INTRODUCTION TO
DIGITAL IMAGE PROCESSING

Lecture #12

Niels Volkmann
Professor

ECE Department
Department of Bioengineering
Quantitative Bioscience Program

Image Compression

- Fundamentals: Coding redundancy
  - The gray level histogram of an image can reveal a great deal of information about the image
  - The probability of the occurrence of gray level r_k is $p(r_k)$,

$$p(r_k) = \frac{n_k}{n} \qquad k = 0,2,\ldots,L-1$$

Image Compression

- Fundamentals: Coding redundancy
  - If the number of bits used to represent each value of r_k is $l(r_k)$, the average number of bits required to represent each pixel is

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k)p(r_k)$$

  - To code an MxN image requires MNL_{avg} bits

Image Compression

- Fundamentals: Coding redundancy
  - some pixel values more common than others

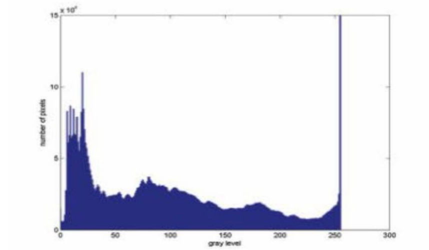


Image Compression

- Fundamentals: Coding redundancy
  - Variable length Coding: assign fewer bits to the more probable gray levels than to less probable ones can achieve data compression

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Image Compression

- Fundamentals: Interpixel (spatial) redundancy
  - neighboring pixels have similar values
  - Binary images
  - Gray scale image (later)

Image Compression

- Fundamentals: Interpixel (spatial) redundancy: Binary images
  - Run-length coding
    - Mapping the pixels along each scan line into a sequence of pairs $(g_1, r_1), (g_2, r_2), \ldots$,
    - Where g_i is the i^{th} gray level, r_i is the run length of i^{th} run

Image Compression

- Example: Run-length coding

Row 1: (0, 16)
Row 2: (0, 16)
Row 3: (0, 7) (1, 2) (0, 7)
Row 4: (0, 4), (1, 8) (0, 4)
Row 5: (0, 3) (1, 2) (0, 6) (1, 3) (0, 2)
Row 6: (0,2) (1, 2) (0,8) (1, 2) (0, 2)
Row 7: (0, 2) (1,1) (0, 10) (1,1) (0, 2)
Row 8: (1, 3) (0, 10) (1,3)
Row 9: (1, 3) (0, 10) (1, 3)
Row 10: (0,2) (1, 1) (0,10) (1, 1) (0, 2)
Row 11: (0, 2) (1, 2) (0, 8) (1, 2) (0, 2)
Row 12: (0, 3) (1, 2) (0, 6) (1, 3) (0, 2)
Row 13: (0, 4) (1,8) (0, 4)
Row 14: (0, 7) (1, 2) (0, 7)
Row 15: (0, 16)
Row 16: (0, 16)

encode

decode

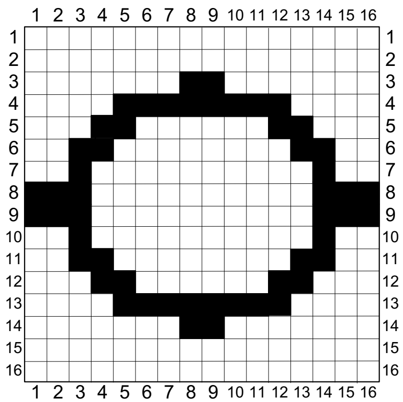


Image Compression

- Fundamentals: Psychovisual redundancy
  - some color differences are imperceptible to the human eye

Image Compression

- Fidelity criteria
  - Root mean square error (e_{rms}) and signal-to-noise ratio (SNR):
  - Let $f(x,y)$ be the input image, $f'(x,y)$ be reconstructed input image from compressed bit stream, then

$$e_{rms} = \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(f'(x,y) - f(x,y) \right)^2 \right)^{1/2} \qquad SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(f'(x,y) \right)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(f'(x,y) - f(x,y) \right)^2}$$

Image Compression

- Fidelity criteria
  - e_{rms} and SNR are convenient objective measures
  - Most decompressed images are viewed by human beings
  - Subjective evaluation of compressed image quality by human observers is often more appropriate

Image Compression

- Fidelity criteria

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Image Compression

Image compression models

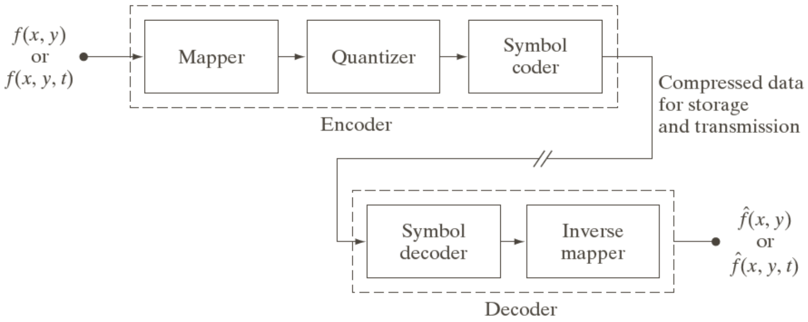


Image Compression

Exploiting Coding Redundancy

- These methods, from information theory, are not limited to images, but apply to any digital information. So we speak of "symbols" instead of "pixel values" and "sources" instead of "images".
- The idea: instead of natural binary code, where each symbol is encoded with a fixed-length code word, exploit nonuniform probabilities of symbols (nonuniform histogram) and use a variable-length code.

Image Compression

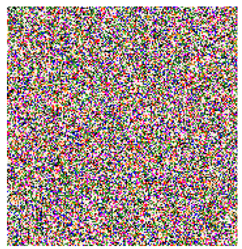
Exploiting Coding Redundancy

- Entropy $$H = -\sum_{k=0}^{L-1} p(r_k)\log_2 p(r_k)$$
- is a measure of the information content of a source.
- If source is an independent random variable then you can't compress to fewer than *H* bits per symbol.
- Assign the more frequent symbols short bit strings and the less frequent symbols longer bit strings. Best compression when redundancy is high (entropy is low, histogram is highly skewed).

Image Compression

Exploiting Coding Redundancy

Low entropy



High entropy

Image Compression

- Exploiting Coding Redundancy
- Huffman Coding
  - Codebook is precomputed and static.
  - Compute probabilities of each symbol by histogramming source.
  - Process probabilities to precompute codebook: code(i).
  - Encode source symbol-by-symbol: symbol(i) -> code(i).
  - The need to preprocess the source before encoding begins is a disadvantage of Huffman coding

Image Compression

- Exploiting Coding Redundancy
- Huffman Coding

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Image Compression

- Exploiting Coding Redundancy
- Huffman Coding

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Image Compression

- Exploiting Coding Redundancy
- Huffman Coding

- Average length of the code is 2.2.bits/symbol
- The entropy of the source is 2.14 bits/symbol

Image Compression

- Exploiting Spatial/Interpixel Redundancy
- Extension of run-length Coding
- Predictive Coding
  - Image pixels are highly correlated (dependent)
  - Predict the image pixels to be coded from those already coded

Image Compression

- Exploiting Spatial/Interpixel Redundancy
- Extension of run-length Coding
- Predictive Coding

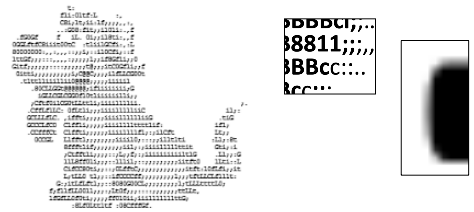


Image Compression

- Exploiting Spatial/Interpixel Redundancy
- Predictive Coding
  - Differential Pulse-Code Modulation (DPCM)
  - Simplest form: code the difference between pixels

Original pixels:
82, 83, 86, 88, 56, 55, 56, 60, 58, 55, 50, ……

→

DPCM:
82, 1, 3, 2, -32, -1, 1, 4, -2, -3, -5, ……

Image Compression

- Exploiting Spatial/Interpixel Redundancy
- Predictive Coding
  - Key features: Invertible, and lower entropy

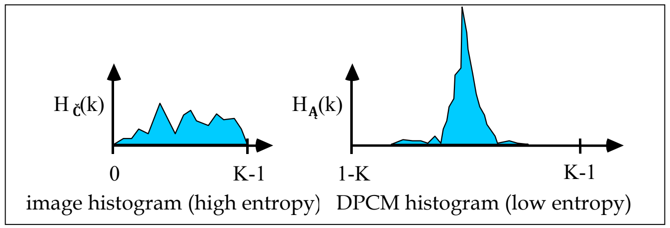


Image Compression

- Exploiting Spatial/Interpixel Redundancy
  - **Higher Order (Pattern) Prediction**
    - Use both 1D and 2D patterns for prediction

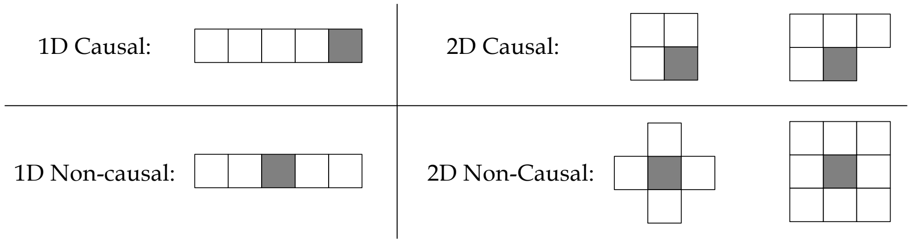


Image Compression

- Quantization
  - **Widely Used in Lossy Compression**
    - Represent certain image components with fewer bits (compression)
    - With unavoidable distortions (lossy)
  - **Quantizer Design**
    - Find the best tradeoff between
      maximal compression ←→ minimal distortion

Image Compression

- Quantization
  - Scalar quantization

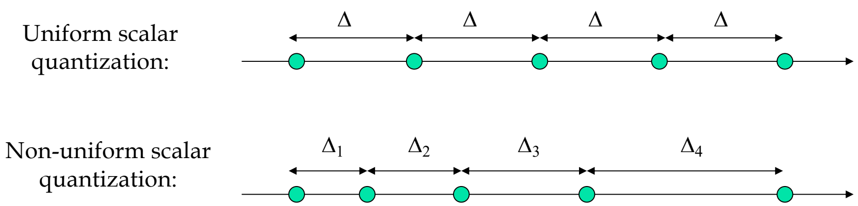
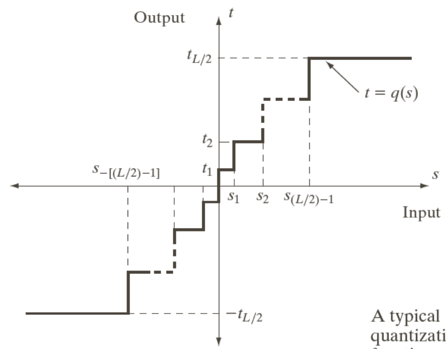


Image Compression

- Quantization
  - Scalar quantization



A typical quantization function.

Image Compression

- Quantization
- Palletizing color images
  - A true color image – 24bits/pixel, R – 8 bits, G – 8 bits, B – 8 bits

Exploits Psychovisual Redundancies

1,677,216 possible colors

256 possible colors

  - Palletized gif images - 8bits/pixel

Image Compression

- Palletized color image (gif)

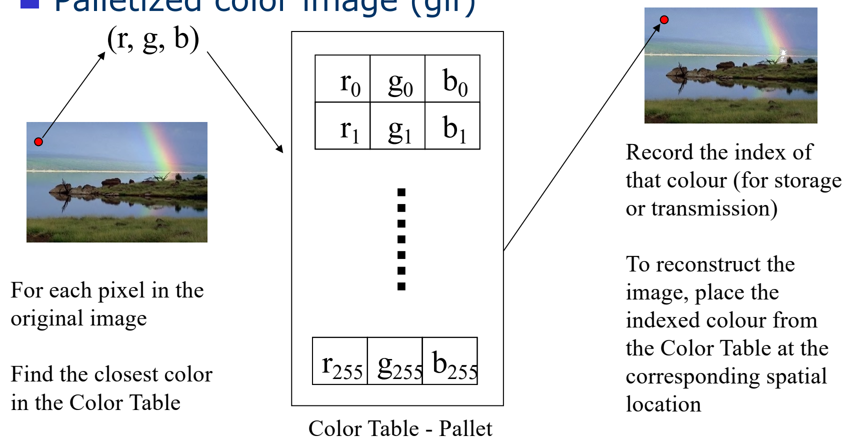


Image Compression

- Construct the pallet (vector quantization, k-means algorithm)

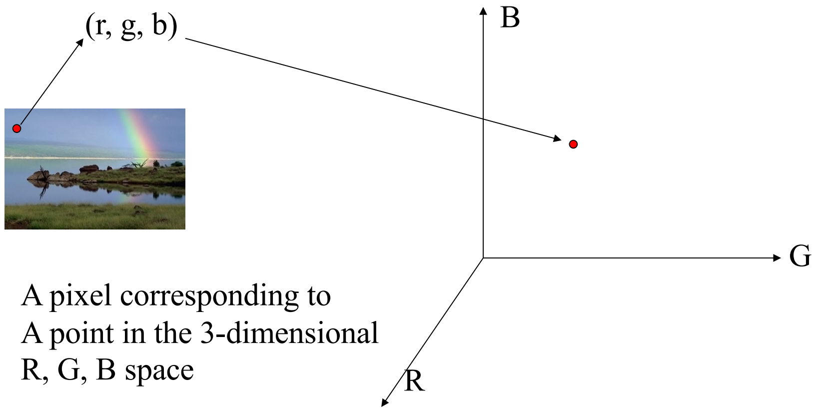


Image Compression

- vector quantization

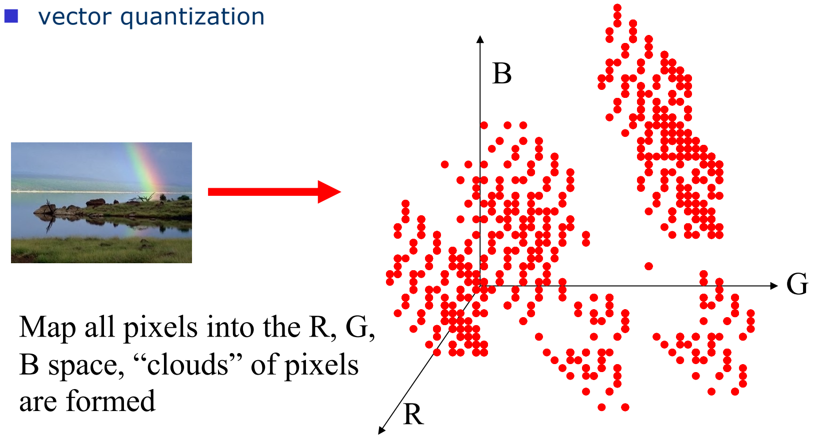


Image Compression

- K-means algorithm

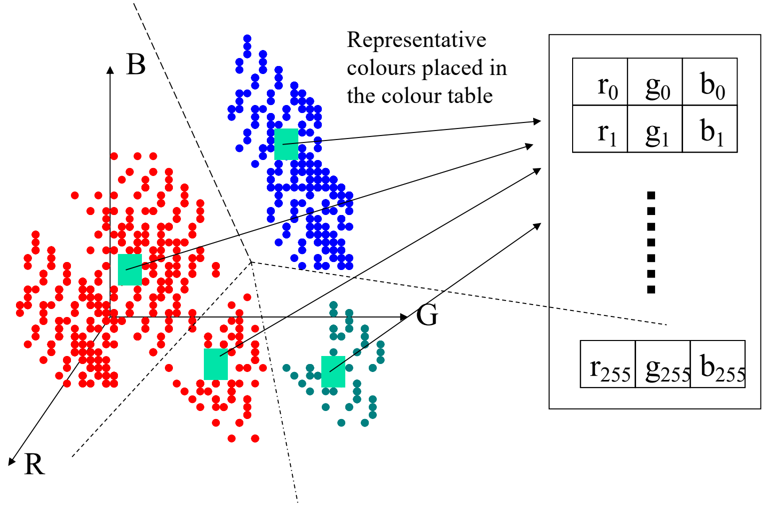


Image Compression

- Discrete Cosine Transform (DCT)
  - A DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT) but using ***only real numbers***.
  - DCTs are generally related to Fourier series coefficients of a ***periodically and symmetrically*** extended sequence
  - DFTs are related to Fourier series coefficients of only ***periodically*** extended sequences.

Image Compression

- Discrete Cosine Transform (DCT)
  - 2D-DCT

$$X(u,v)=\frac{4C(u)C(v)}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}x(m,n)\cos\left(\frac{(2m+1)u\pi}{2N}\right)\cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

  - Inverse 2D-DCT

$$x(m,n)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)X(u,v)\cos\left(\frac{(2m+1)u\pi}{2N}\right)\cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

$$\text{where}\quad C(u)=\begin{cases}\frac{1}{\sqrt{2}} & u=0\\ 1 & u=1,\cdots,N-1\end{cases}$$

Image Compression

$$\begin{bmatrix}
55 & 36 & -34 & 25 & 66 & 35 & 4 & 37\\
55 & 25 & -12 & 48 & 59 & 38 & 2 & 41\\
51 & 40 & 43 & 54 & 51 & 42 & 3 & 39\\
49 & 49 & 51 & 49 & 51 & 37 & 3 & 39\\
50 & 50 & 51 & 48 & 54 & 36 & 2 & 43\\
51 & 52 & 52 & 51 & 55 & 41 & 4 & 41\\
51 & 51 & 52 & 54 & 55 & 42 & 1 & 45\\
52 & 51 & 53 & 51 & 53 & 42 & 2 & 41
\end{bmatrix}$$

image block

2D-DCT

DC component

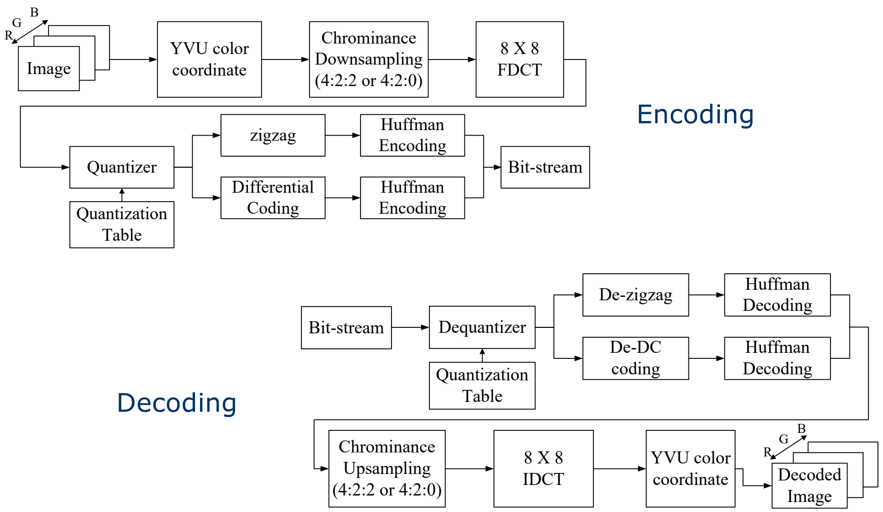
low frequency ⟶ high frequency

low frequency ↓ high frequency

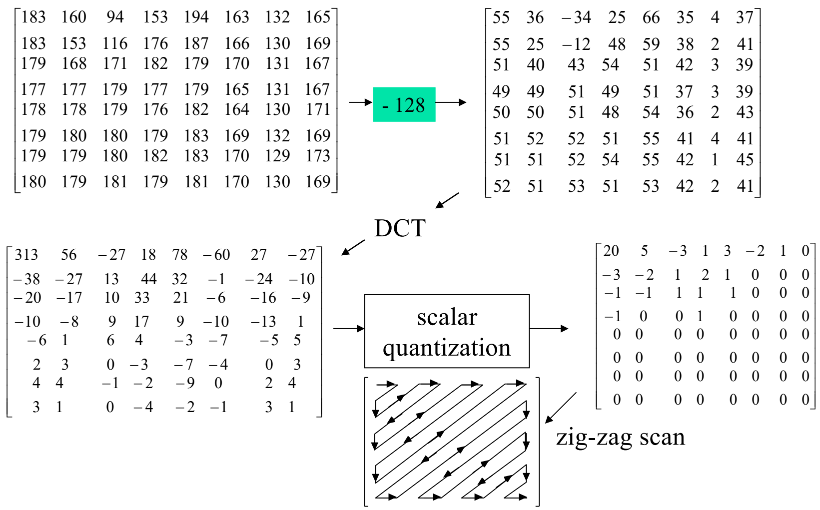
$$\begin{bmatrix}
313 & 56 & -27 & 18 & 78 & -60 & 27 & -27\\
-38 & -27 & 13 & 44 & 32 & -1 & -24 & -10\\
-20 & -17 & 10 & 33 & 21 & -6 & -16 & -9\\
-10 & -8 & 9 & 17 & 9 & -10 & -13 & 1\\
-6 & 1 & 6 & 4 & -3 & -7 & -5 & 5\\
2 & 3 & 0 & -3 & -7 & -4 & 0 & 3\\
4 & 4 & -1 & -2 & -9 & 0 & 2 & 4\\
3 & 1 & 0 & -4 & -2 & -1 & 3 & 1
\end{bmatrix}$$

DCT block

JPEG Compression



JPEG Compression



JPEG Compression

- The Baseline System – Quantization

$$X'(u,v) = Round\left(\frac{X(u,v)}{Q(u,v)}\right)$$

X(u,v): original DCT coefficient
X'(u,v): DCT coefficient after quantization
Q(u,v): quantization value

JPEG Compression

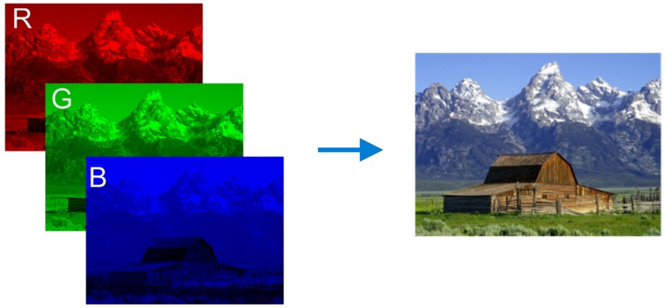
- Why quantization?
  - to achieve further compression by representing DCT coefficients with no greater precision than is necessary to achieve the desired image quality
  - Generally, the “high frequency coefficients” have larger quantization values
  - Quantization sets most coefficients to zero, making the compression system efficient, but it’s the main source that makes the system “lossy”

JPEG Compression

- Exploiting Psychovisual Redundancy
- Exploit variable sensitivity of humans to colors:
  - We're more sensitive to differences between dark intensities than bright ones.
    - Encode log(intensity) instead of intensity.
  - We're more sensitive to high spatial frequencies of green than red or blue.
    - Sample green at highest spatial frequency, blue at lowest.
  - We're more sensitive to differences of intensity in green than red or blue.
    - Use variable quantization: devote most bits to green, fewest to blue.

JPEG Compression

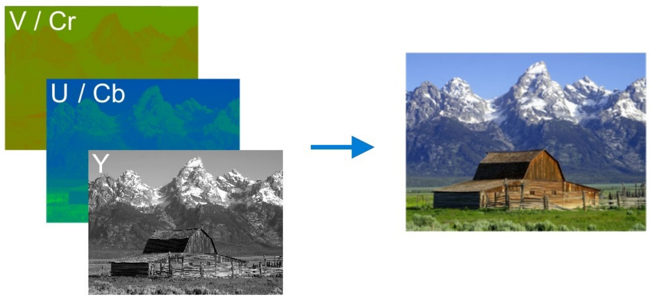
RGB



JPEG Compression

YUV – YCbCr encodes luminance and chrominance

YUV - YCbCr



JPEG Compression

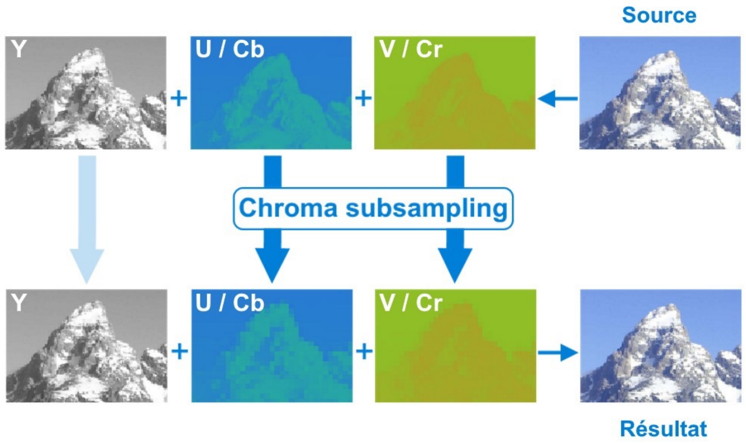
YUV – YCbCr encodes luminance and chrominance

$$\begin{bmatrix} Y' \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1146 & -0.3854 & 0.5 \\ 0.5 & -0.4542 & -0.0458 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5748 \\ 1 & -0.1873 & -0.4681 \\ 1 & 1.8556 & 0 \end{bmatrix} \begin{bmatrix} Y' \\ C_B \\ C_R \end{bmatrix}$$

JPEG Compression

- Exploiting Psychovisual Redundancy



JPEG Compression

- Exploiting Psychovisual Redundancy
  - In JPEG and MPEG

    Cb and Cr are sub-sampled