Multimedia Systems and Applications

Image Compression - JPEG

James Wang

Drawn from "Fundamentals of Multimedia, Ze-Nian Li, Mark S, Drew

http://www.cs.sfu.ca/mmbook

Overview of JPEG

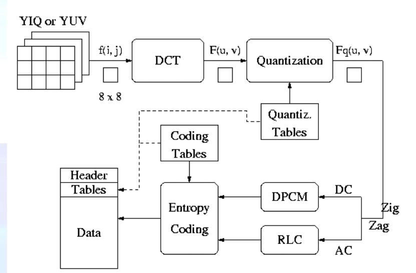
What is JPEG?

- **"Joint Photographic Expert Group". Voted as international standard in 1992.**
- **Works with colour and greyscale images, e.g., satellite, medical, ...**

Motivation

- **The *compression ratio* of lossless methods (e.g., Huffman, Arithmetic, LZW) is** not high enough **for image and video compression, especially when the distribution of pixel values is relatively flat.**
- **JPEG uses *transform coding*, it is largely based on the following observations:**
  - Observation 1: A large majority of useful image contents change relatively slowly across images, i.e., it is unusual for intensity values to alter up and down several times in a small area, for example, within an 8 x 8 image block. Translate this into the spatial frequency domain, it says that, generally, lower spatial frequency components contain more information than the high frequency components which often correspond to less useful details and noises.
  - Observation 2: Psychophysical experiments suggest that humans are more receptive to loss of higher spatial frequency components than loss of lower frequency components.

JPEG Encoding



JPEG Decoding

How to decode the JPEG Pictures?

Reverse the Process!

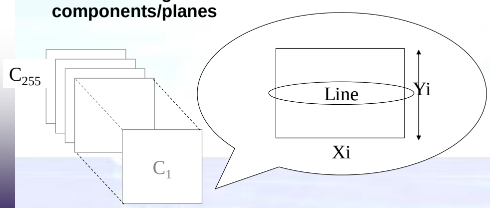
- **i.e. The flow arrow will point backwards**
- **This will imply that the transform function needs to have an equivalent *inverse.***

Image Compression Process

- **There are FOUR (4) modes:**
  - *Lossy sequential DCT (supported by all)*
  - *Extended lossy DCT (further enhancements)*
  - *Lossless mode*
  - *Hierarchical mode*
- **The baseline process takes the following techniques: *Block, MCU, FDCT, RLE and Huffman. We shall discuss the details later.***

Image Preparation

- A source image consists of 1..255 components/planes

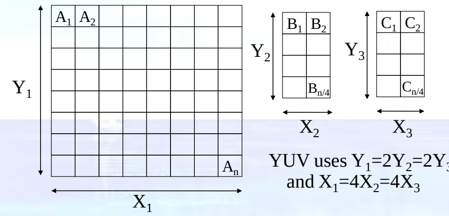


- RGB Color has its 3 components with equal resolution

7

Image Preparation

- The resolution of individual components may also be different!



8

Image Preparation

- Each image is represented by *n* bits with values in the range of 0 .. $(2^n - 1)$
- Lossy mode → 8 or 12 bits
- Lossless modes → 2 to 12 bits

9

Sampling factors

- X, Y are maximum value
- H_i = Horizontal sampling ratio,
- V_i = Vertical sampling ratio
- H_{max} = Horizontal sampling ratio,
- V_{max} = Vertical sampling ratio

$$X_i = X * H_i / H_{max}$$
$$Y_i = Y * V_i / V_{max}$$

10

Sampling factors

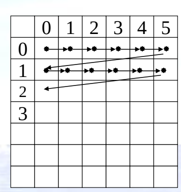
Example:

- Given X = 512, Y=512, $H_{max} = 4$ $V_{max} = 2$
- Level 0: $H_0 = 4$, $V_0=1$
  - $X_0 = X * H_0 / H_{max} = 512 * 4 / 4 = 512$
  - $Y_0 = Y * V_0 / V_{max} = 512 * 1 / 2 = 256$
- Level 1: $H_1 = 2$, $V_1 = 2$
  - $X_0 = X * H_0 / H_{max} = 512 * 2 / 4 = 256$
  - $Y_0 = Y * V_0 / V_{max} = 512 * 2 / 2 = 512$
- Level 2: $H_2 = 1$, $V_2 = 1$
  - $X_0 = X * H_0 / H_{max} = 512 * 1 / 4 = 128$
  - $Y_0 = Y * V_0 / V_{max} = 512 * 1 / 2 = 256$

11

Interleaving of data units

- Non-interleaved order of data

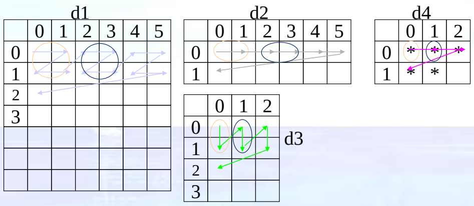


Strictly Left-to-Right and Top-to-Bottom

12

Interleaving of data units

- MCU (Minimum Coded Units) = combination of interleaved data units of different components.



d1: H=2, V=2;   d2: H=2, V=1;   d3: H=1,V=2;   d4: H=1; V=1

Interleaving of data units

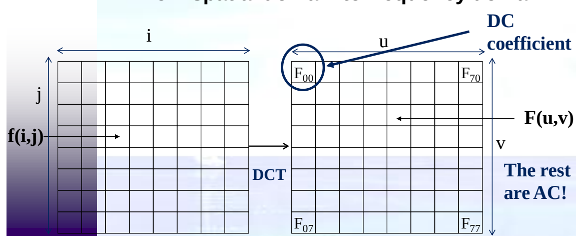
- The MCU in this example are:
- $MCU_1 = d^1_{00} d^1_{01} d^1_{10} d^1_{11} d^2_{00} d^2_{01} d^3_{00} d^3_{10} d^4_{00}$
- $MCU_1 = d^1_{02} d^1_{03} d^1_{12} d^1_{13} d^2_{02} d^2_{03} d^3_{01} d^3_{11} d^4_{01}$
- Note: The data units from the same component are shown with the same color.

Major Steps

1. DCT (Discrete Cosine Transformation)
2. Quantization
3. Zigzag Scan
4. DPCM on DC component
5. RLE on AC Components
6. Entropy Coding

Discrete Cosine Transform (DCT)

- From spatial domain to frequency domain



Discrete Cosine Transform (DCT)

- The transform function maps the pixel value at (i,j) in the range of [1,255] to [-128, 127]
- These 8 X 8 data units are defined by f(i,j) where i, j are in the range of 0 to 7
- F(0,0) is called the DC coefficient. Others are called AC coefficients.
- F(0,0) is a function that counts the occurrence/frequency that are zero in both directions.
- The value represents an average of the overall 8x8 input matrix.

Discrete Cosine Transform (DCT)

- Discrete Cosine Transform (DCT):

$$F(u,v) = \frac{\Lambda(u)\Lambda(v)}{4} \sum_{i=0}^{7}\sum_{j=0}^{7} \cos\frac{(2i+1)\cdot u\pi}{16} \cdot \cos\frac{(2j+1)\cdot v\pi}{16} \cdot f(i,j)$$

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & for\ \xi = 0 \\ 1 & otherwise \end{cases}$$

- Inverse Discrete Cosine Transform (IDCT):

$$\tilde{f}(i,j) = \frac{1}{4} \sum_{u=0}^{7}\sum_{v=0}^{7} \Lambda(u)\Lambda(v) \cos\frac{(2i+1)\cdot u\pi}{16} \cdot \cos\frac{(2j+1)\cdot v\pi}{16} \cdot F(u,v)$$

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & for\ \xi = 0 \\ 1 & otherwise \end{cases}$$

Discrete Cosine Transform (DCT)

- **Example:**

132	136	138	140	144	145	147	155
136	140	140	147	140	148	155	156
140	143	144	148	150	152	154	155
144	144	146	145	149	150	153	160
150	152	155	156	150	145	144	140
144	145	146	148	143	158	150	140
150	156	157	156	140	146	156	145
148	145	146	148	156	160	140	145

j (rows), i (columns)

DCT →
← IDCT

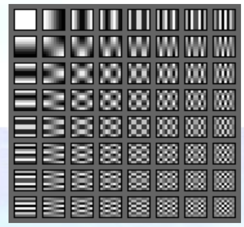
172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

v (rows), u (columns)

19

Discrete Cosine Transform (DCT)

- **The 64 (8 x 8) DCT basis functions:**



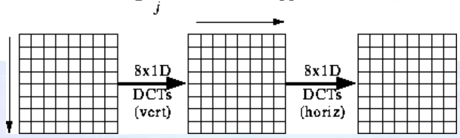
20

Discrete Cosine Transform (DCT)

- **Factoring reduces problem to a series of 1D DCTs:**

$$F[u,v] = \frac{1}{2}\sum_{i} A(u)\cos\frac{(2i+1)u\pi}{16}G[i,v]$$

$$G[i,v] = \frac{1}{2}\sum_{j} A(v)\cos\frac{(2j+1)v\pi}{16}f[i,j]$$



- **Most software implementations use fixed point arithmetic. Some fast implementations approximate coefficients so all multiplies are shifts and adds.**

21

Quantization

- **Formula:**

  F'[u, v] = round (F[u, v] / q[u, v]).

- **Why?**

  To reduce number of bits per sample

- **Example: 101101 = 45 (6 bits).**
  q[u, v] = 4 --> Truncate to 4 bits: 1011 = 11.
- **Quantization error is the main source of the Lossy Compression.**

22

Quantization

- Uniform Quantization

Each F[u,v] is divided by the same constant *N*.

- Non-uniform Quantization using Quantization Tables

  Eye is most sensitive to low frequencies (upper left corner), less sensitive to high frequencies (lower right corner)

23

Quantization

The Luminance Quantization Table q(u, v)

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

24

Quantization

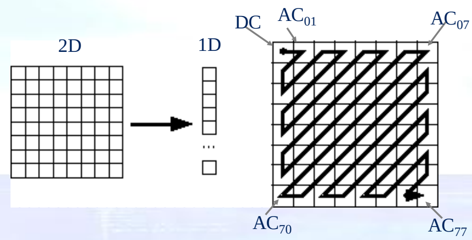
The *Chrominance Quantization Table* q(u, v)

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

The numbers in the above quantization tables can be scaled up (or down) to adjust the so called quality factor.
Custom quantization tables can also be put in image/scan header.

25

Zig-zag Scan



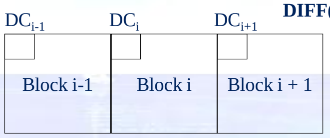
Why?

1. **To group low frequency coefficients in top of vector.**
2. **Maps 8 x 8 to a 1 x 64 vector**

26

Differential Pulse Code Modulation (DPCM) on DC component

- **DC component is large and varied, but often close to previous value.**
- **Encode the difference from previous 8 x 8 blocks → DPCM**



$$DIFF(i-1,i)=DC_{i-1}-DC_i$$

27

Run Length Encode (RLE) on AC components

- **1 x 64 vector has lots of zeros in it**
- **Keeps *skip* and *value*, where *skip* is the number of zeros and *value* is the next non-zero component.**
- **Send (0,0) as end-of-block sentinel value.**

28

Entropy Coding

- **Categorize DC values into SIZE (number of bits needed to represent) and actual bits.**

Size	BCD	Diff DC Coefficient Value
0	0000	0
1	0001	-1, 1
2	0010	-3, -2, 2, 3
3	0011	-7..-4, 4..7
4	0100	-15..-8, 8..15
5	0101	-31..-16, 16..31
6	0110	-63..-32, 32..63
7	0111	-127..-33, 33..127
8	1000	-255..-128, 128..255
9	1001	-511..-256, 256, 511
10	1010	-1023..-512, 512..1023

- ***Example:* if DC value is 4, 3 bits are needed. Send off SIZE as Huffman symbol, followed by actual 3 bits.**

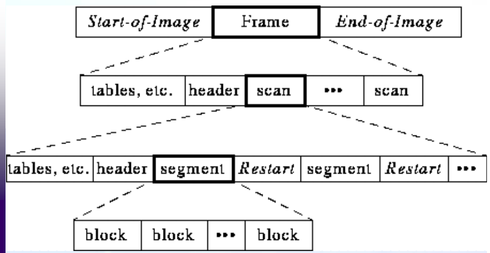
29

Entropy Coding

- **For AC components two symbols are used:**
  Symbol_1: (*skip, SIZE*),
  Symbol_2: actual bits.
- **Symbol_1 (*skip, SIZE*) is encoded using the Huffman coding, Symbol_2 is not encoded.**
- **Huffman Tables can be custom (sent in header) or default.**

30

A Glance at the JPEG Bitstream



31

A Glance at the JPEG Bitstream

- **A "Frame" is a picture, a "scan" is a pass through the pixels (e.g., the red component), a "segment" is a group of blocks, a "block" is an 8 x 8 group of pixels.**
- Frame header:
  - sample precision
  - (width, height) of image
  - number of components
  - unique ID (for each component)
  - horizontal/vertical sampling factors (for each component)
  - quantization table to use (for each component)
- Scan header
  - Number of components in scan
  - component ID (for each component)
  - Huffman table for each component (for each component)
- Misc. (can occur between headers)
  - Quantization tables
  - Huffman Tables
  - Arithmetic Coding Tables
  - Comments
  - Application Data

32

The Four JPEG Modes (Revisited)

The four JPEG modes:

- **Sequential Mode**
- **Loss-less Mode**
- **Progressive Mode**
- **Hierarchical Mode**

  - Note: In "Motion JPEG", Sequential JPEG is applied to each image in a video.

33

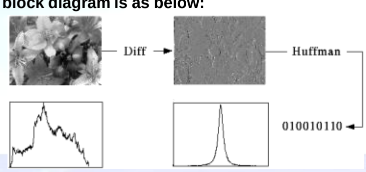
Sequential Mode JPEG

- **Each image component is encoded in a single left-to-right, top-to-bottom scan. *Baseline Sequential Mode*, the one that we described above, is a simple case of the Sequential mode:**
- **It supports only 8-bit images (not 12-bit images)**
- **It uses only Huffman coding (not Arithmetic coding)**

34

Lossless Mode JPEG

- **A special case of the JPEG where indeed there is no loss. Its block diagram is as below:**



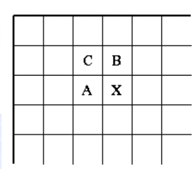
- **It does not use DCT-based method! Instead, it uses a *predictive* (differential coding) method:**
  - A predictor combines the values of up to three neighbouring pixels (not blocks as in the Sequential mode) as the predicted value for the current pixel, indicated by "X" in the figure on next page.
  - The encoder then compares this prediction with the actual pixel value at the position "X", and encodes the difference (prediction residual) losslessly.

35

Lossless Mode JPEG

- **It can use any one of the following seven predictors :**

•Predictor	•Prediction
•1	•A
•2	•B
•3	•C
•4	•A + B - C
•5	•A + (B - C) / 2
•6	•B + (A - C) / 2
•7	•(A + B) / 2



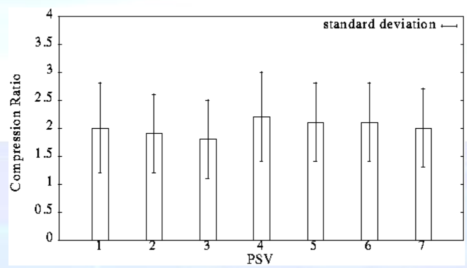
36

Lossless Mode JPEG

- **Since it uses only previously encoded neighbours, the very first pixel I(0, 0) will have to use itself. Other pixels at the first row always use P1, at the first column always use P2.**
- **Effect of Predictor (test result with 20 images is shown on next page)**
- Note: **"2D" predictors (4-7) always do better than "1D" predictors.**

37

Lossless Mode JPEG



38

Lossless Mode JPEG

- Comparison with Other Lossless Compression Programs compression ratio):

Compression Program	Compression Ratio Lena	football	F-18	flowers
lossless JPEG	**1.45**	**1.54**	**2.29**	**1.26**
optimal lossless JPEG	**1.49**	**1.67**	**2.71**	**1.33**
compress (LZW)	**0.86**	**1.24**	**2.21**	**0.87**
gzip (Lempel-Ziv)	**1.08**	**1.36**	**3.10**	**1.05**
gzip -9 (optimal Lempel-Ziv)	**1.08**	**1.36**	**3.13**	**1.05**
pack (Huffman coding)	**1.02**	**1.12**	**1.19**	**1.00**

39

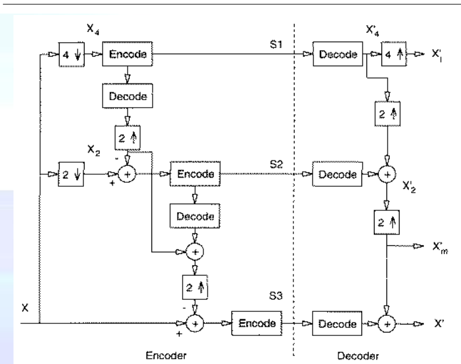
Progressive Mode JPEG

- Goal: display low quality image and successively improve.
- Two ways to successively improve image:

1. *Spectral selection*: Send DC component and first few AC coefficients first, then gradually some more ACs.
2. *Successive approximation*: send DCT coefficients MSB (most significant bit) to LSB (least significant bit).
   (Effectively, it is sending quantized DCT coefficients frist, and then the difference between the quantized and the non-quantized coefficients with finer quantization stepsize.)

40

Hierarchical Mode JPEG



41

Hierarchical Mode JPEG

(a) **Down-sample by factors of 2 in each dimension, e.g., reduce 640 x 480 to 320 x 240**
(b) **Code smaller image using another JPEG mode (Progressive, Sequential, or Lossless).**
(c) **Decode and up-sample encoded image**
(d) **Encode difference between the up-sampled and the original using Progressive, Sequential, or Lossless.**

- **Can be repeated multiple times.**
- **Good for viewing high resolution image on low resolution display.**
- **A Three-level Hierarchical JPEG Encoder (From *V. Bhaskaran and K. Konstantinides, "Image and Video Compression Standards: Algorithms and Architectures", 2nd ed., Kluwer Academic Publishers, 1997.*)**

42