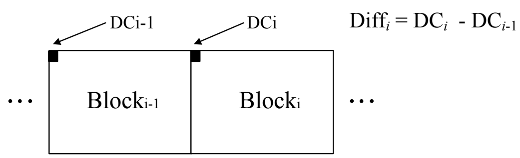


ZigZag Scan

Differential Coding - JPEG

- We set DC0 = 0.
- DC of the current block DCi will be equal to DCi-1 + Diffi .
- Therefore, in the JPEG file, the first coefficient is actually the difference of DCs. Then the difference is encoded with Huffman coding algorithm together with the encoding of AC coefficients

Differential Coding :



Zero-Run-Length Coding-JPEG

- The notation (L,F)
  - L zeros in front of the nonzero value F
- EOB (End of Block)
  - A special coded value means that the rest elements are all zeros
  - If the last element of the vector is not zero, then the EOB marker will not be added
- An Example:

1. ***57, 45, 0, 0, 0, 0, 23, 0, -30, -16, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ..., 0***
2. ***(0,57) ; (0,45) ; (4,23) ; (1,-30) ; (0,-16) ; (2,1) ; EOB***
3. ***(0,57) ; (0,45) ; (4,23) ; (1,-30) ; (0,-16) ; (2,1) ; (0,0)***
4. ***(0,6,111001);(0,6,101101);(4,5,10111);(1,5,00001);(0,4,0111);(2,1,1);(0,0)***
5. ***1111000 1111001 , 111000 101101 , 1111111110011000 10111 , 11111110110 00001 , 1011 0111 , 11100 1 , 1010***

Zero-Run-Length Coding-JPEG

Huffman table of Luminance AC coefficients

run/category	code length	code word
0/0 (EOB)	4	1010
15/0 (ZRL)	11	11111111001
0/1	2	00
...	...	...
0/6	7	1111000
...	...	...
0/10	16	1111111110000011
1/1	4	1100
1/2	5	11011
...	...	...
1/10	16	1111111110001000
2/1	5	11100
...	...	...
4/5	16	1111111110011000
...	...	...
15/10	16	1111111111111110

JPEG Steps

1. The representation of the colors in the image is converted from RGB to YCbCr, consisting of one luma component (Y), representing brightness, and two chroma components, (Cb and Cr), representing color. This step is sometimes skipped.

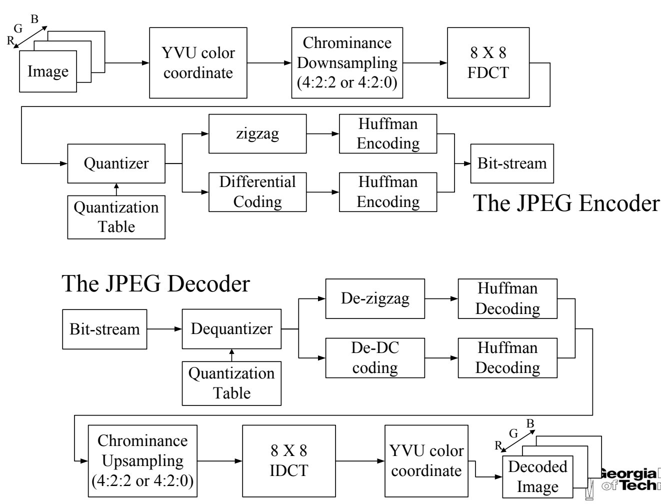
2. The resolution of the chroma data is reduced, usually by a factor of 2. This reflects the fact that the eye is less sensitive to fine color details than to fine brightness details.

3. The image is split into blocks of 8×8 pixels, and for each block, each of the Y, Cb, and Cr data undergoes a discrete cosine transform (DCT). A DCT is similar to a Fourier transform in the sense that it produces a kind of spatial frequency spectrum.

4. The amplitudes of the frequency components are quantized. Human vision is much more sensitive to small variations in color or brightness over large areas than to the strength of high-frequency brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components. The quality setting of the encoder (for example 50 or 95 on a scale of 0–100 in the Independent JPEG Group's library[4]) affects to what extent the resolution of each frequency component is reduced. If an excessively low quality setting is used, the high-frequency components are discarded altogether.

5. The resulting data for all 8×8 blocks is further compressed with a loss-less algorithm, a variant of Huffman encoding.

JPEG



Quantization in JPEG

- Quantization is the step where we actually throw away data.
- Luminance and Chrominance Quantization Table
  - lower numbers in the upper left direction
  - large numbers in the lower right direction
  - The performance is close to the optimal condition

Quantization $F(u,v)_{Quantization} = round\left(\frac{F(u,v)}{Q(u,v)}\right)$

Dequantization $F(u,v)_{deQ} = F(u,v)_{Quantization} \times Q(u,v)$

$$Q_Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad Q_C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Quantization of 8 x 8 Image

139.	144.	149.	153.	155.	155.	155.	155.
144.	151.	153.	156.	159.	156.	156.	156.
150.	155.	160.	163.	158.	156.	156.	156.
159.	161.	162.	160.	160.	159.	159.	159.
159.	160.	161.	162.	162.	155.	155.	155.
161.	161.	161.	161.	160.	157.	157.	157.
162.	162.	161.	163.	162.	157.	157.	157.
162.	162.	161.	161.	163.	158.	158.	158.

I

139.	145.	150.	154.	154.	153.	154.	153.
145.	15.0	154.	157.	157.	155.	156.	156.
150.	155.	158.	161.	160.	157.	157.	155.
159.	161.	161.	163.	161.	158.	159.	158.
159.	160.	161.	163.	161.	157.	156.	155.
163.	162.	160.	162.	161.	157.	157.	158.
162.	161.	159.	162.	161.	157.	157.	157.
164.	162.	160.	163.	162.	158.	159.	160.

QI

Quantization of 8 x 8 DCT

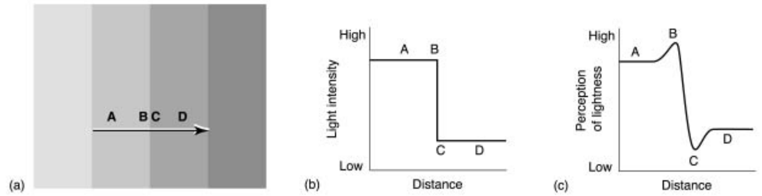
314.91	-0.26	-3.02	-1.30	0.53	-0.42	-0.68	0.33
-5.65	-4.37	-1.56	-0.79	-0.71	-0.02	0.11	-0.30
-2.74	-2.32	-0.39	0.38	0.05	-0.24	-0.14	-0.02
-1.77	-0.48	0.06	0.36	0.22	-0.02	-0.01	0.08
-0.16	-0.21	0.37	0.39	-0.03	-0.17	0.15	0.32
0.44	-0.05	0.41	-0.09	-0.19	0.37	0.26	-0.25
-0.32	-0.09	-0.08	-0.37	-0.12	0.43	0.27	-0.19
-0.65	0.39	-0.94	-0.46	0.47	0.30	-0.14	-0.11

DCT

315.00	0.00	-3.00	-1.00	1.00	0.00	-1.00	0.00
-6.00	-4.00	-2.00	-1.00	-1.00	0.00	0.00	0.00
-3.00	-2.00	0.00	0.00	0.00	0.00	0.00	0.00
-2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-1.00	0.00	-1.00	0.00	0.00	0.00	0.00	0.00

Q-DC

Mach Bands



JPEG: Image Quality and Coding Modes

Bpp	Quality	Sufficiency
0.25-0.5	moderate - good sufficient	some applications
0.5-0.75	good - very good sufficient	many applications
0.75-1.5	excellent sufficient	most applications
1.5-2.0	indistinguishable from original	most demanding applications

Coding Modes: *sequential, progressive, hierarchical,* and *lossless*