JPEG Algorithm for Full-Color Still Image Compression

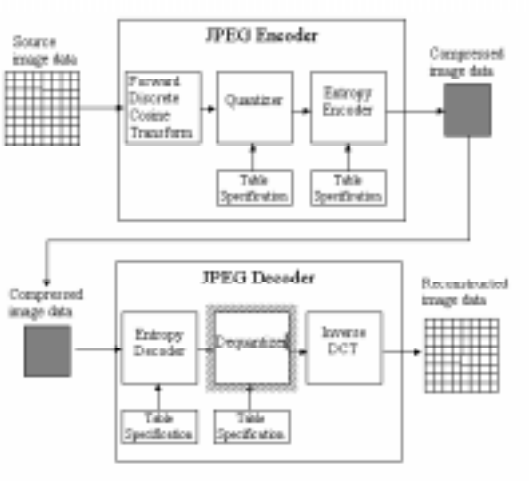
- average compression ratio 15:1
- 4 modes of operation
  - *i. sequential DCT-based encoding*
    - encoded in a single left-to-right, top-to-bottom scan
  - *ii. progressive DCT-based encoding*
    - encoded in multiple scans, in order to produce a quick, rough decoded image when the transmission time is long
  - *iii. lossless encoding*
    - encoded to guarantee the exact reproduction
  - *iv. hierarchical encoding*
    - encoded in multiple resolutions

JPEG CODEC

- JPEG sequential encoder and decoder

JPEG Encoder

- consist of three main blocks
  1. forward discrete cosine transform (FDCT) block
  2. quantizer
  3. entropy encoder



- the input to the encoder is shifted
  $[0, 2^P-1] \rightarrow [-2^{P-1}, 2^{P-1}-1]$

- the source image is divided into 8x8 block, and transformed into the frequency domain using the FDCT

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

where

$$C(u) = \begin{cases} 1/\sqrt{2} & for \quad u = 0 \\ 1 & for \quad u > 0 \end{cases}$$

- F(0,0) : DC coefficient
  the remaining 63 coefficients : AC coefficients
- for a grayscale image,
  DCT coefficients $\in$ [-1024, 1023]
- fast DCT algorithms are available
- most spatial frequencies have zero or near-zero values

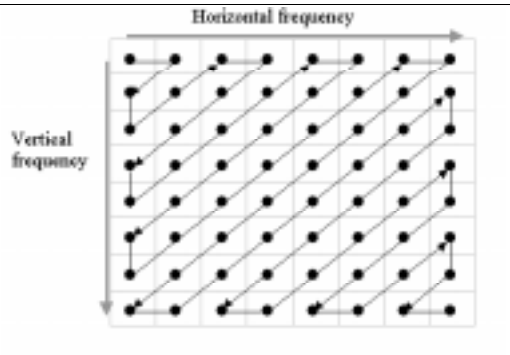
- all 64 DCT coefficient are quantized using a 64-element quantization table
  $\Rightarrow$ discard information which is not visually significant

$$F_q(u,v) = Round\left[\frac{F(u,v)}{Q(u,v)}\right]$$

where Q(u,v) are quantization coefficients specified by a quantization table

8	6	5	8	12	20	26	30
6	6	7	10	13	29	30	28
7	7	8	12	20	29	35	28
7	9	11	15	26	44	40	31
9	11	19	28	34	55	52	39
12	18	28	32	41	52	57	46
25	32	39	44	52	61	60	51
36	46	48	49	56	50	52	50

- after quantization, the 63 AC coefficients are ordered into the "zig-zag" sequence (for entropy encoding)



- the probability of coefficients being zero is an increasing monotonic function of the index
- the dc coefficients are encoded using the predictive coding techniques
  ∵ there is usually a strong correlation between dc coefficients of adjacent 8x8 blocks
- *entropy coding* : provide additional compression by encoding the quantized DCT coefficients into more compact form
- specified two entropy methods:
  Huffman coding and *arithmetic coding*

- Huffman encoder converts the DCT coefficients using 2 steps:

1. forming intermediate symbol sequence
   - each ac coefficient is represented by a pair of symbols:
     i. symbol-1 (RUNLENGTH, SIZE),
     ii. symbol-2 (AMPLITUDE)

   RUNLENGTH = the number of consecutive zero-valued ac coefficients preceding the nonzero ac coefficient ∈ [0,15]

   SIZE = the number of bits used to encode AMPLITUDE ∈ [0 to 10 bits]

   AMPLITUDE = in the range of [-1023, 1024]

   e.g. if the sequence of ac coefficients is
   0, 0, 0, 0, 0, 0, 476
   ⇒ (6, 9)(476)

   - if RUNLENGTH > 15,
     symbol-1 (15,0) ⇒ RUNLENGTH = 16
     e.g. (15,0)(15,0)(7,4)(12)
     ⇒ RUNLENGTH = 39
     SIZE = 4
     AMPLITUDE = 12
   - (0,0) ⇒ *End Of Block* (EOB)

   - for dc coefficients:
     symbol-1 (SIZE)
     symbol-2 (AMPLITUDE)
     the range of dc coefficients = [-2048, 2047]

2. converting intermediate symbol sequence into binary sequence using Huffman tables
   - symbols are replaced with variable length codes: dc coefficient, then ac coefficients
   - each symbol-1 encoded with a *Variable-Length Code* (VLC)
   - symbol-2 are encoded using *a Variable-Length Integer* (VLI) code
     e.g. (1,4)(12) → (1111101011100)

Size	Amplitude range
1	(-1,1)
2	(-3,-2)(2,3)
3	(-7..-4)(4..7)
4	(-15..-8)(8..15)
5	(-31..-16)(16..31)
6	(-63..-32)(32..63)
7	(-127..-64)(64..127)
8	(-255..-128)(128..255)
9	(-511..-256)(256..511)
10	(-1023..-512)(512..1023)

JPEG Decoder

- all the steps from the encoding process are inversed and implemented in reverse order
- IDCT equation:

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

where $C(u) = \begin{cases} 1/\sqrt{2} & for \quad u=0 \\ 1 & for \quad u>0 \end{cases}$

Compression Measures

- *Compression Ratio*, C_R

$$C_R = \frac{original\ data\ size}{compressed\ data\ size}$$

- there is a trade-off between the compression ratio and the picture quality
- a measure for the quality of picture:

  Number of bits/pixel,

$$N_b = \frac{encoded\ number\ of\ bits}{number\ of\ pixels}$$

N_b (bits/pixel)	Picture Quality
0.25-0.50	Moderate to good quality
0.50-0.75	Good to very good quality
0.75-1.00	Excellent quality
1.50-2.00	Usually indistinguishable from the original

- another statistical measure:

  Root Mean Square (RMS) error

$$RMS = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(X_i - \hat{X}_i)^2}$$

Sequential JPEG Encoding Example

Encoding a Single Block

(a) Original 8x8 block

140	144	147	140	140	155	179	175
144	152	140	147	140	148	167	179
152	155	136	167	163	162	152	172
168	145	156	160	152	155	136	160
162	148	156	148	140	136	147	162
147	167	140	155	155	140	136	162
136	156	123	167	162	144	140	147
148	155	136	155	152	147	147	136

(b) Shifted block

12	16	19	12	11	27	51	47
16	24	12	19	12	20	39	51
24	27	8	39	34	34	24	44
40	17	28	32	27	27	8	32
34	20	28	20	8	8	19	34
19	39	12	27	12	12	8	34
8	28	-5	39	16	16	12	19
20	27	8	27	19	19	19	8

(c) Block after FDCT

185	-17	14	-8	23	-9	-13	-18
20	-34	26	-9	-10	10	13	6
-10	-23	-1	6	-18	3	-20	0
-8	-5	14	-14	-8	-2	-3	8
-3	9	7	1	-11	17	18	15
3	-2	-18	8	8	-3	0	-6
8	0	-2	3	-1	-7	-1	-1
0	-7	-2	1	1	4	-6	0

(d) Quantization table (quality=2)

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

- quantization table:
  $Q[i][j] = 1+[(1+i+j)\times quality]$

(e) Block after quantization

61	-3	2	0	2	0	0	-1
4	-4	2	0	0	0	0	0
-1	-2	0	0	-1	0	-1	0
0	0	1	0	0	0	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

(f) Zig-zag sequence

61, -3,4,-1,-4,2,0,2,-2,0,0,0,0,0,2,0,0,0,1,0,0,0,
0,0,0,-1,0,0,-1,0,0,0,0,0,-1,0,0,0,0,0,0,0,-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

(g) Intermediate symbol sequence

(6)(61),(0,2)(-3),(0,3)(4),(0,1)(-1),(0,3)(-4),
(0,2)(2), (1,2)(2),(0,2)(-2),(5,2)(2),(3,1)(1),
(6,1)(-1),(2,1)(-1),(4,1)(-1),(7,1)(-1),(0,0)

(h) Encoded bit sequence (total 98 bits)

(110)(111101),(01)(00),(100)(100),(00)(0),
(100)(011),(01)(10),(11011)(10),(01)(01),
(01)(01),(11111110111)(10),(111010)(1),
(1111011)(0),(11100)(0),(111011)(0),
(11111010)(0),(1010)

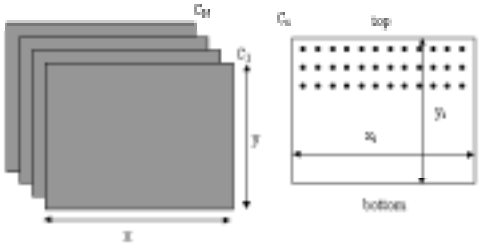
(RUNLENGTH,SIZE)	Code Word
(0,0) EOB	1010
(0,1)	00
(0,2)	01
(0,3)	100
(1,2)	11011
(2,1)	11100
(3,1)	111010
(4,1)	111011
(5,2)	11111110111
(6,1)	1111011
(7,1)	11111010

$$C_R = \frac{original\ data\ size}{compressed\ data\ size} =$$

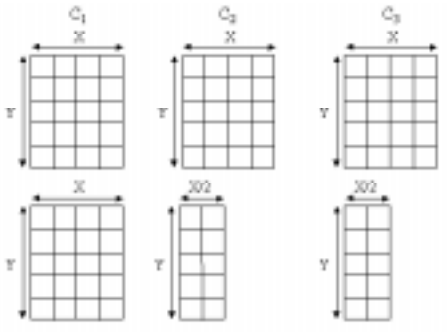
$$N_b = \frac{encoded\ number\ of\ bits}{number\ of\ pixels} =$$

JPEG Compression of Color Images

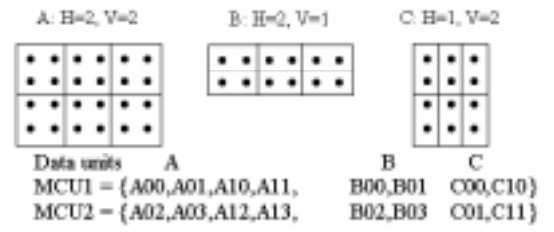
- the sequential JPEG algorithm can be easily expanded for multiple-component images
- the JPEG source image model consists of 1 to 255 image components, called color or spectral bands



- each component may have a different number of pixels in the horizontal (x_i) and vertical (y_i) axis

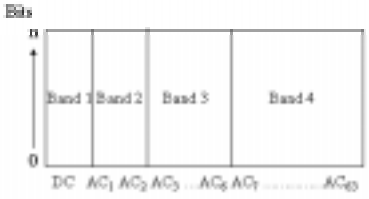


- the color components can be processed in two ways
  1. *non-interleaved data ordering*
     processing is performed component by component from left-to-right and top-to-bottom
  2. *interleaved data ordering*
     different components are combined into *Minimum Coded Units* (MCUs)

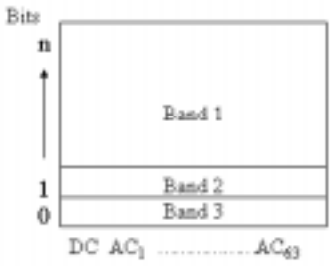


Progressive JPEG Compression

- produce quickly a rough image, and then improve its quality using multiple scans
- the progressive JPEG mode of operation produces a sequence of scans, each scan coding a subset of DCT coefficients
  $\Rightarrow$ must have an additional buffer
- can be achieved using three algorithms:
  1. *progressive spectral selection algorithm*
     - the DCT coefficients are grouped into several spectral bands
     - typically, low-frequency DCT coefficients bands are sent first, and then higher-frequency coefficients
     - for example:
       band 1: DC coefficient only
       band 2: AC_1 and AC_2 coefficients
       band 3: AC_3, AC_4, AC_5, AC_6 coefficients
       band 4: AC_7, . . ., AC_{63} coefficients



  2. *progressive successive approximation algorithm*
     - all DCT coefficients are sent first with lower precision, and then refined in later scans
     - for example:
       band 1: all DCT coefficients divided by 4
       band 2: all DCT coefficients divided by 2
       band 3: all DCT coefficients at full resolution



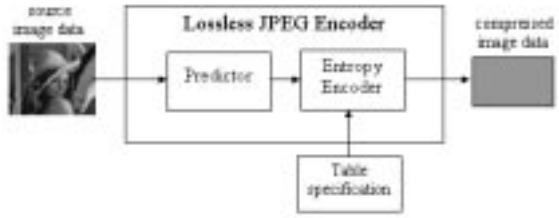
  3. *combined progressive algorithm*
     - combine both spectral selection and successive approximation algorithms

Sequential Lossless JPEG Compression

- a simple predictive compression algorithm is used instead of the DCT-based technique

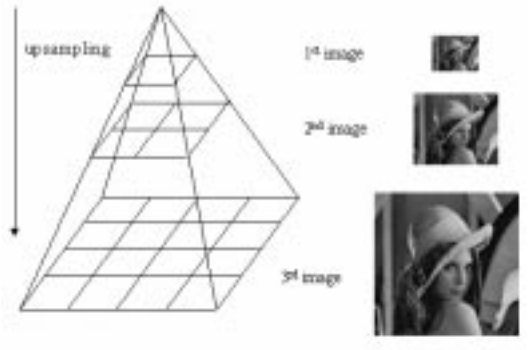
$$\hat{X} = f(A, B, C)$$
$$\Delta X = X - \hat{X}$$

ΔX is encoded using the Huffman coding



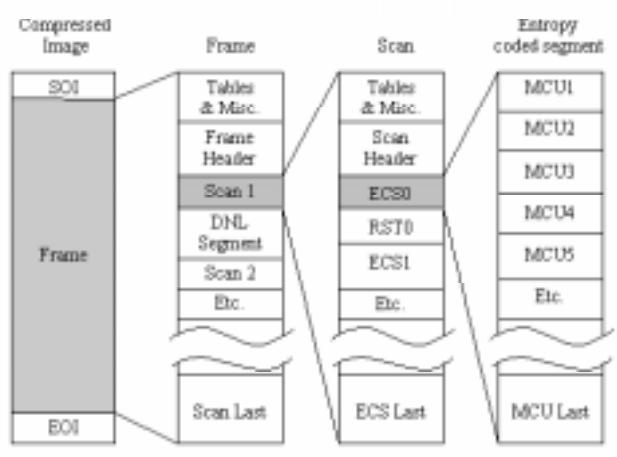
Hierarchical JPEG Compression

- offer a progressive representation of a decoded image similar to progressive JPEG, but also provide encoded images at multiple resolution
- create a set of compressed images beginning with small images, and then continuing with images with increased resolutions: *downsampling*, or *pyramidal coding*



Compressed Data Format

- various markers and parameters make up the baseline DCT-based JPEG format



- all markers are in the form of a 2-byte code:a 0×FF byte followed by a byte that is not equal to 0×FF or zero