אוניברסיטת חיפה   החוג למדעי המחשב
Department of Computer Science   University of Haifa

Color Imaging Seminar

Lecture in the subject of

Perceptual Optimizations for JPEG Compression

Yair Moshe

Nov. 24th, 2004

Original by Yair Moshe - November, 2004

Extended By Hagit Hel-Or – June 2007

Additional Sources:

Dr. Philip TseMultimedia Coding and Processing Course

What We're Going to Talk About

- Introduction to image compression
- The JPEG standard
- JPEG perceptual extensions
  - DCTune
  - DCTune in color
  - Adaptive DCTune
- Summary

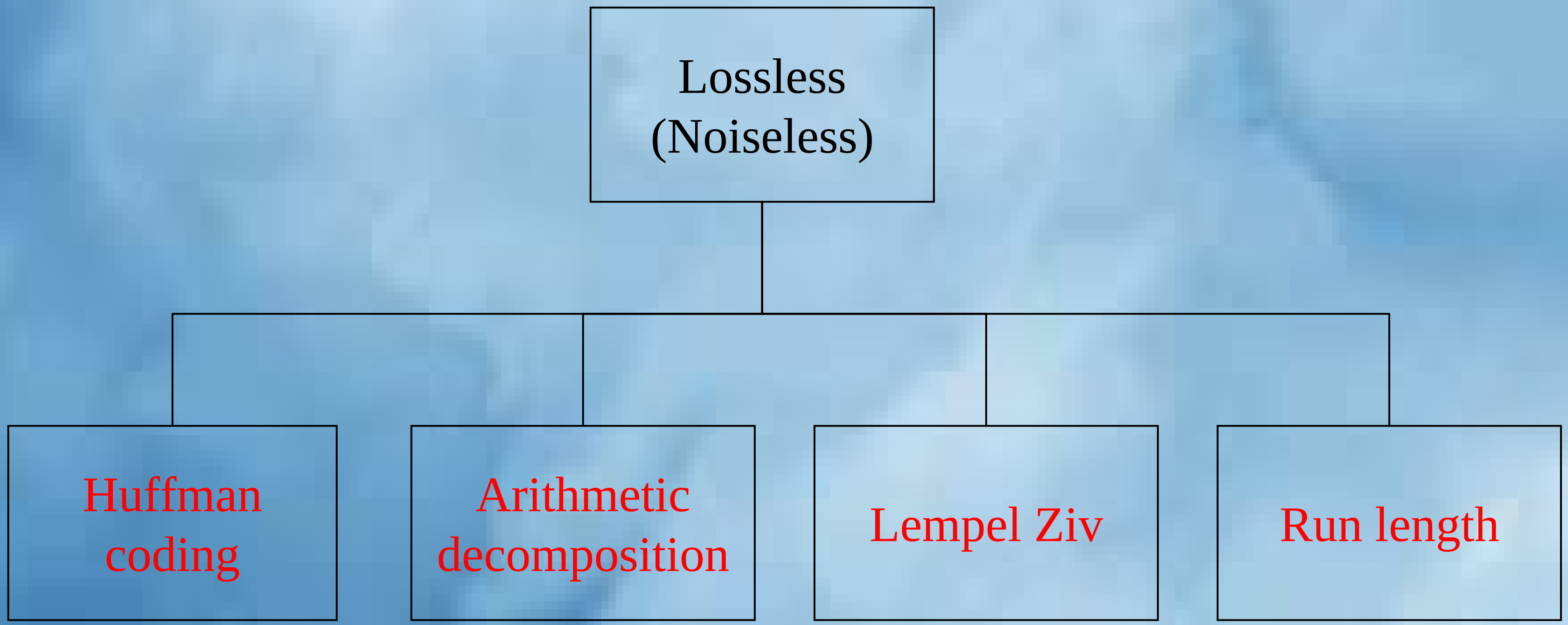
Image Compression

- Compression is conversion of information to a representation or form that requires fewer bits than the original
- Image compression is needed in order to make image files smaller and more efficient. Useful for:
  - Transmitting images across network
  - Storing images
- An example: 1280 pixels x 1024 pixels x 3 bytes per pixel = 3,932,160 bytes

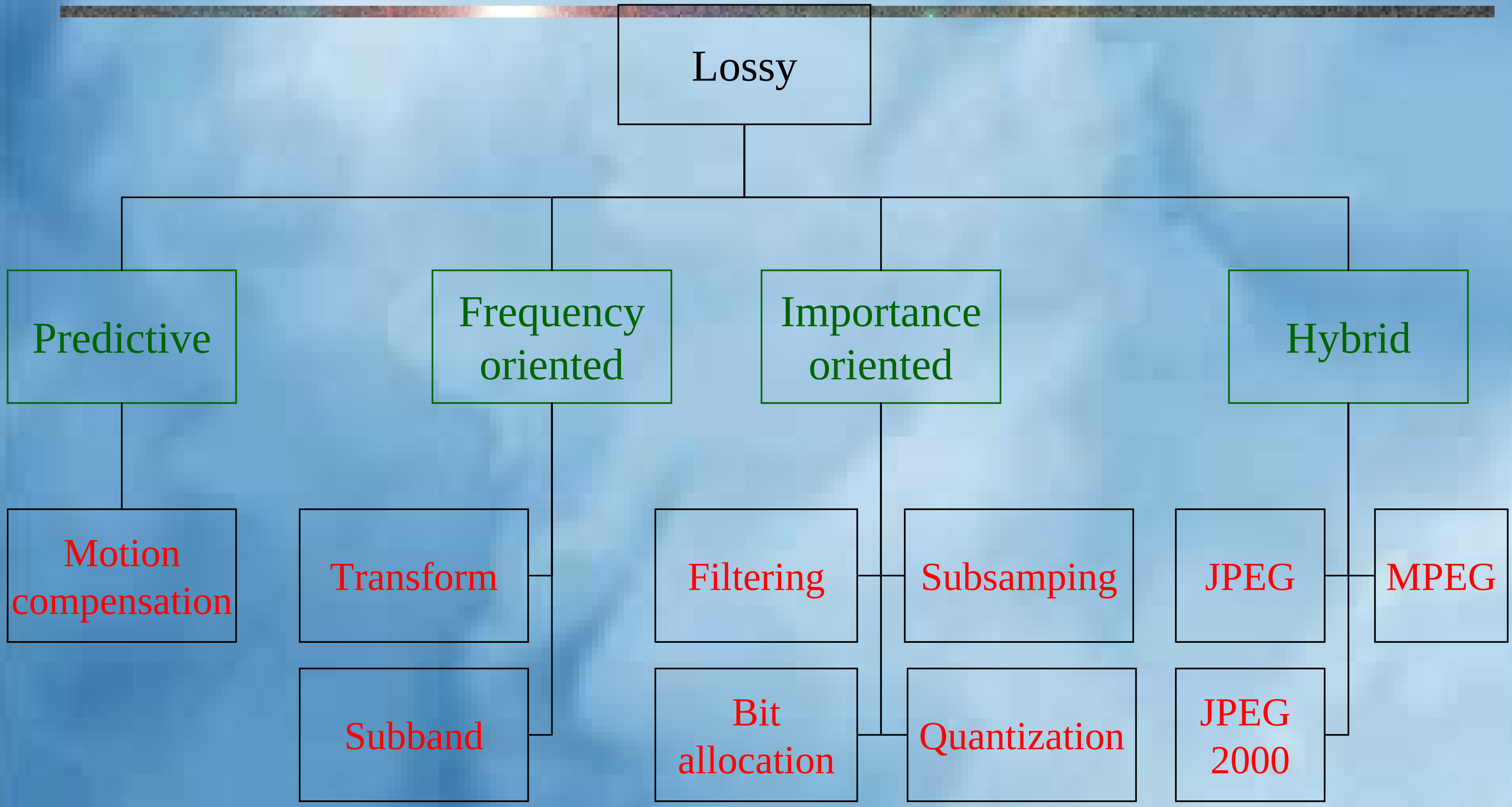
Lossless vs. Lossy

- Lossless image compression
  - Decompressed image is the same as the original
- Lossy image compression
  - Decompressed image is not the same as the original but looks quite similar
  - Exploits known limitations of the human visual system
  - Tradeoffs
    - Size against image quality
    - Time of compression/decompression against image quality
- Hybrid compression

Lossless Compression



Lossy Compression



Lossy Compression - Examples

Original image (230KB)

ABC-compressed image (3.65KB)

JPEG-compressed image (3.75KB)

JPEG2000-compressed image (3.78KB)

Is Lossy Compression "Better"?

+ Much higher compression ratios comparing to lossless compression
  - Typical compression ratios are 4:1 for GIF (lossless) comparing to 10-20:1 for JPEG (lossy)
- Intended to be looked at by humans and therefore may not be suitable for machine-processed images
- The loss accumulates as you repeatedly compress and decompress an image

Compression - Ratios

Lossless Compression Ratios:

- Text 2:1
- Bilevel images 15:1
- Facsimile transmission 50:1

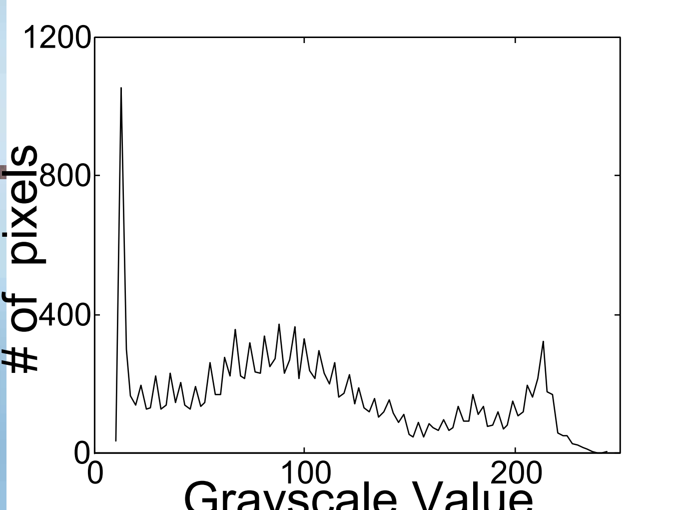
Lossy Compression Ratios:

- JPEG image 15:1
- H.261 or px64 100:1 to 2000:1
- MPEG video 200:1

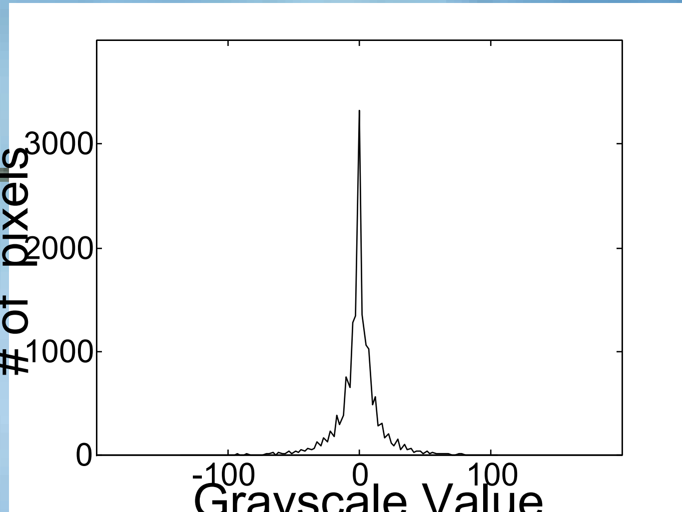
Image Redundancies to Exploit

- Spatial redundancy
  - Natural images tend to have flat areas
- Psycho-visual redundancy
  - Example: The eye is more sensitive to changes in luminance than to changes in chrominance
  - Another example: The eye is more sensitive to changes in low frequencies than to changes in high ones
- Statistical Redundancy
  - Some values are more frequent than others

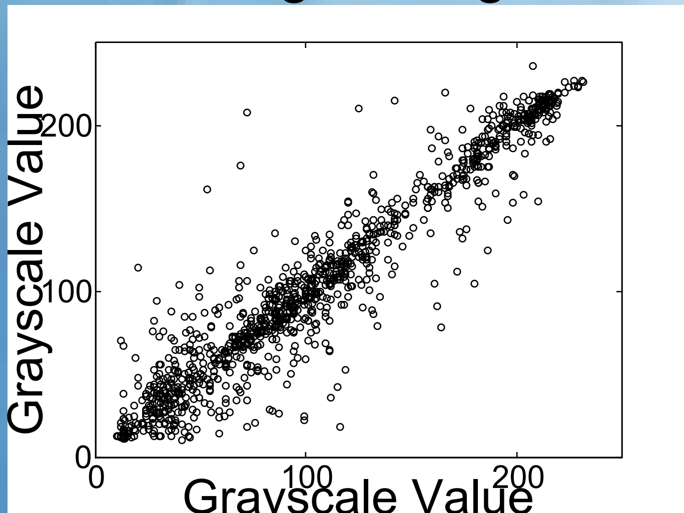
Image Histogram

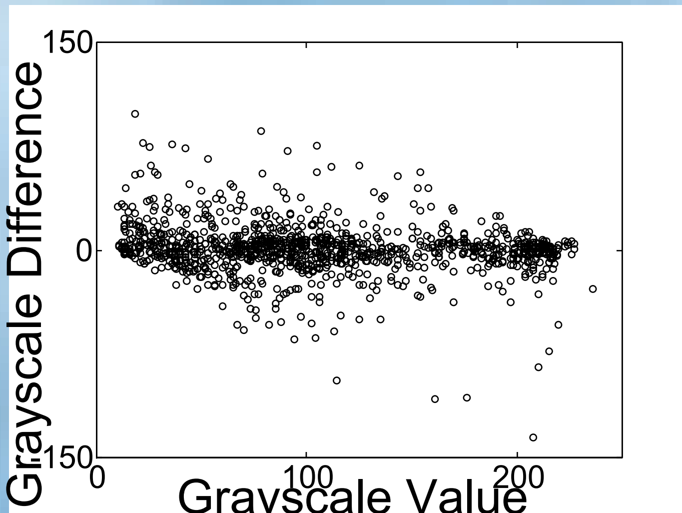


Difference Histogram



Neighboring Pixels are correlated:

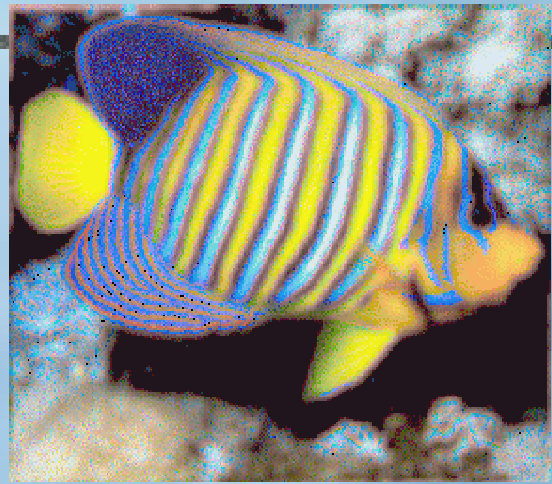




YIQ - Color Space

Original

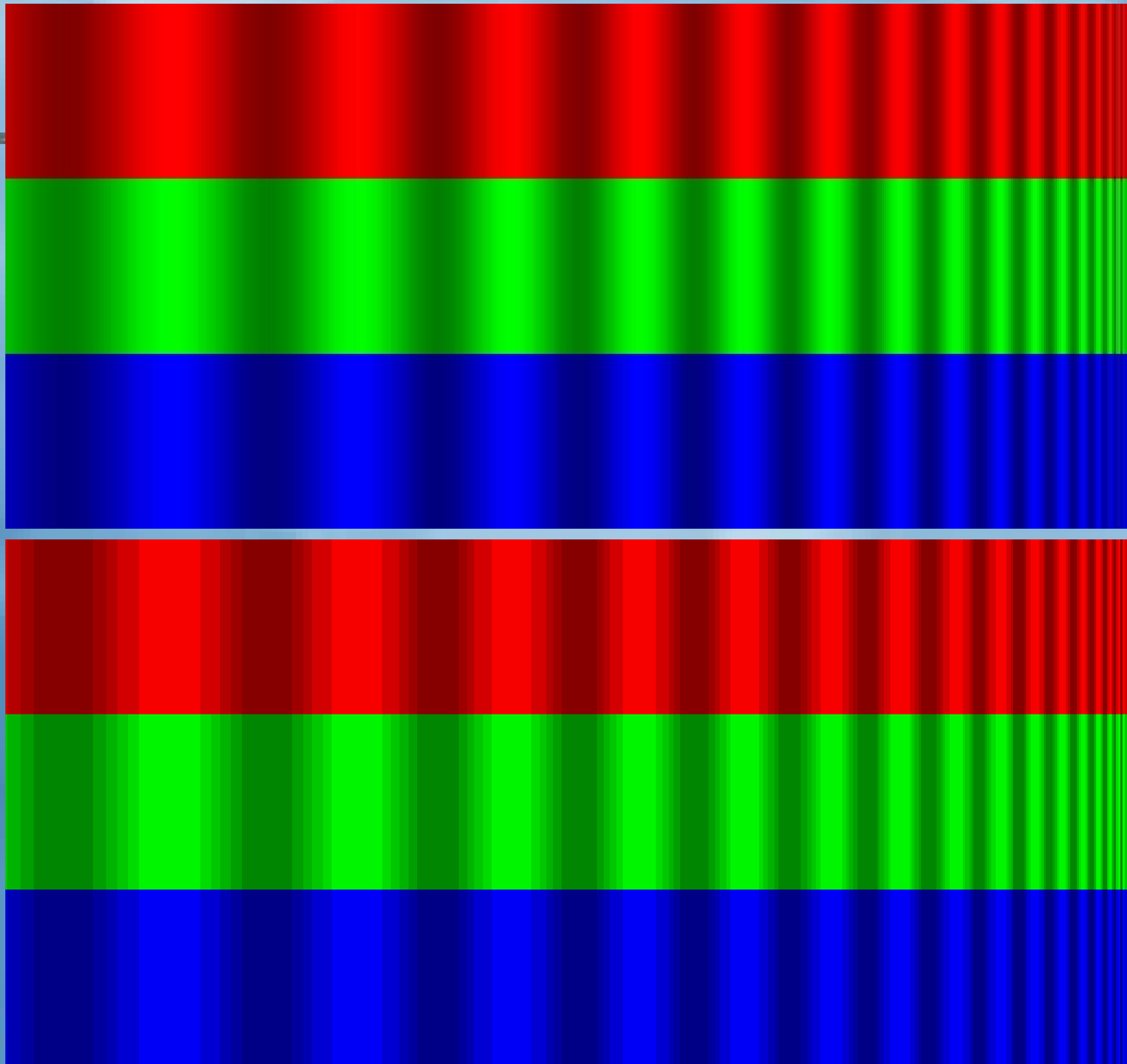
Y - Blur



I - Blur

Q - Blur

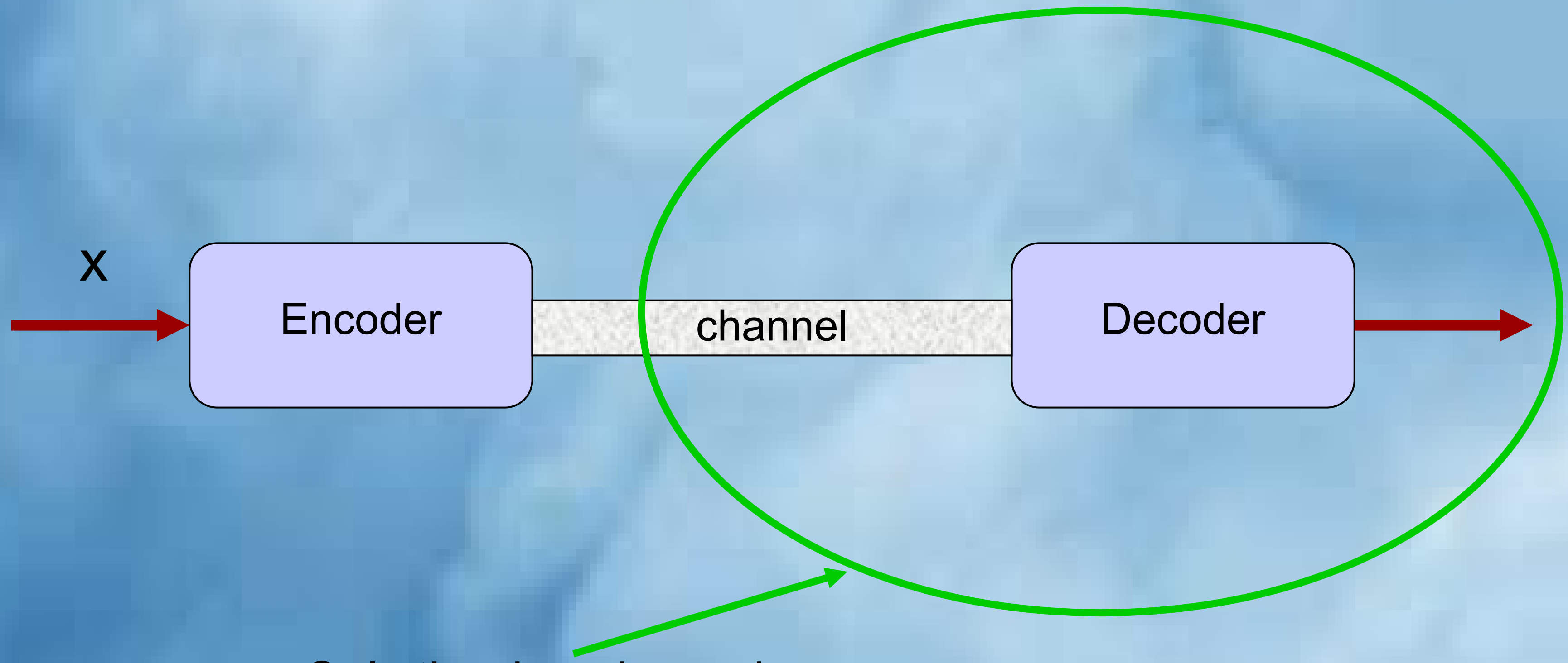
Quantization errors are spatially dependent



Multimedia Compression Standards

Short name	Official name	Standard Groups	Compression Ratios
JPEG 2000	Digital compression and coding of continuous-tone still images	Joint Photographic Experts Group	15:1 to 50:1 (full color still-frame applications)
H.261 px64	Video encoder/decoder for audio-visual services at px64 Kbps	Specialist Group on Coding for Visual Telephony	100:1 to 2000:1 (video-based tele-communications)
MPEG, MPEG2, MPEG4	Coding of moving pictures and associated audio	Moving Pictures Experts Group	50:1 to 2000:1 (motion-intensive applications)

Compression Standards



Only the decoder and syntax of information in the channel is standardized

What We're Going to Talk About

- Introduction to image compression
- The JPEG standard
- JPEG perceptual extensions
  - DCTune
  - DCTune in color
  - Adaptive DCTune
- Summary

The JPEG Standard

- A standardized image compression mechanism
- Created in 1992 by the JPEG (Joint Photographic Experts Group) committee
- Official names are
  - ISO/IEC DIS 10918-1
  - ITU-T Recommendation T.81
- Could be actually found everywhere today
  - Internet, digital photography, image archives and databases, motion JPEG…

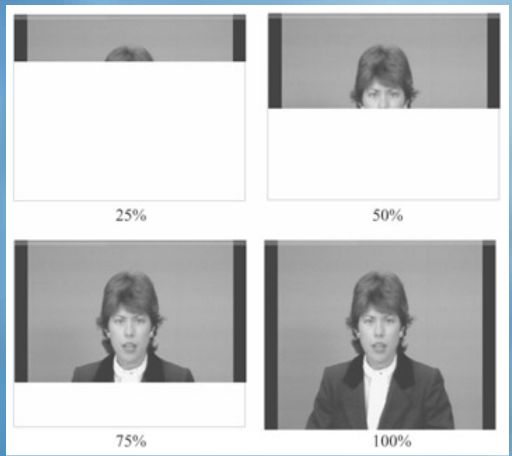
The JPEG Standard

- Designed for compressing natural, real-world scenes
- Lossy compression
- Can provide 10-20:1 compression of full-color data with subjectively transparent quality
- Deals with gray-scale or full color images
  - 24 bits/pixel (16 million colors).

JPEG Modes

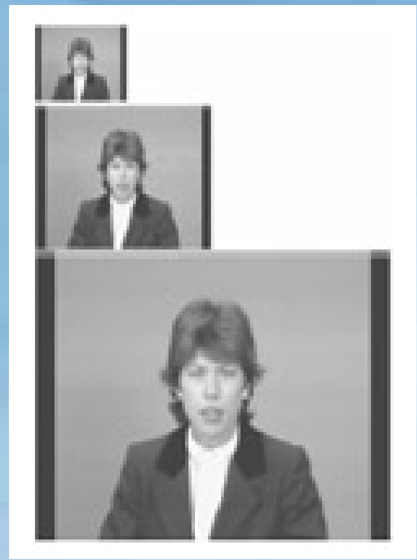
- Three modes of operation

Baseline Sequential



Progressive

Hierarchical



- Lossless mode is also supported

This is the most common mode and the only one we're going to talk about

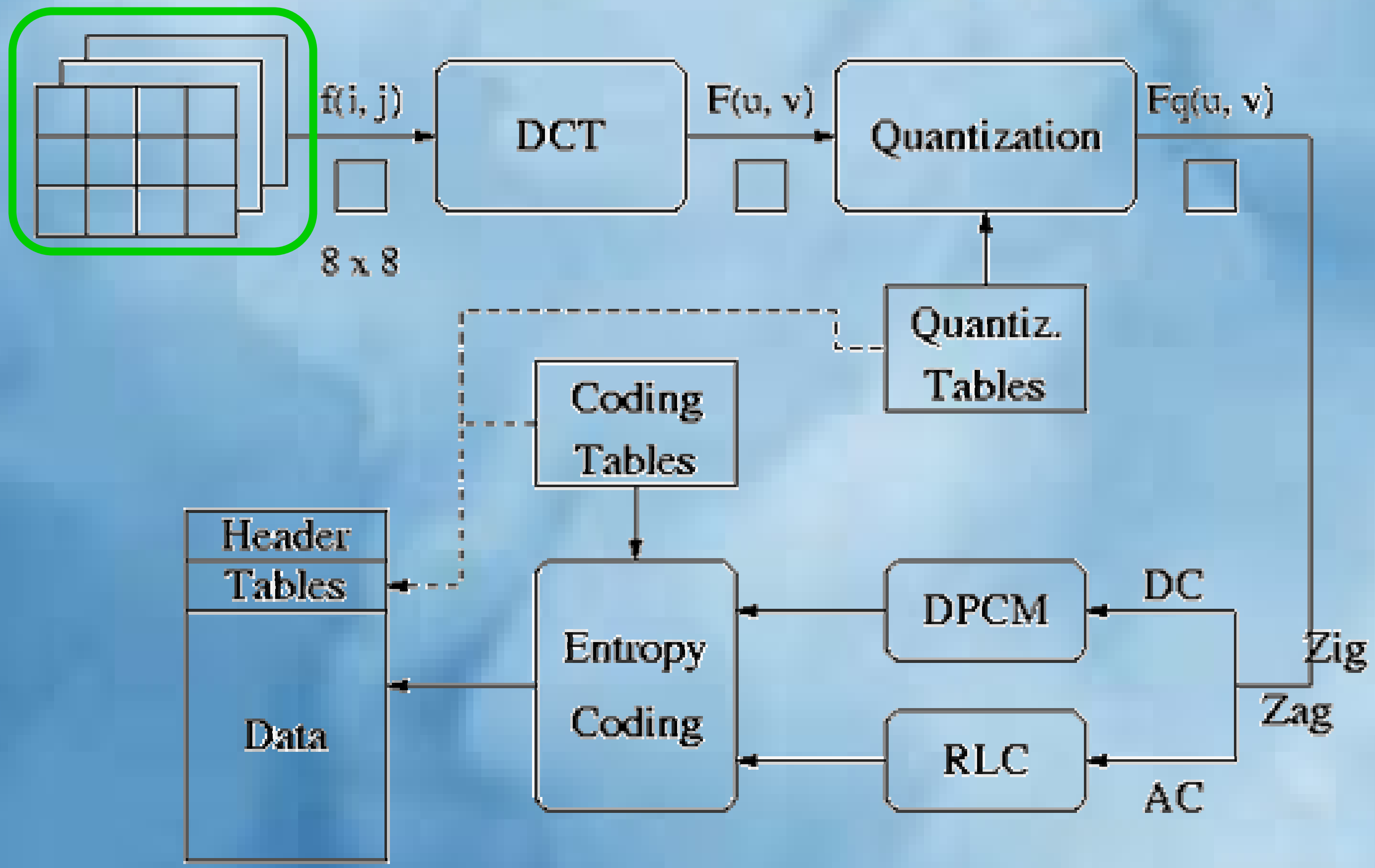
Progressive Encoding - example

- Progressive encoding
  - Increases resolution when more data are received
  - Suitable for low bandwidth transmission links

Non-progressive

Progressive

JPEG Encoder



Color Space Conversion

- The YCbCr color space is used
  - RGB: Red/Green/Blue
  - YCbCr: Luminance/Blue Chrominance/Red Chrominance
    - Advantage: The human visual system is less sensitive to color than to luminance
  - RGB to YCbCr conversion:

$$Y = 0.299R + 0.587G + 0.114B$$

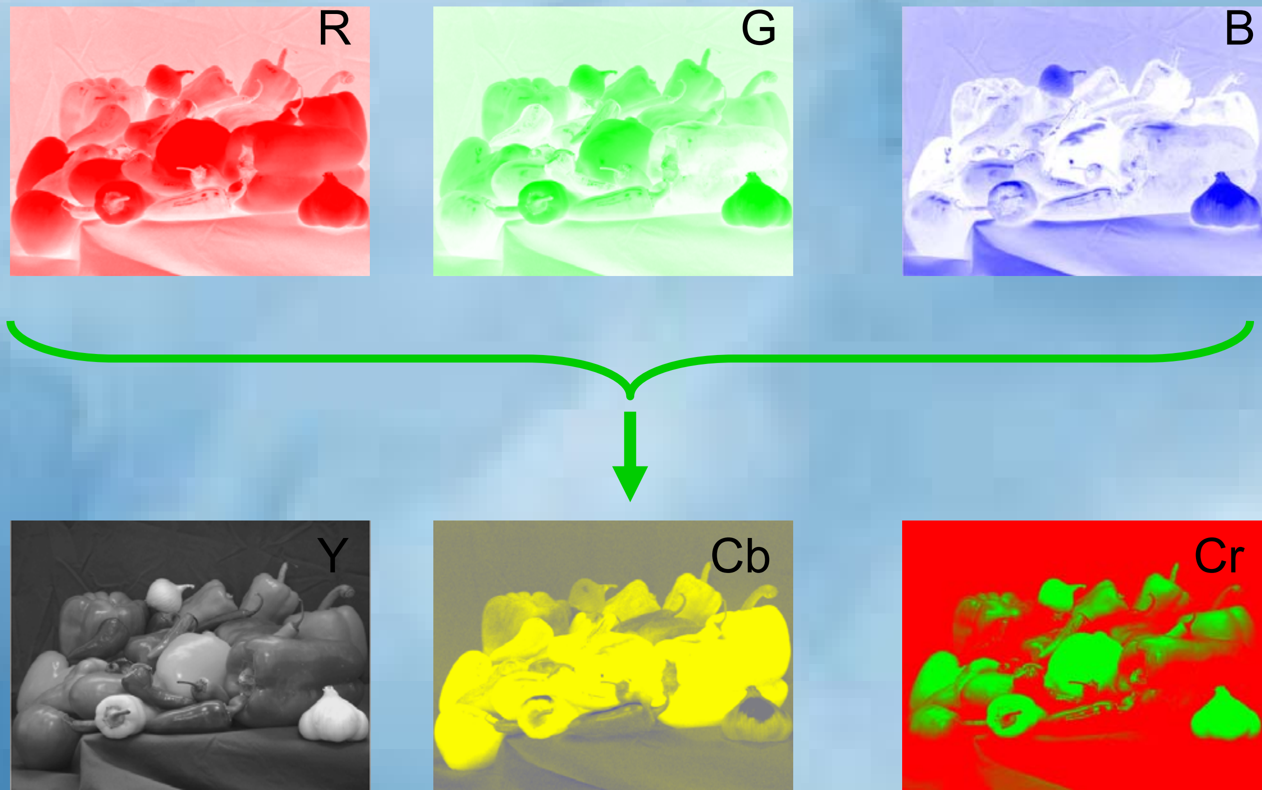
$$Cb = 0.564(B - Y)$$

$$Cr = 0.713(R - Y)$$

Color Space Conversion

- Example:

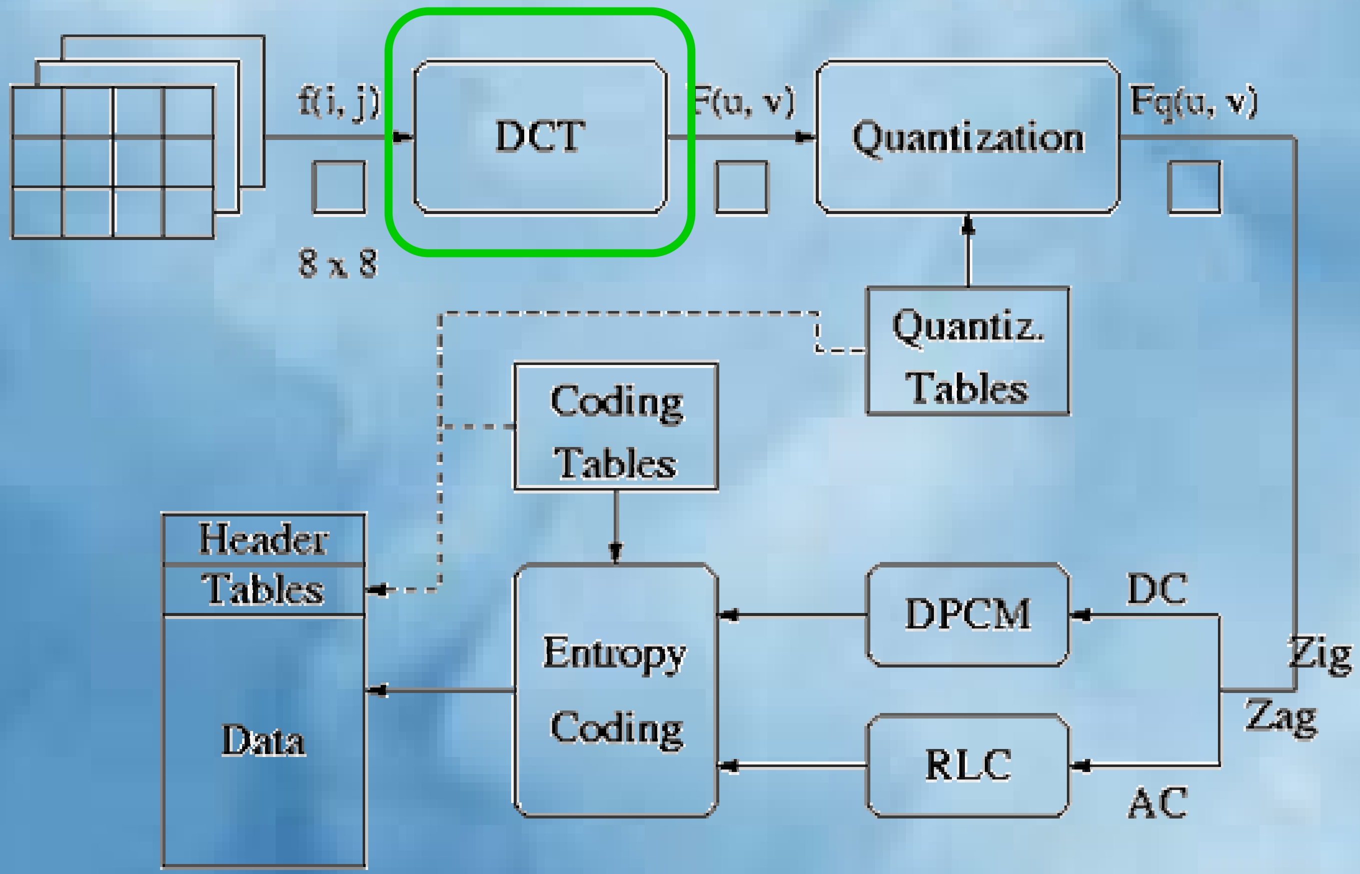
=



Color Space Conversion

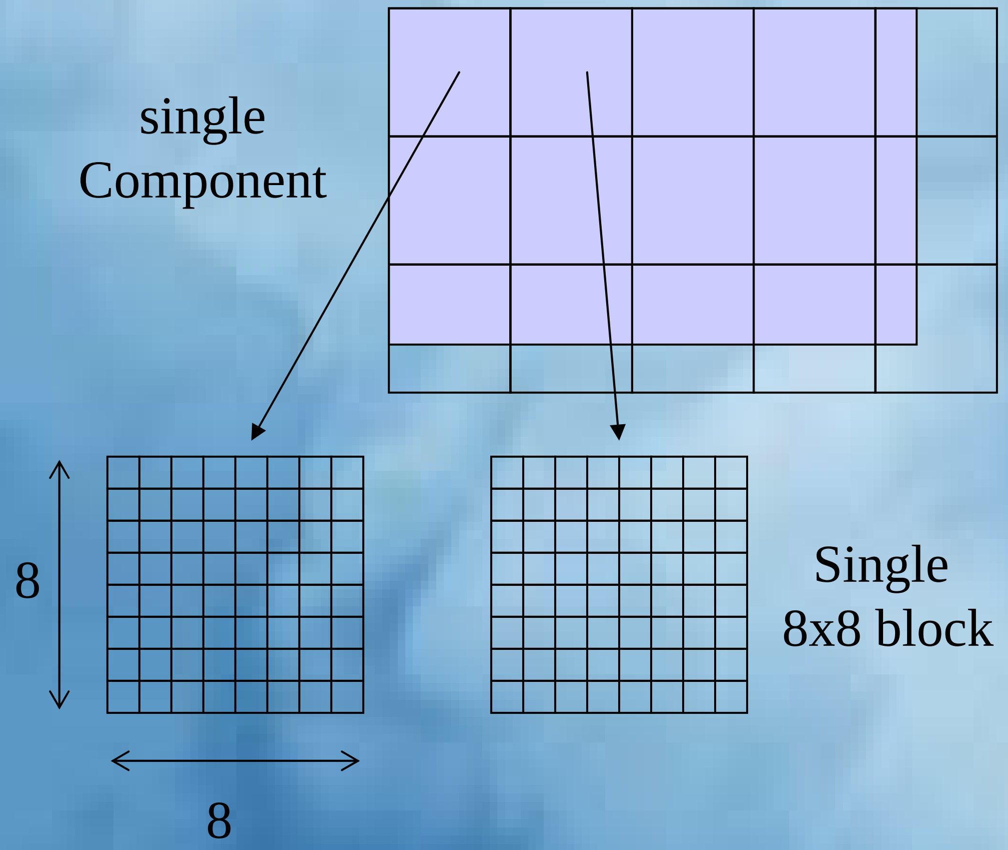
- Color space conversion is optional but important
- Another optional action is down-sampling the chrominance component
  - 4:2:2 : Down-sample 2:1 horizontally
  - 4:1:1 : Down-sample 2:1 horizontally and 2:1 vertically
- Input data is shifted so it is distributed about zero
  - An 8-bit input sample in the range [0 255] is shifted to the range [-128 127] by subtracting 128

JPEG Encoder



DCT (Discrete Cosine Transform)

- Image is processed one 8x8 block at a time



DCT (Discrete Cosine Transform)

- A 2-D DCT transform is applied to the block:

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

DCT coefficients ($F(u,v)$); $C(u)$, $C(v)$ as defined below; samples ($f(x,y)$)

$$C(n) = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ 1 & n \neq 0 \end{cases}$$

  - Perform nearly as well as the ideal KLT transform
  - Closely related to the DFT transform
  - Computationally efficient

DCT (Discrete Cosine Transform)

8x8 block of image data

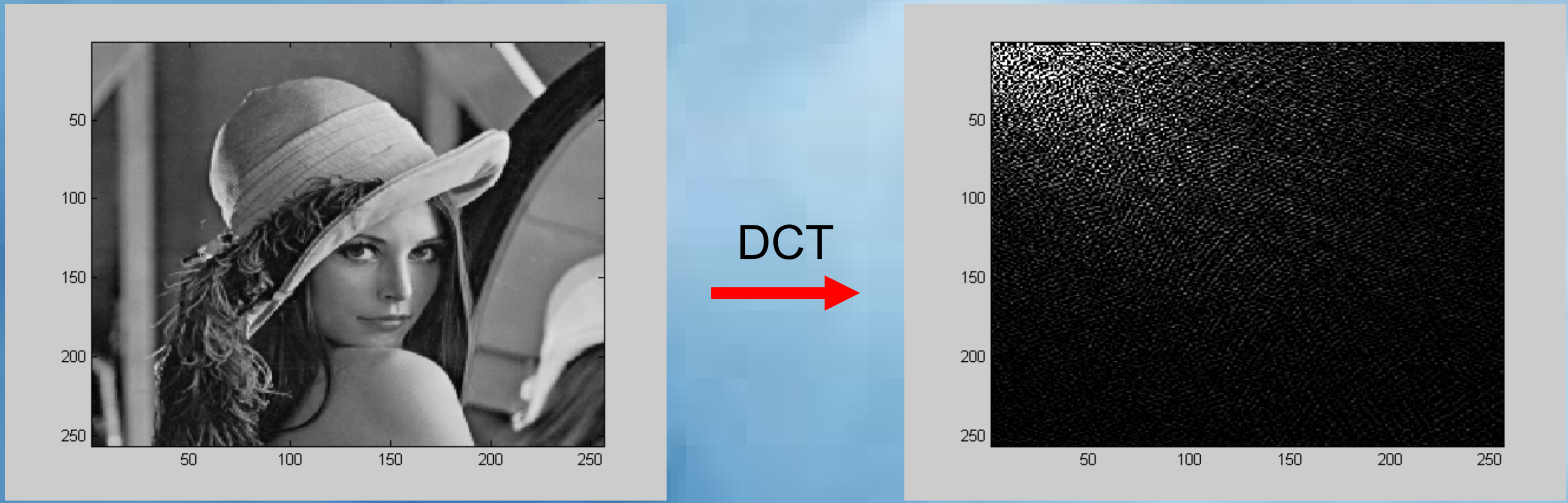
232	233	234	228	222	217	216	217
229	229	227	224	221	217	215	216
226	226	223	221	220	217	216	216
226	223	222	221	219	219	217	217
225	223	222	221	219	219	219	219
223	223	221	221	218	219	220	221
222	224	223	223	222	222	223	226
222	225	226	225	224	225	229	233

DCT coefficients after transformation

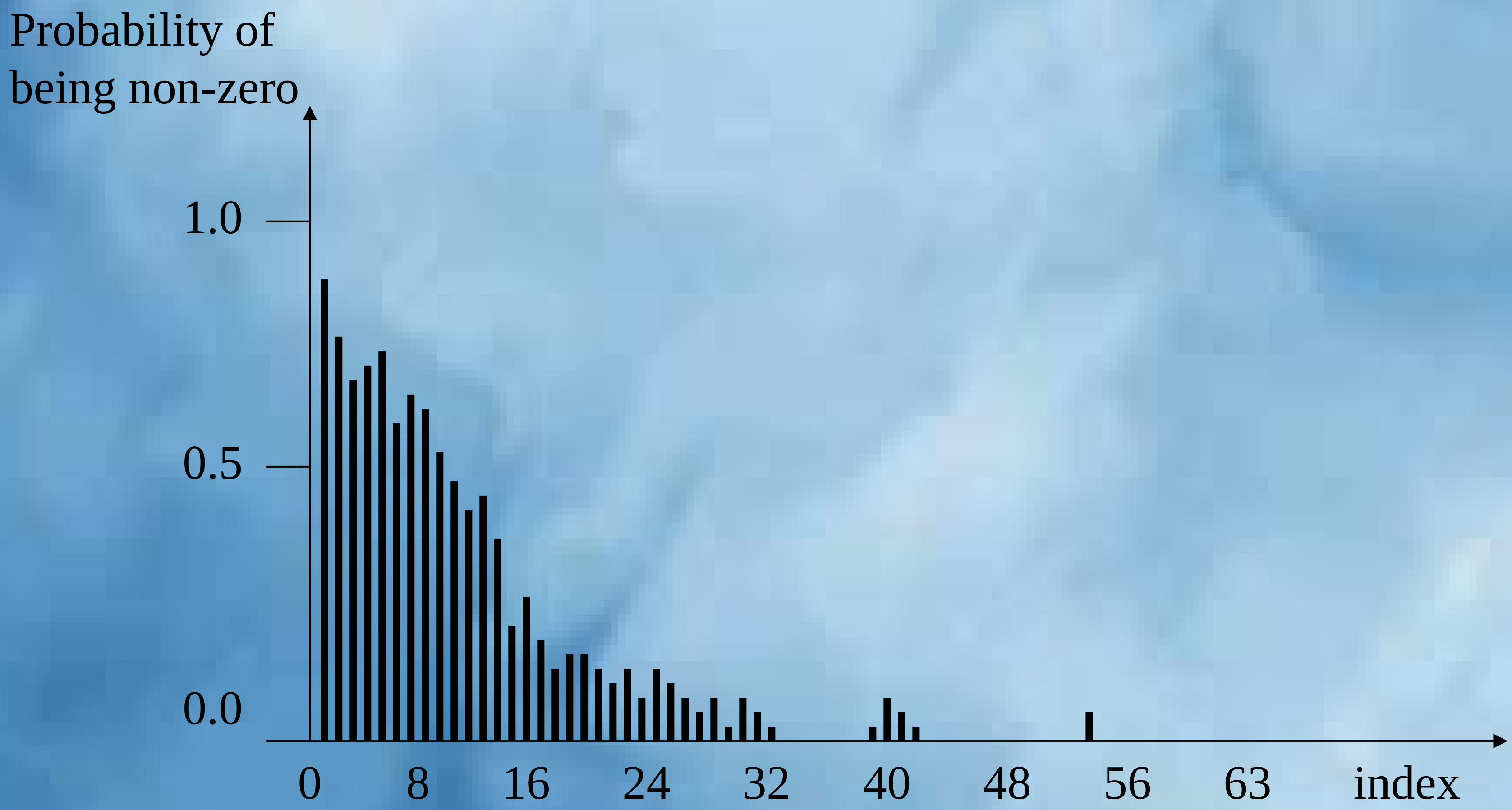
1779	19.3	4.0	-5.2	0.6	-0.6	0.1	-0.6
-3.0	15.7	-2.4	0.4	0.2	-0.1	0.4	0.8
14.7	-0.1	-0.5	-4.2	-0.5	-0.9	-0.2	0.6
-0.8	3.5	-0.2	-0.8	-0.2	0.8	0.2	-0.3
5.6	-0.3	0.1	-0.9	-0.3	0.9	0.8	0.5
-0.2	0.7	-0.6	0.0	0.0	-0.1	0.3	-0.1
0.0	-0.1	0.5	-0.6	-0.5	0.2	-0.4	0.1
0.7	0.1	-0.2	-0.2	-0.1	-0.4	-0.3	0.4

DCT (Discrete Cosine Transform)

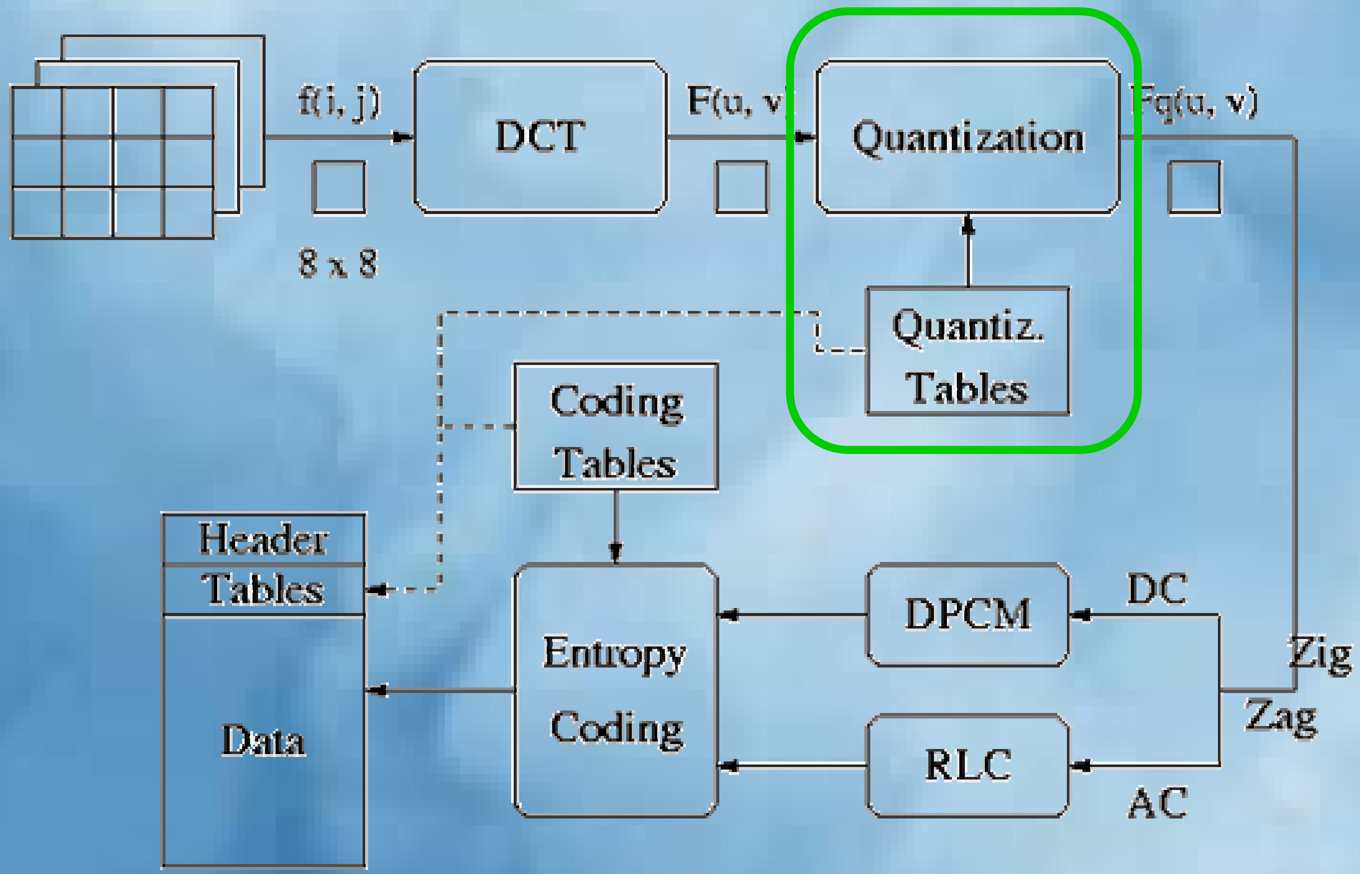
- DCT transform advantages:
  - Decrease spatial redundancy by decorrelating the information in the block
  - Leave only a small number of visually significant transform coefficients



Probability Distribution of DCT Coefficients



JPEG Encoder



Quantization

- Quantization is the process of approximating a signal by a limited number of values
  - Allows to reduce the number of bits needed to represent the signal
- This is the only place in the encoder where information is lost
  - Many-to-one mapping
  - Main process to control image quality and compression
- Remove the components of the transformed data that are less visually significant

JPEG Quantization

- For every block, the 64 DCT coefficients are uniformly quantized
- Step size is given in an application-specific quantization matrices
  - This allows different weighting to be applied according to the sensitivity of the human visual system to a coefficient of the frequency (psycho-visual redundancy)
  - Different matrices to luminance an to chrominance
  - Matrices are derived empirically
  - Quantization matrices must be sent to the decoder

Quantization

- Quantization process at the encoder:

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

Transform coefficient: $F(u,v)$

Quantized transform coefficient: $F^{q}(u,v)$

Quantization parameter (quantization table value): $Q(u,v)$

- De-quantization process at the decoder:

$$F^{Q}(u,v) = F^{q}(u,v) * Q(u,v)$$

reconstructed quantized coefficient: $F^{Q}(u,v)$

Sample Quantization Tables

Luminance

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

Chrominance

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

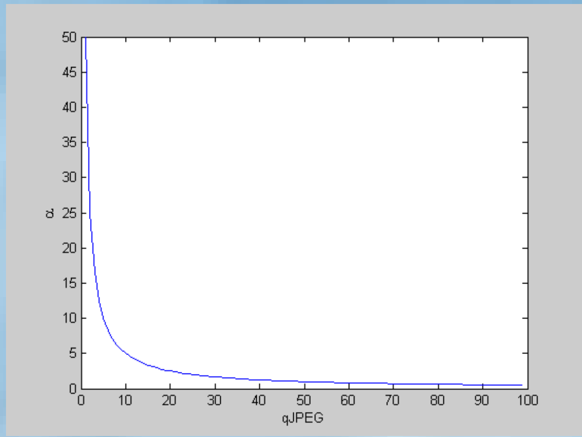
Quality Factor

- Quantization matrices and the quality factor are not standardized but the recommendation is used by many implementations
- A quality factor (q_JPEG) controls the elements of the quantization matrix
- $1 \leq q\_JPEG \leq 100$
- Typical values used are: $50 \leq q\_JPEG \leq 95$
- The quantization matrices we've just seen are for $q\_JPEG = 50$

Quality Factor

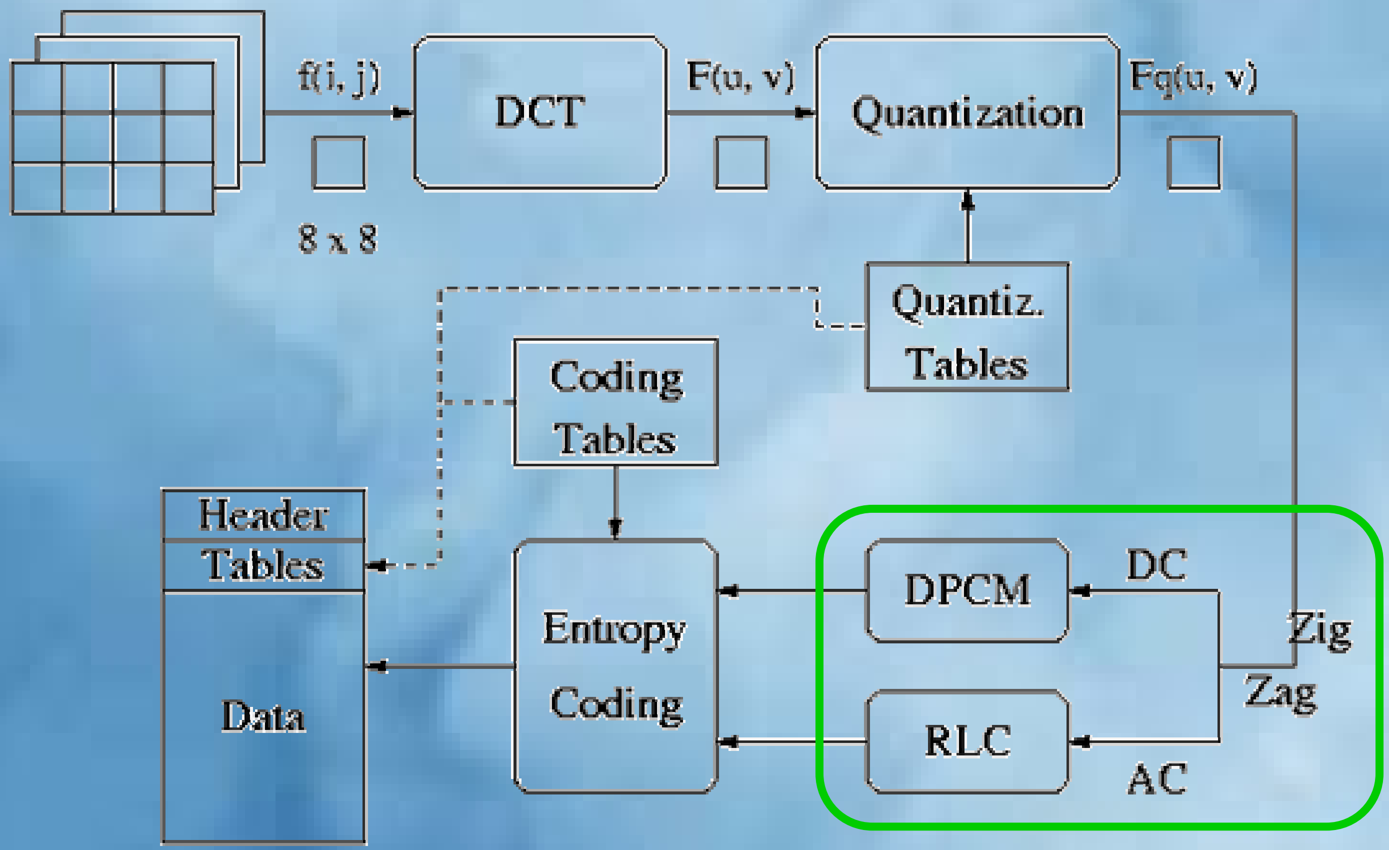
- For other quality factors, the elements of these quantization matrices are multiplied by the compression factor α, defined as:

$$\alpha = \begin{cases} \dfrac{50}{q\_JPEG} & 1 \le q\_JPEG \le 50 \\ 2 - \dfrac{2q\_JPEG}{100} & 51 \le q\_JPEG \le 99 \end{cases}$$



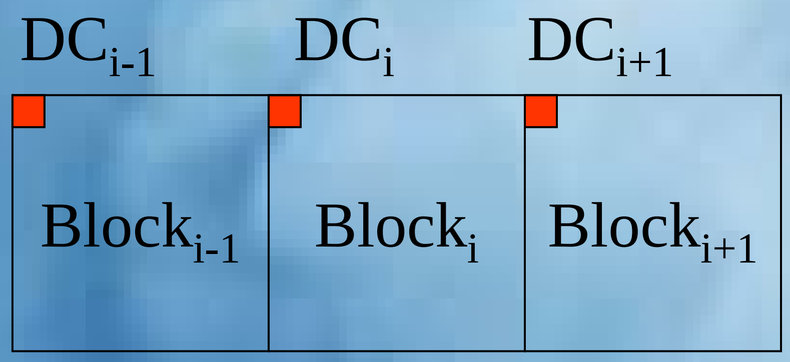
- The minimum legal α·Q(u,v) is 1
- For α=100, all Q(u,v) equal 1
  - This is not lossless compression!
- Let's look at an example…

JPEG Encoder



Coding of Coefficients

- The DC (the upper leftmost) coefficient and the 63 AC coefficients are coded separately
- The DC is a measure of the average value of the 64 image samples
- All DCs of all blocks are encoded first.



Coding of Coefficients

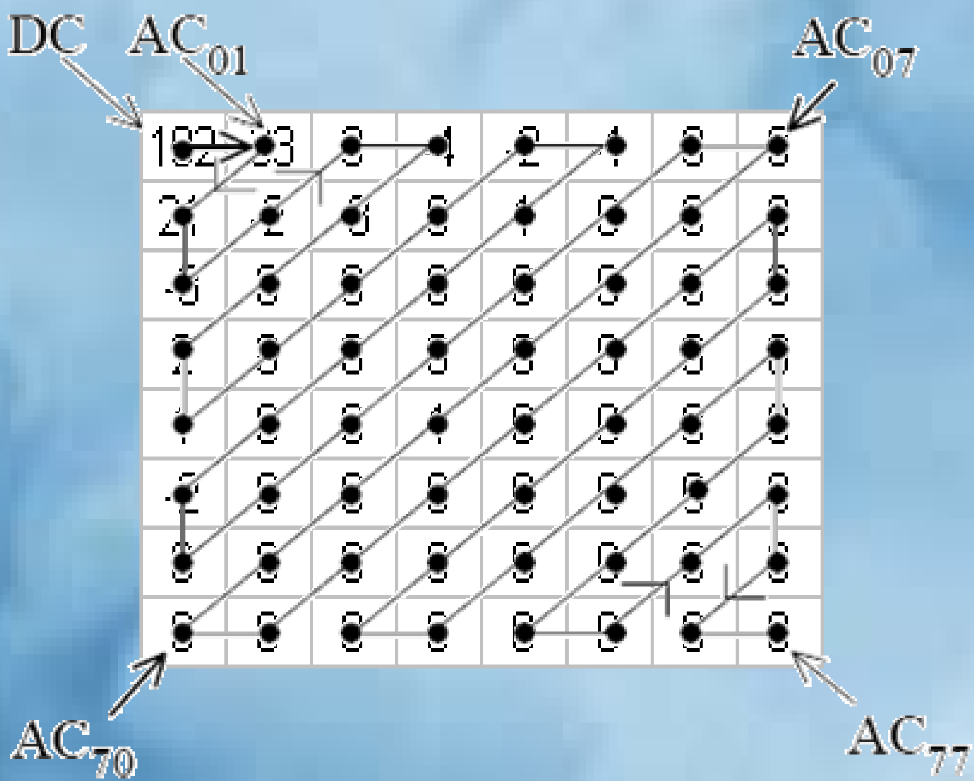
- DC is predicted from the previous block
  - Only the difference is coded:

$$DC_{pred} = DC_{cur} - DC_{prev}$$

  - Exploits the spatial correlation between DC coefficients in adjacent blocks

Zigzag Scan

- AC coefficients are reordered using zigzag scan and run-level coded.

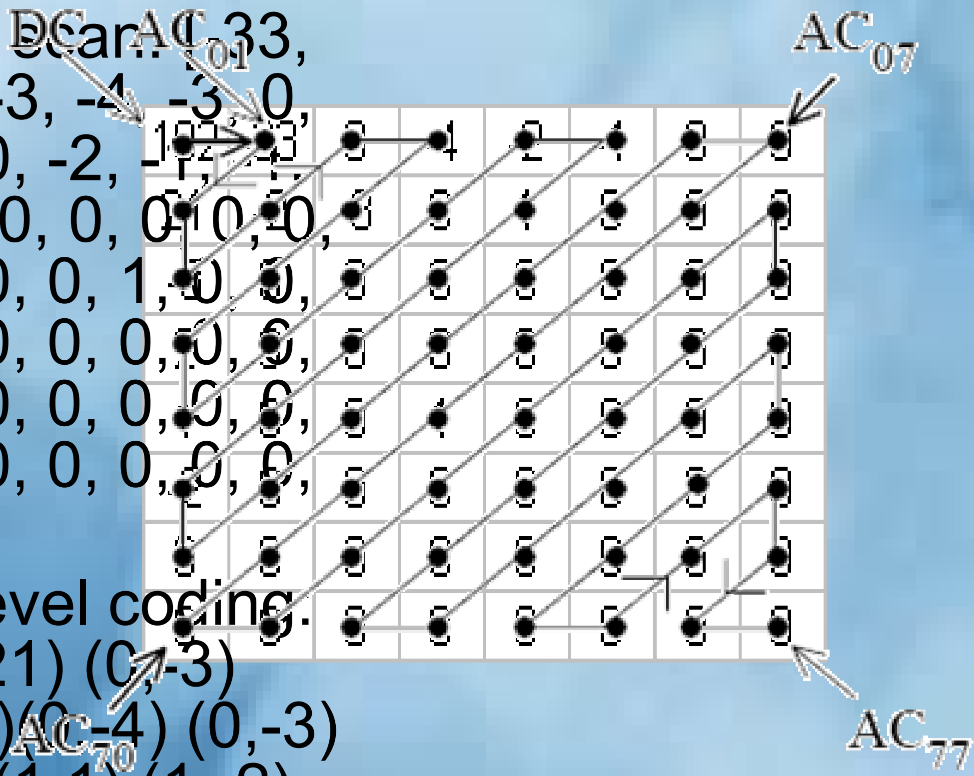


- Facilitates entropy coding by encountering the most likely nonzero coefficients first

Run-Level Coding

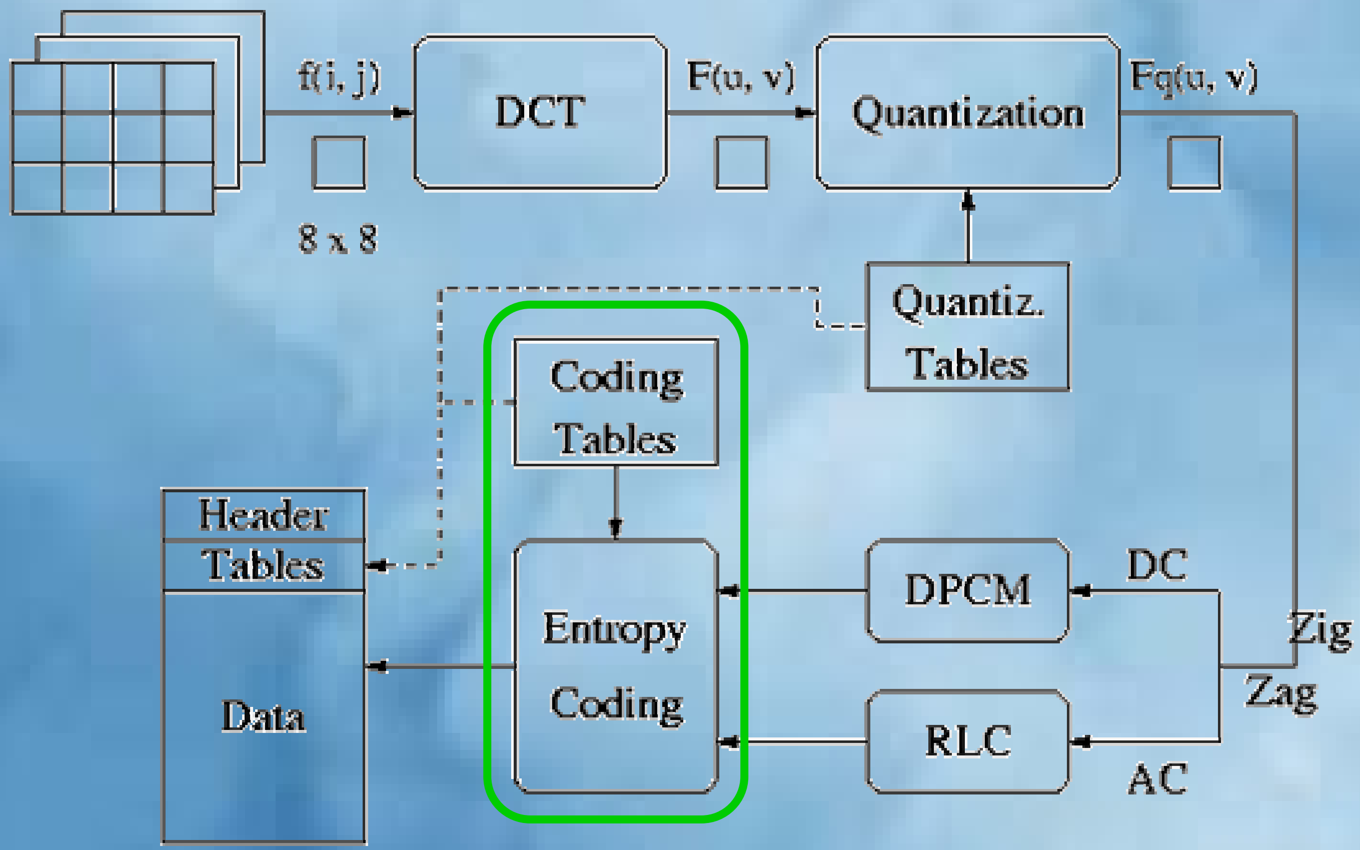
- Compactly represents coefficients as a series of (run, level) pairs
  - Run indicates the number of zeros preceding a non-zero value
  - Level indicates the sign and magnitude of the non-zero value following them

Run-Level Coding



- After zizag scan: [-33, 21, -3, -2, -3, -4, -3, 0, 2, 1, 0, 1, 0, -2, -1, -1, 0, 0, 0, -2, 0, 0, 0, 0, 0, 0, 0, 0, 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]
- After run-level coding: (0,-33) (0,21) (0,-3) (0,-2) (0,-3) (0,-4) (0,-3) (1,2) (0,1) (1,1) (1,-2) (0,-1) (0,-1) (3,-2) (11,1)

JPEG Encoder



Entropy Coding

- Represents frequently occurring (run, level) pairs with a small number of bits and infrequently occurring (run, level) pairs with a larger number of bits
  - Decrease statistical redundancy
- Two alternative entropy coding methods are allowed in JPEG:
  - Huffman coding
  - Arithmetic coding
- In both cases, coding tables must be sent to the decoder

Huffman vs. Arithmetic Coding

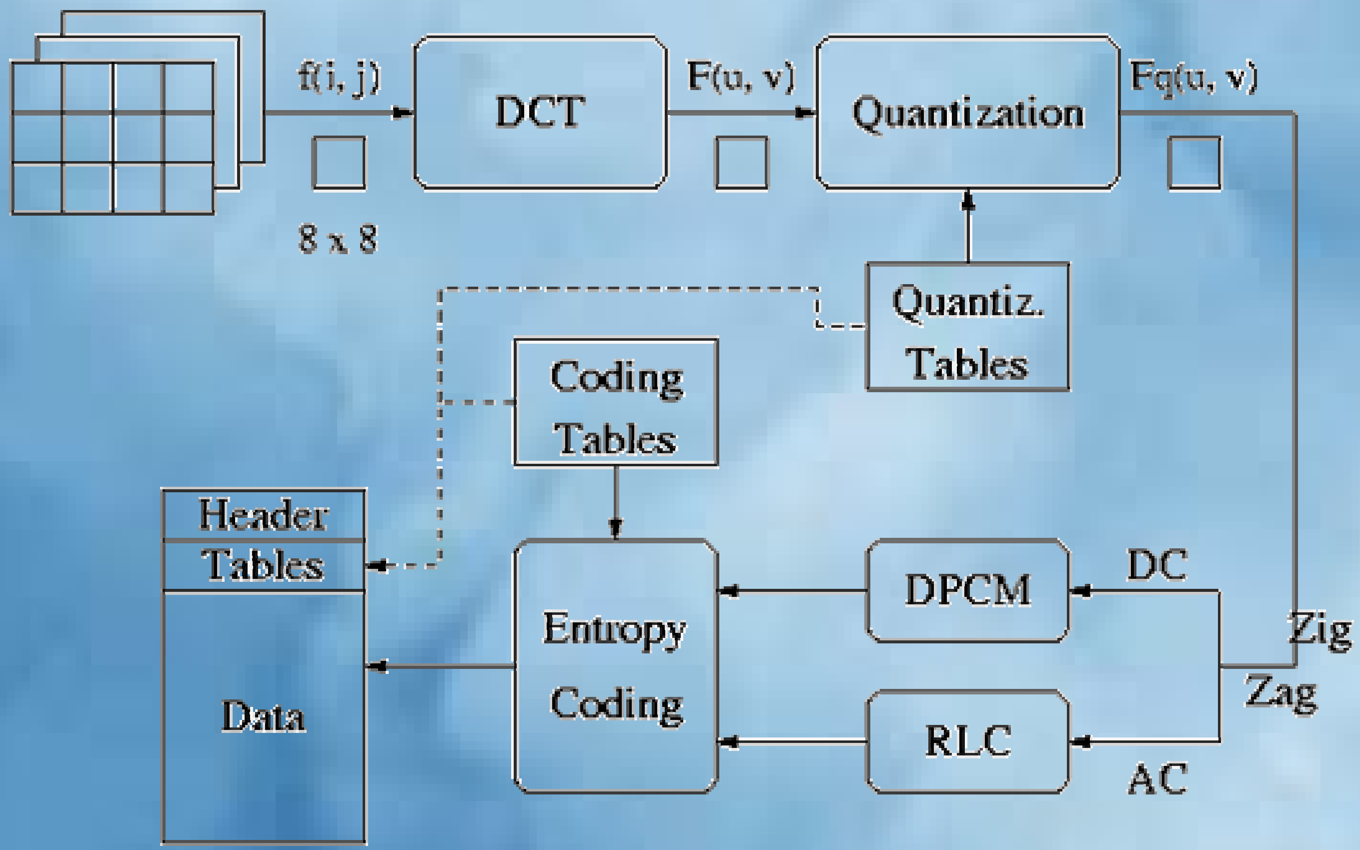
- Huffman coding
  - Maps each input symbol to a codeword with an integral number of bits
- Arithmetic coding
  - Codewords can have fractional number of bits
  - Better compression (produces a 5-10% smaller images)
  - Higher complexity
  - Subject to patents in JPEG so is not commonly used

Huffman Coding Example

No Huffman code contains other code as a prefix

Symbol	Probability	"regular" Code	Huffman code
A	0.014	000	01011
B	0.024	001	01001
C	0.117	010	00
D	0.701	011	1
E	0.101	100	011
F	0.027	101	01000
G	0.016	110	01010
bits/symbol		3	1.557

JPEG Encoder



JPEG Example

Original

JPEG Example

JPEG Compressed - High Quality

JPEG Example

JPEG Compressed - Low Quality

JPEG Example

Original

JPEG Example

JPEG Compressed - Low Quality

What We're Going to Talk About

- Introduction to image compression
- The JPEG standard
- JPEG perceptual extensions
  - DCTune
  - DCTune in color
  - Adaptive DCTune
- Summary

Perceptual Extensions

- Quantization matrices are not specified in the standard and they have a major influence on the tradeoff between image quality and bit-rate

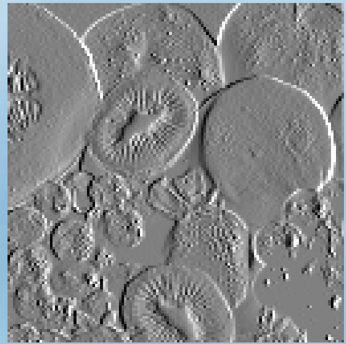
Let’s optimized them according to the properties of the human visual system

Naive Approach

Peterson et al. (1991,1992)
Ahumada et al. (1992,1993)

- Image-Independent Perceptual approach
- For each frequency *ij*, measure the smallest coefficient that yields a visible signal – t_{ij}
- Use this threshold to ensure that all errors are invisible
- The matrix is computed independent of the image
  - Big problem since visual thresholds are dependent of the image upon which they are superimposed

Visual sensitivity is variable - visibility of errors depends on the background image (frequency and orientation).



noise

image1 + noise

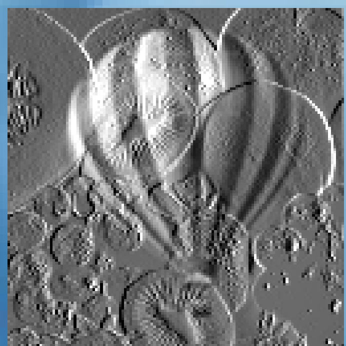
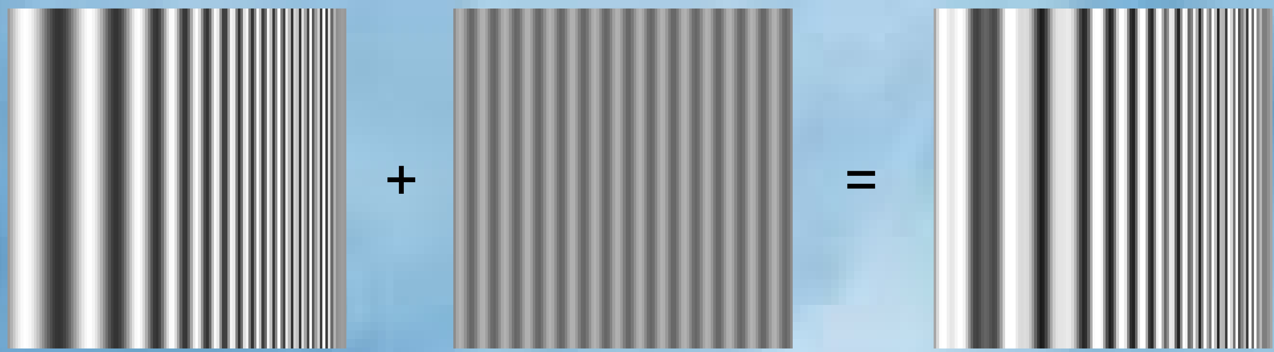
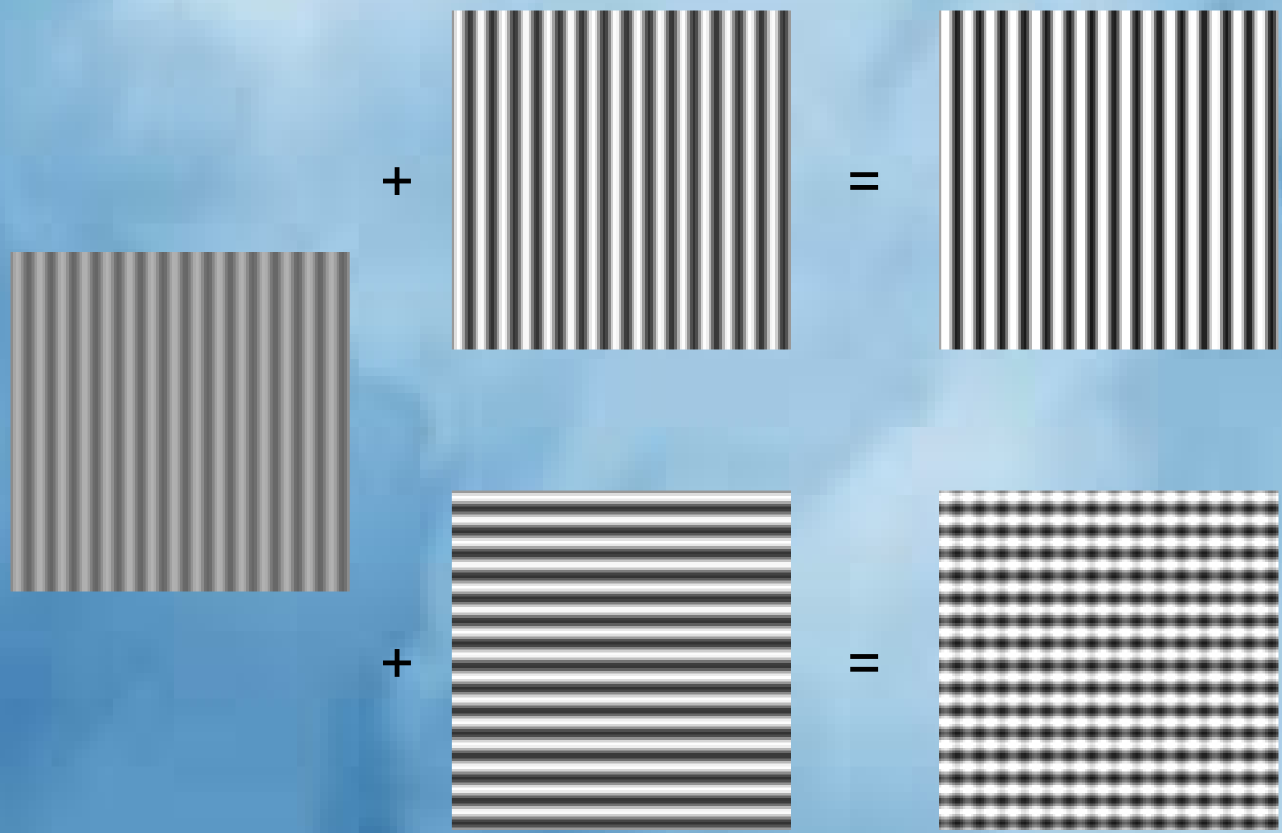


image2 + noise

Noise depends on frequency



Noise depends on Orientation



DCTune

Watson (1993)

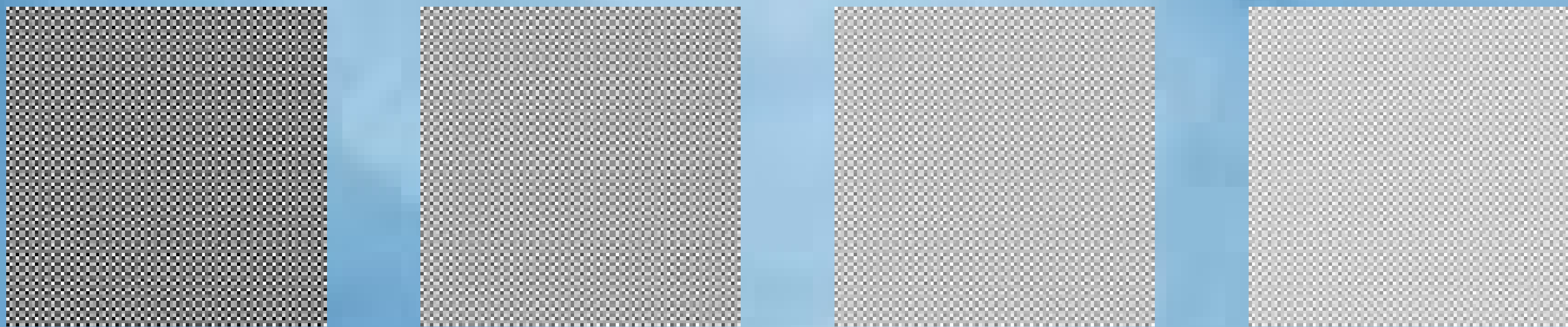
- Design custom quantization matrix tailored to a particular image
- Use properties of the human visual system:
  - Luminance masking
  - Contrast masking
  - Error pooling
- Allow selectable quality

Luminance Masking

- Threshold for a luminance pattern depends substantially upon the mean luminance of the local image region
  - The brighter the background, the higher the luminance threshold

Luminance Masking

Each of these images have the same amplitudes (contrasts) but with different means.



Quantization of higher luminance, would be less noticeable than lower luminance using the same quantization bin size.

Therefore, a larger amount of quantization will be possible in brighter regions, without loss of significant data.

Luminance Masking

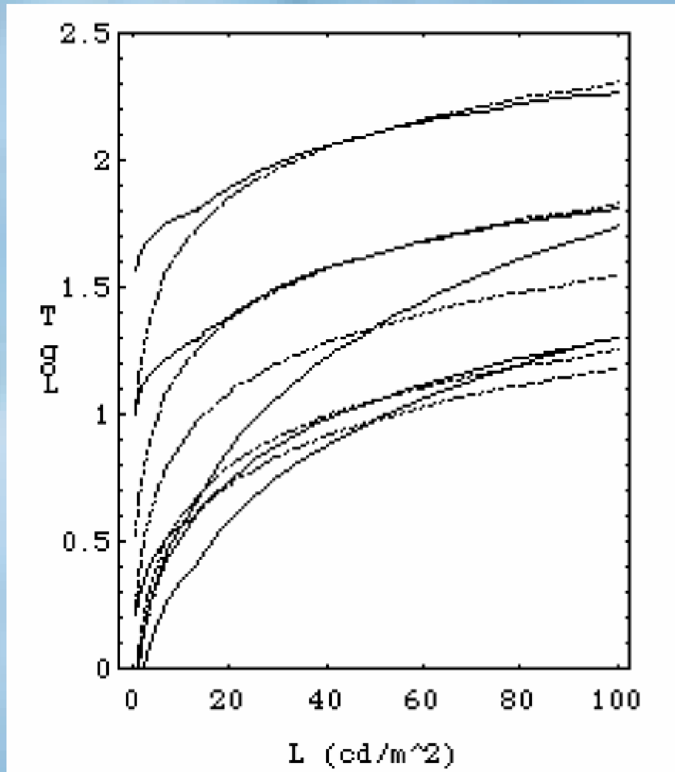
DC coefficient of block k

Control the degree of masking. Suggested value: 0.649

Threshold from naive approach

$$t_{ijk} = t_{ij}\left(\frac{c_{00k}}{\overline{c}_{00}}\right)^{a_T}$$

DC coefficient of mean display luminance



log(t_{ij}) as function of the luminance of the block *L* for 5 frequencies

- a_T=0 : No luminance masking

Contrast Masking

- Threshold for a visual pattern is typically reduced in the presence of other patterns
  - Threshold error in a DCT coefficient in a particular block is a function of the value of that coefficient in the original image
- Use a model that has been widely used in vision models

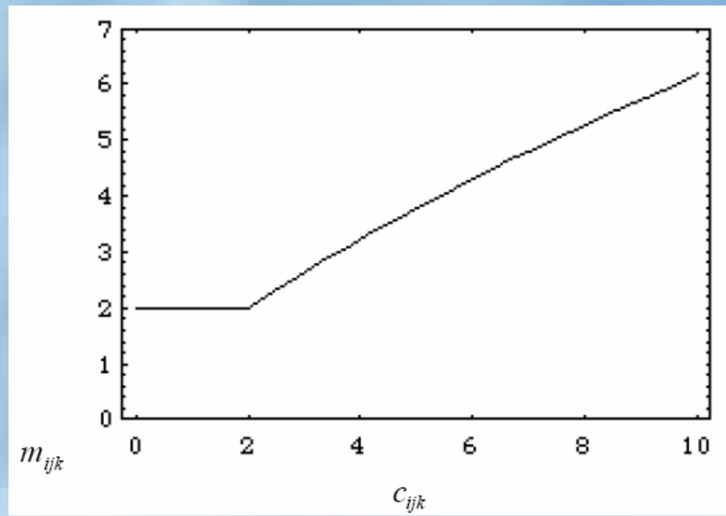
Contrast Masking

DCT coefficient

w_{ij} is in the range [0 1]

$$m_{ijk} = \max\left(t_{ijk}, \left|c_{ijk}\right|^{w_{ij}} t_{ijk}^{\,1-w_{ij}}\right)$$

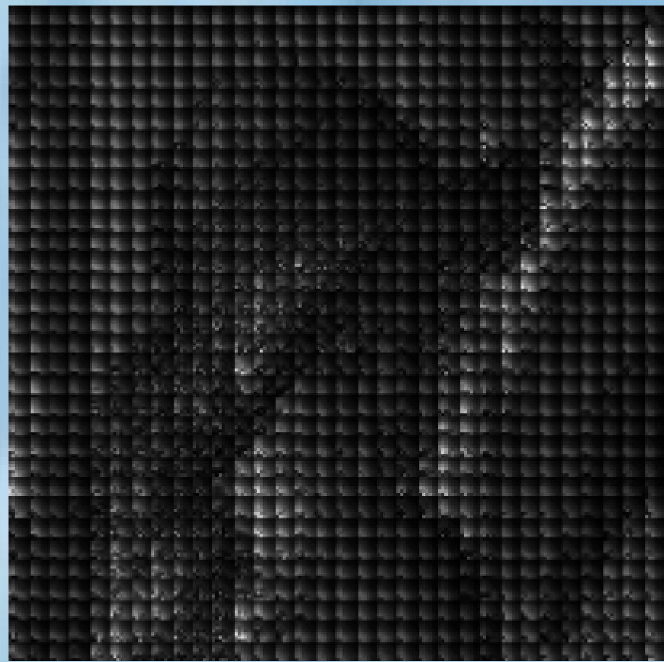
Absolute threshold



Masked threshold as a function of DCT coefficient for parameters w_{ij}=0.7, t_{ijk}=2

- w_{ij} may differ for each frequency, so we have a 8x8 matrix of exponents
  - w_{00} = 0 : Exclude DC from contrast masking
- w_{ij} = 0 : No constant masking
- w_{ij} = 1 : Weber Law (threshold is constant in log)

Contrast Masking



Masked threshold m_{ijk} for w_{ij}=0.7, a_T=0.649

Just-Noticeable Difference

- Express the magnitude of the signal in multiples of the threshold for the signal
  - Transfer all errors to a “common coin” of perceptual error

Just Noticeable Differences (JNDs)

Quantization error

$$d_{ijk} = \frac{e_{ijk}}{m_{ijk}}$$

Masked threshold that we have just computed

Error Pooling

- The visibility of many errors of varying magnitudes is not equal to the visibility of the largest error
- Pooling of errors occurs over both frequencies and blocks of the image

Spatial Error Pooling

- Use the β-norm (Minkowski metric):

β_s implements different types of pooling

JNDs for frequency *ij* over all blocks *k*. This is the perceptual error matrix.

$$p_{ij} = \left(\sum_k \left| d_{ijk} \right|^{\beta_s} \right)^{\frac{1}{\beta_s}}$$

- β_s = 1 : Linear summation of absolute values
- β_s = 2 : Energy or standard deviation
- β_s = ∞ : Maximum (only the largest error matters)
- In practice, β_s of about 4 had been observed

Frequency Error Pooling

- Use again the β-norm:

$$P = \left(\sum_{ij} {p_{ij}}^{\beta_f} \right)^{\frac{1}{\beta_f}}$$

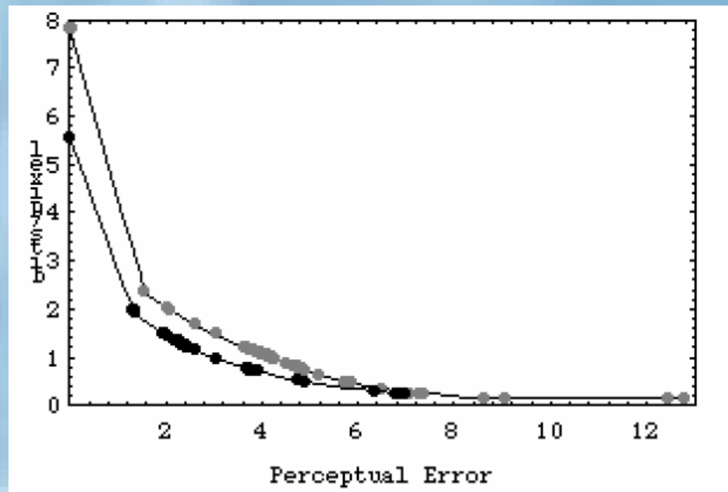
- We get single-valued perceptual error metric that we can used for optimization
  - Obtain minimum bit-rate for a given *P*
  - Obtain minimum *P* for a given bit-rate

Selectable Quality

- The quality factor that is commonly implemented in JPEG encoders allows an arbitrary bit-rate, but do not suggest optimum quality at that bit-rate.
- In our case, we assume that visual artifacts in different frequencies are independent
  - Optimization is individual for each frequency and is much simpler
  - Minimum bit-rate for a given $P = \Psi$ is achieved when all $p_{ij} = \Psi$
  - We achieve this by setting β_f to ∞ : P is given by $\max(p_{ij})$

Selectable Quality

- Bit-rate is a decreasing function of Ψ
  - Helps is yielding a given bit-rate

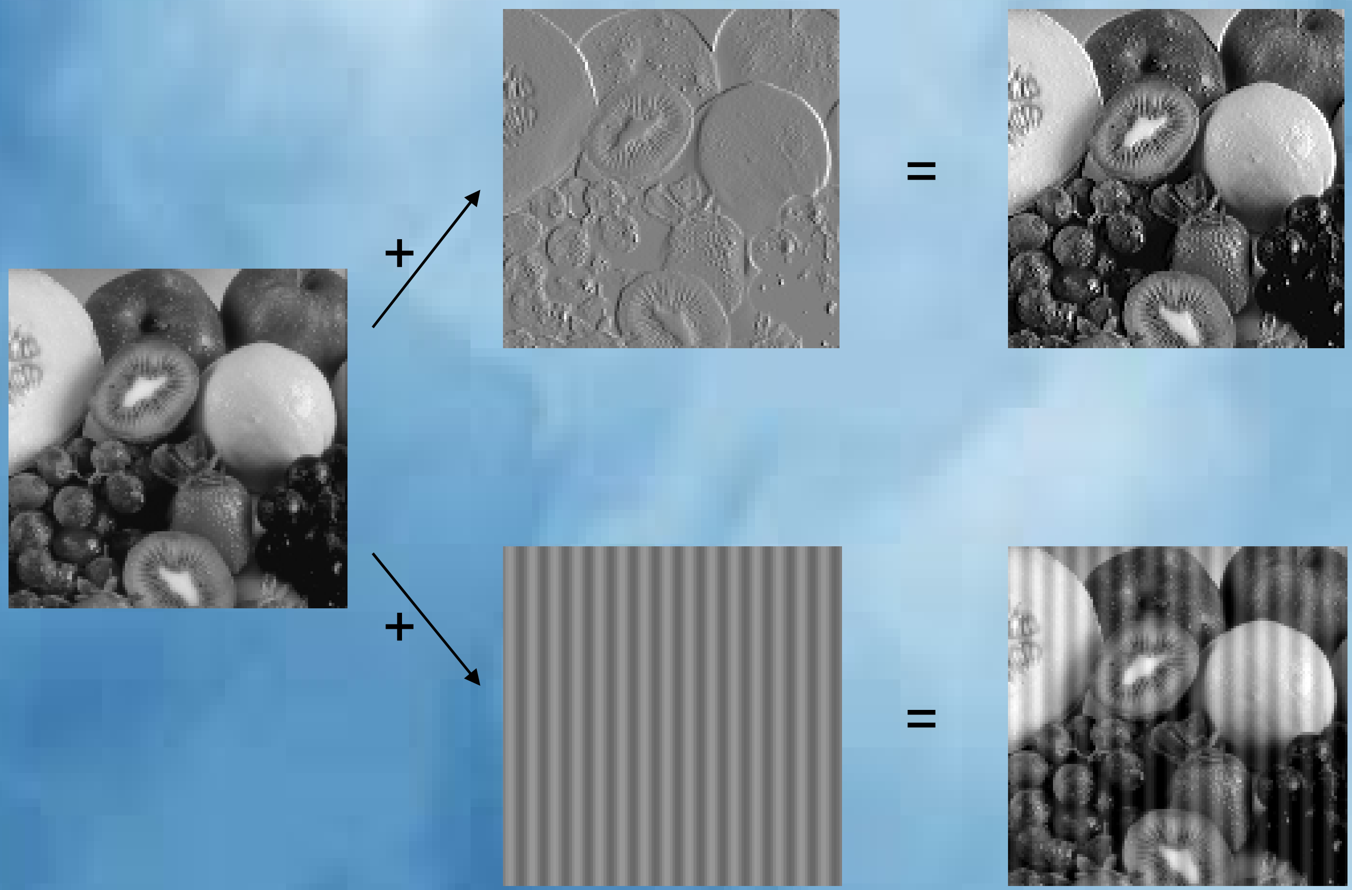


Bit-rate as a function of Ψ for Lena (lower curve) and Mandrill (upper curve) images.

- All optimization is done in the DCT domain
- Computationally modest
  - Estimation of the quantization matrix requires a maximum of 10 iterations (and probably less)
  - Each iteration consists of few simple operation on each DCT coefficient

DCTune defines a Perceptual Image Quality Metric

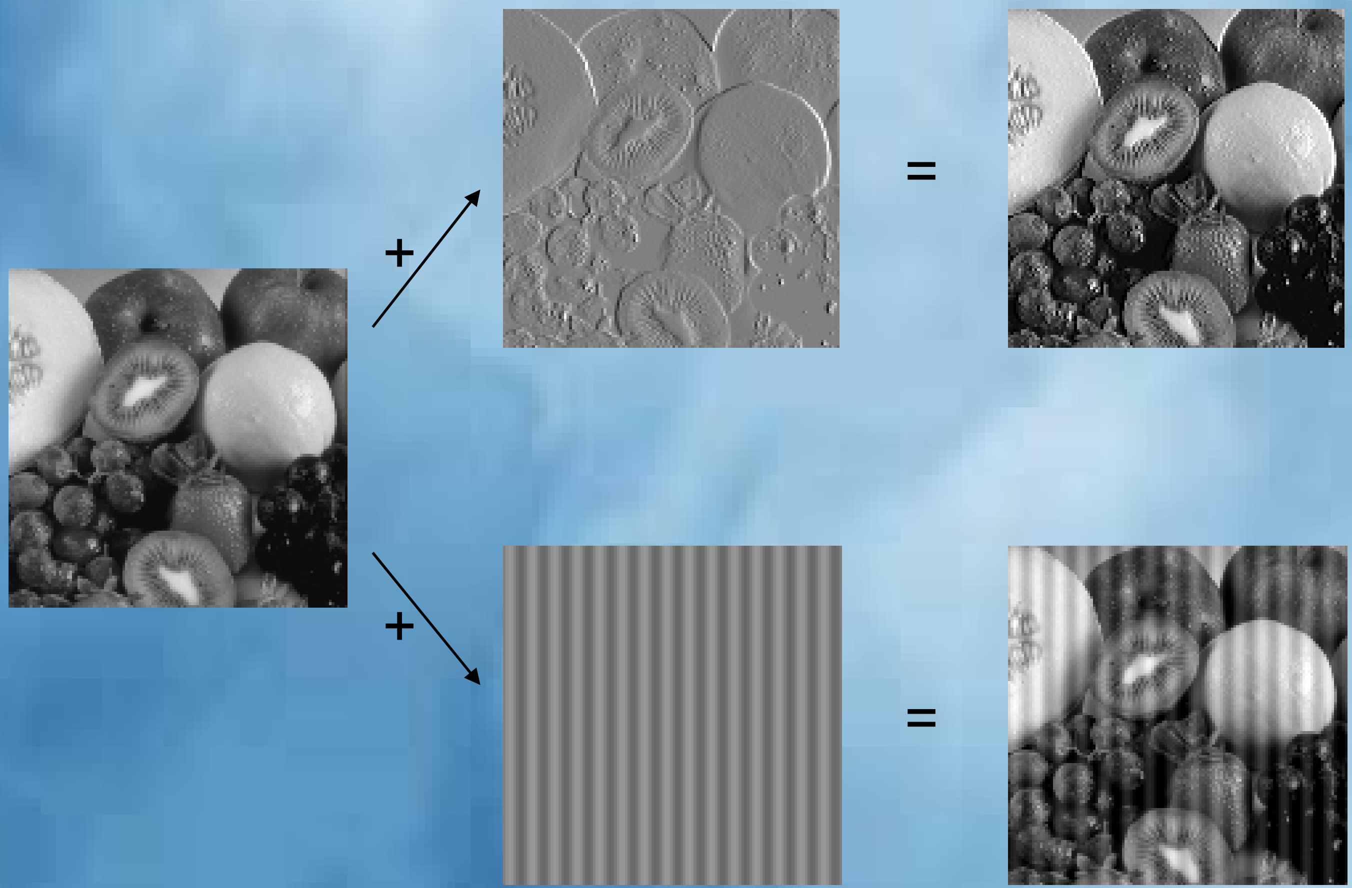
$\Delta e^2 = 4{,}882{,}611$



$\Delta e^2 = 4{,}882{,}611$

DCTune finds Quantization Matrix that minimizes Image quality difference between original and compressed.

$\Delta e^2 = 4{,}882{,}611$



$\Delta e^2 = 4{,}882{,}611$

And the Results…

IIP, 0.5 bits/pixel (1:16 compression ratio)

DCTune, 0.5 bits/pixel (1:16 compression ratio)

And the Results…

IIP, 0.25 bits/pixel (1:32 compression ratio)

DCTune, 0.25 bits/pixel (1:32 compression ratio)

What We're Going to Talk About

- Introduction to image compression
- The JPEG standard
- JPEG perceptual extensions
  - DCTune
  - DCTune in color
  - Adaptive DCTune
- Summary

DCTune in Color

Watson (1994)

- JPEG allows distinct quantization matrices for each color channel
- Algorithm is similar to “DCTune in grayscale”
  - We add an index Φ to each variable, describing the channel in the perceptual color space
  - Practically this has been tried extensively only with the YCbCr color space

Luminance Masking

- All three color channels are adjusted by the luminance channel coefficients

DC coefficient of block k for luminance channel

Control the degree of masking.

Threshold from naive approach

$$t_{ijk\phi} = t_{ij\phi} \left(\frac{c_{00kY}}{\overline{c}_{00Y}} \right)^{a_T}$$

DC coefficient of mean display luminance

Contrast Masking

- Assume that masking occurs only within a single block, frequency and color channel
  - This is wrong but gives only small errors

DCT coefficient

$w_{ij\Phi}$ is in the range [0 1]

$$m_{ijk\phi} = t_{ijk\phi} \max\left(1, \left|\frac{c_{ijk\phi}}{t_{ijk\phi}}\right|^{w_{ij\phi}}\right)$$

Absolute threshold according to luminance channel

Color Channels Error Pooling

- Error pooling is done as before over blocks and over frequencies
- We get one P_{Φ} variable for each color channel
- Finally the β-norm is used again over the color channels:

$$P = \left(\sum_{\phi} {p_{\phi}}^{\beta_c} \right)^{\frac{1}{\beta_c}}$$

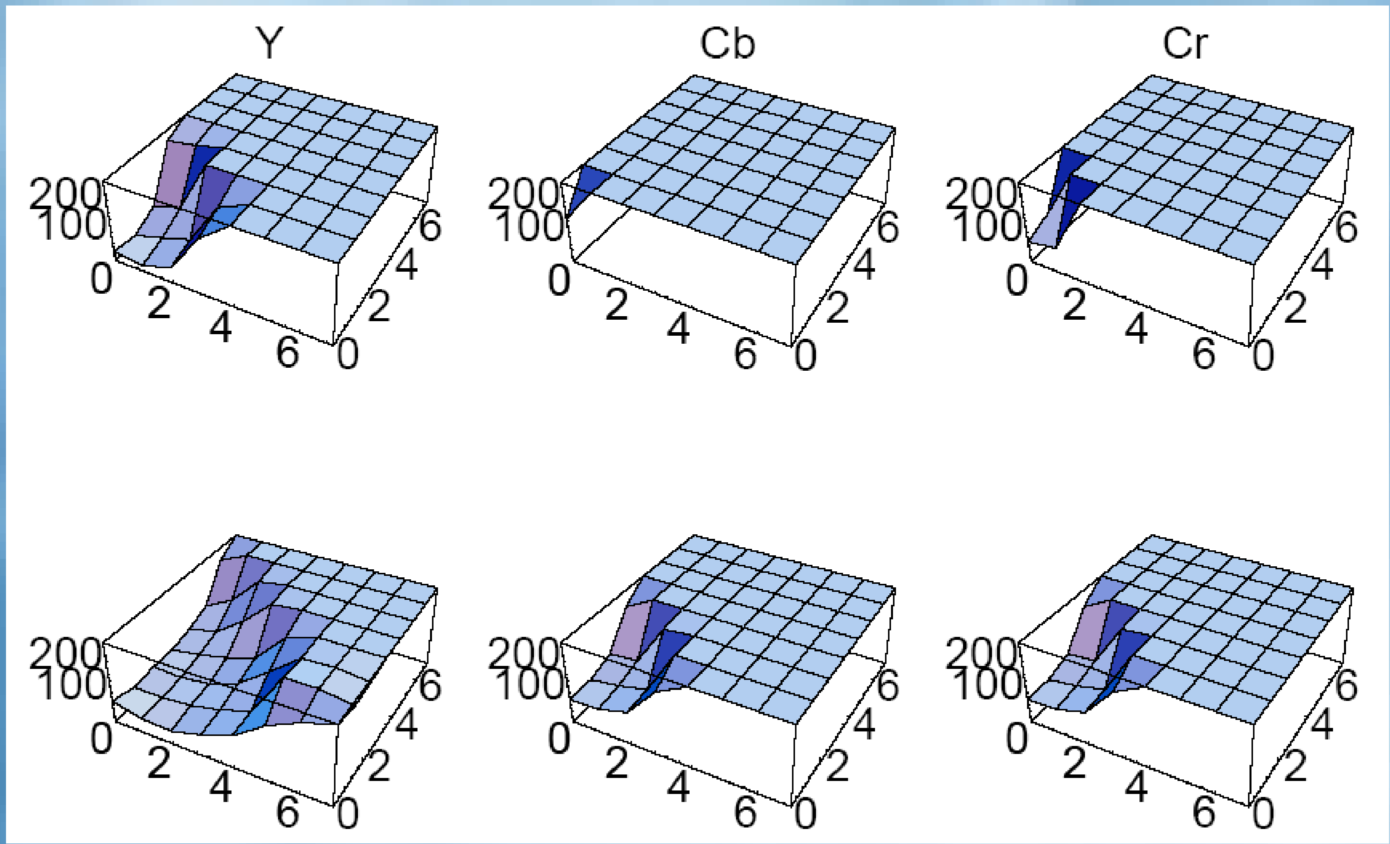
Selectable Quality

- β_c is assumed to be ∞
- The search space is much larger now – 255^3 possibilities
- Use hierarchical direct search algorithm
  - Each of the 3x64=192 quantization matrix entries is optimized separately

Example Quantization Matrices

DCTune

Standard JPEG



YCbCr color quantization matrices for 0.25 bits/pixel

Note: HIgher values = more compressed

And the Results…

Paint Shop Pro, 1:48 compression

DCTune, 1:48 compression ratio

- DCTune looks worse !!!

And the Results…

And the Results…

Paint Shop Pro, 1:48 compression

DCTune, 1:48 compression ratio

- DCTune looks worse again !!!
- Probably due to the fact that it optimizes to spatial frequencies but not to color matching

What We’re Going to Talk About

- Introduction to image compression
- The JPEG standard
- JPEG perceptual extensions
  - DCTune
  - DCTune in color
  - Adaptive DCTune
- Summary

Adaptive DCTune

Rosenholtz & Watson (1996)

- An extension to JPEG allows spatial adaptive quantization
  - Still one quantization matrix for each color channel
  - A different multiplier for the matrix can be specified for each block
  - This applies only to the AC coefficients
- A table of 31 different multipliers is used
  - A cost of 6 bits on coding a different multiplier from one block to the next
  - No cost if the multiplier stays the same

Proposed Solution

- Use the same error metrics as before
- Optimize the quantization matrix *Q* and the multipliers table *M* separately
  - Optimizing them together has a huge complexity
  - It looks reasonable that *Q* and *M* are independent
- First Optimize *Q* and after that optimize *M*
  - Do this the same way as before

Proposed Solution

- Problem: Bits spent of changing multipliers might be as high as 6% of the total bits
- Solution:
  - Lower a multiplier from a recommended value to that of the previous multiplier if the coarser quantization does not save at least 6 bits
  - Never raise the value of a multiplier

And the Results…

Original

Multipliers matrix *M* – lighter blocks will be more coarsely quantized

And the Results…

Non-adaptive DCTune, 1.02 bits/pixel

Adaptive DCTune, 1.01 bits/pixel

- Yields 22% reduction in bit-rate over non-adaptive coding according to the proposed perceptual error model

What We’re Going to Talk About

- Introduction to image compression
- The JPEG standard
- JPEG perceptual extensions
  - DCTune
  - DCTune in color
  - Adaptive DCTune
- Summary

Summary

- The proposed method computes a visually optimal quantization matrix for a given image
- For grayscale images, produce better results than image independent matrices
  - Adaptive version is even better
- For color images, doesn't perform well
- Provides a meaningful quality scale for JPEG compression

Future Directions

- JPEG2000 (Dec., 2000)
  - Uses discrete wavelet transform
  - Can usually compress by at least 20% more than JPEG
  - Many other advantages: handles well text, region of interest, error resilience tools…

References - JPEG

The JPEG Standard

- ISO/IEC DIS 10918-1 / ITU-T Recommrndation T.81: Digital Compression and Coding of Continuous-Tone Still Images. 1992.

Books

- Pennebaker B. William, Mitchell L. Joan, JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, 1993.
- Salomon, David. Data Compression – The Complete Reference, 3rd edition, Springer, 2004.
- Richardson E. G. Ian. Video Codec Design, Wiley, 2002.

References - JPEG

Papers

- Wallace, K. Gregory. The JPEG Still Picture Compression Standard, Commun. of the ACM. 34:4, 1991, pp.30–44.
- Furht B. A Survey of Multimedia Compression Techniques and Standards. Part I: JPEG Standard, Journal of Real-Time Imaging, vol. 1, no. 1, April 1995, pp. 49-67.

Web sites

- Official site of JPEG group: http://www.jpeg.org/
- JPEG FAQ: http://www.faqs.org/faqs/jpeg-faq/

References – Perceptual Extensions

Papers

- Watson, B. Andrew. DCTune: A Technique for Visual Optimization of DCT Quantization Matrices for Individual Images. Soc. for Information Display, Digest of Tech. Papers XXIV, 1993, pp. 946-949.
- Watson, B. Andrew. Visually Optimal DCT Quantization Matrices for Individual Images. Proc. Of IEEE Data Compression Conference, April 1993, pp. 178-187.
- Watson, B. Andrew. DCT Quantization Matrices Visually Optimized for Individual Images. Proc. of SPIE, Human Vision, Visual Processing, and Digital Display IV, 1993, pp. 202-216.
- Watson, B. Andrew. Perceptual Optimization of DCT Color Quantization Matrices. Proc. of IEEE Int. Conf. on Image Processing, 1994, pp. 100-104.
- Rosenholtz, Ruth and Watson B. Andrew. Perceptual Adaptive JPEG Coding. Proc. Of IEEE Int. Conf. on Image Processing, vol. 1, 1996, pp. 901-904.

References – Perceptual Extensions

Web sites

- DCTune utility 2.0: http://vision.arc.nasa.gov/dctune/
- Analysis of DCTune: http://ise.stanford.edu/class/psych221/projects/98/dctune/yuke/index.htm