JPEG Compression Standard

- Joint Photographic Experts Group (JPEG)
- The first international image compression standard
- Both an ISO standard and a CCITT Recommendation
  - ISO IS 10918-1
  - CCITT T.81
- 10:1 to 50:1 compression without visibly affecting image quality
- A DCT-based block coding algorithm

  energy較集中在low-freq的coefficient
- Base algorithm of "motion JPEG"
- Codec = encoder + decoder

  所有define的standard都是指decoding standard

Observations

- The design of the DCT transform coding method in JPEG is based on 3 major observations

Observation 1: Useful image contents change relatively slowly across the image, i.e., it is unusual for intensity values to vary widely several times in a small area, for example, within an 8×8 image block.

Much of the information in an image is repeated, hence there is "spatial redundancy."

Observation 2: Psychophysical experiments suggest that humans are much less likely to notice the loss of very high spatial frequency components than the loss of lower frequency components.

The spatial redundancy can be reduced by reducing the high spatial frequency contents. 犧牲 high freq 部份

Observation 3: Visual acuity (accuracy in distinguishing closely spaced lines) is much greater for luminance ("black and white") than for color.

Chroma subsampling (4:2:0) is used in JPEG.

JPEG是 color blind (RGB都做一樣的運算)

References

- JPEG: Still Image Data Compression Standard, by Pennebaker and Mitchell

- Greg Wallace, "The JPEG still picture compression standard," CACM, Apr. 1991

4 Modes of Operations

- Sequential ("baseline JPEG")

  Each image component is encoded in a single left-to-right, top-to-bottom scan.

- Progressive

  The image is encoded in multiple scans for applications in which transmission time is long, and the viewer prefers to watch the image being built up in multiple coarse-to-clear passes.

- Lossless

  Each pixel value is exactly reconstructed without loss

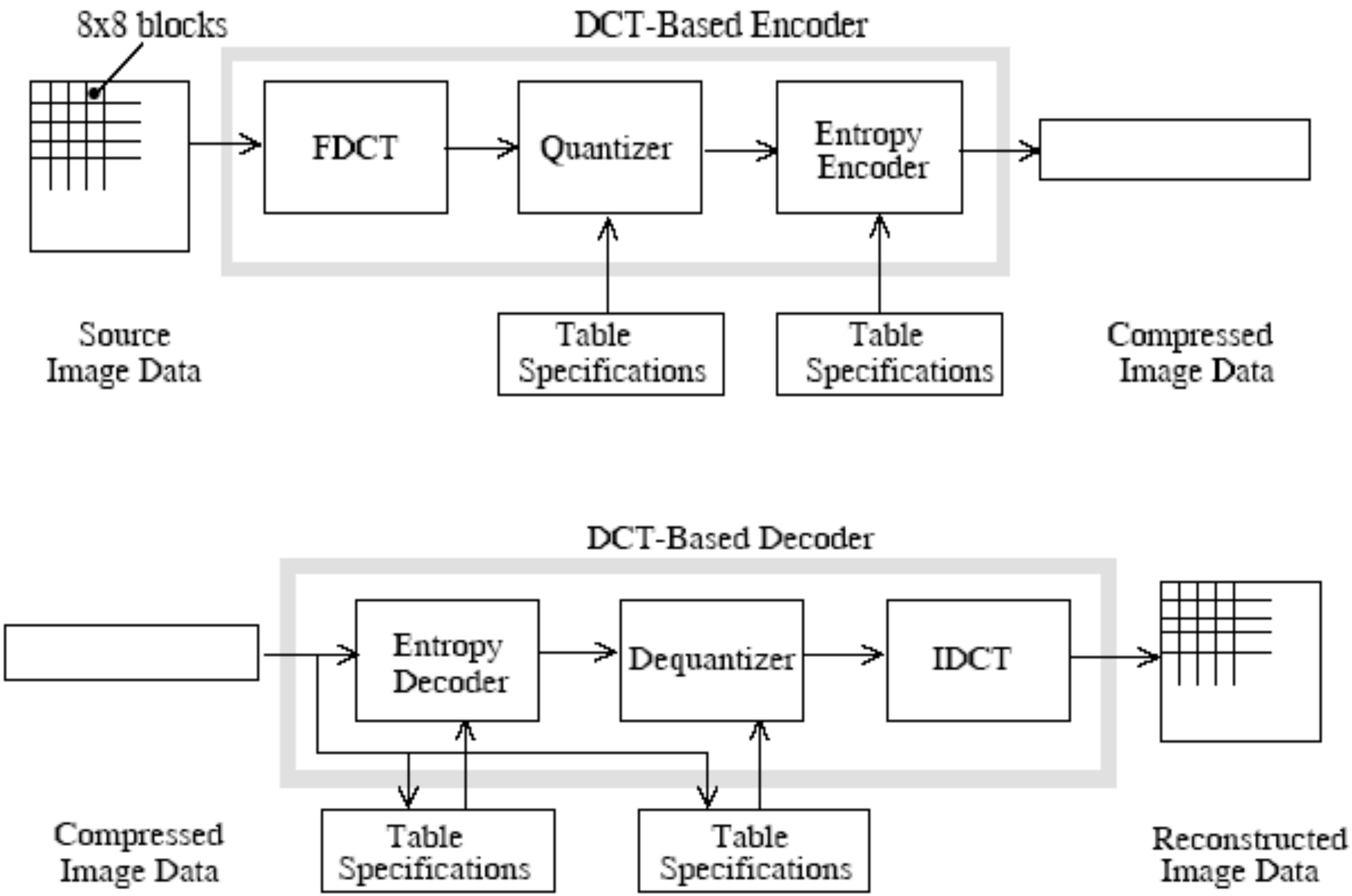
- Hierarchical

  The image is encoded at multiple resolutions so that lower-resolution versions may be accessed without first having to decompress the image at its full resolution.

  含蓋不同resolution

  ∴可依照不同人對resolution的需求，傳送給他們

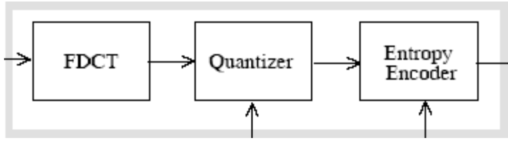
System Block Diagram



Main Steps in JPEG Compression

- DCT on image blocks
- Quantization
- Zig-zag ordering and run-length encoding
- Entropy coding

8x8 FDCT and IDCT



- At the input to the encoder the source image samples are
  - decomposed into 8x8 blocks
  - shifted from unsigned integer representation with range [0, 255] to signed integer with range [-128, 127]
- FDCT

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

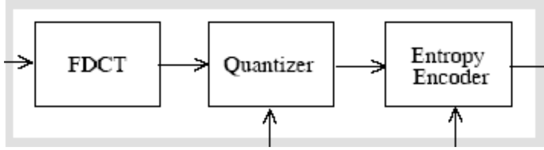
- IDCT

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

where: $C(u),\ C(v) = 1/\sqrt{2}$ for $u, v = 0$;

$C(u),\ C(v) = 1$ otherwise.

Quantization



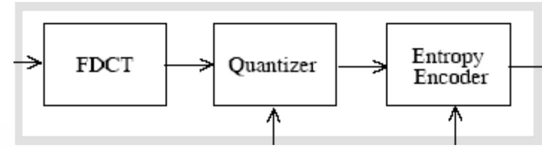
- To discard information that is not visually significant
- Quantization is defined as division of each DCT coefficient by its corresponding quantizer step size, followed by rounding to the nearest integer.

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

- Inverse quantization

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

Quantization Tables



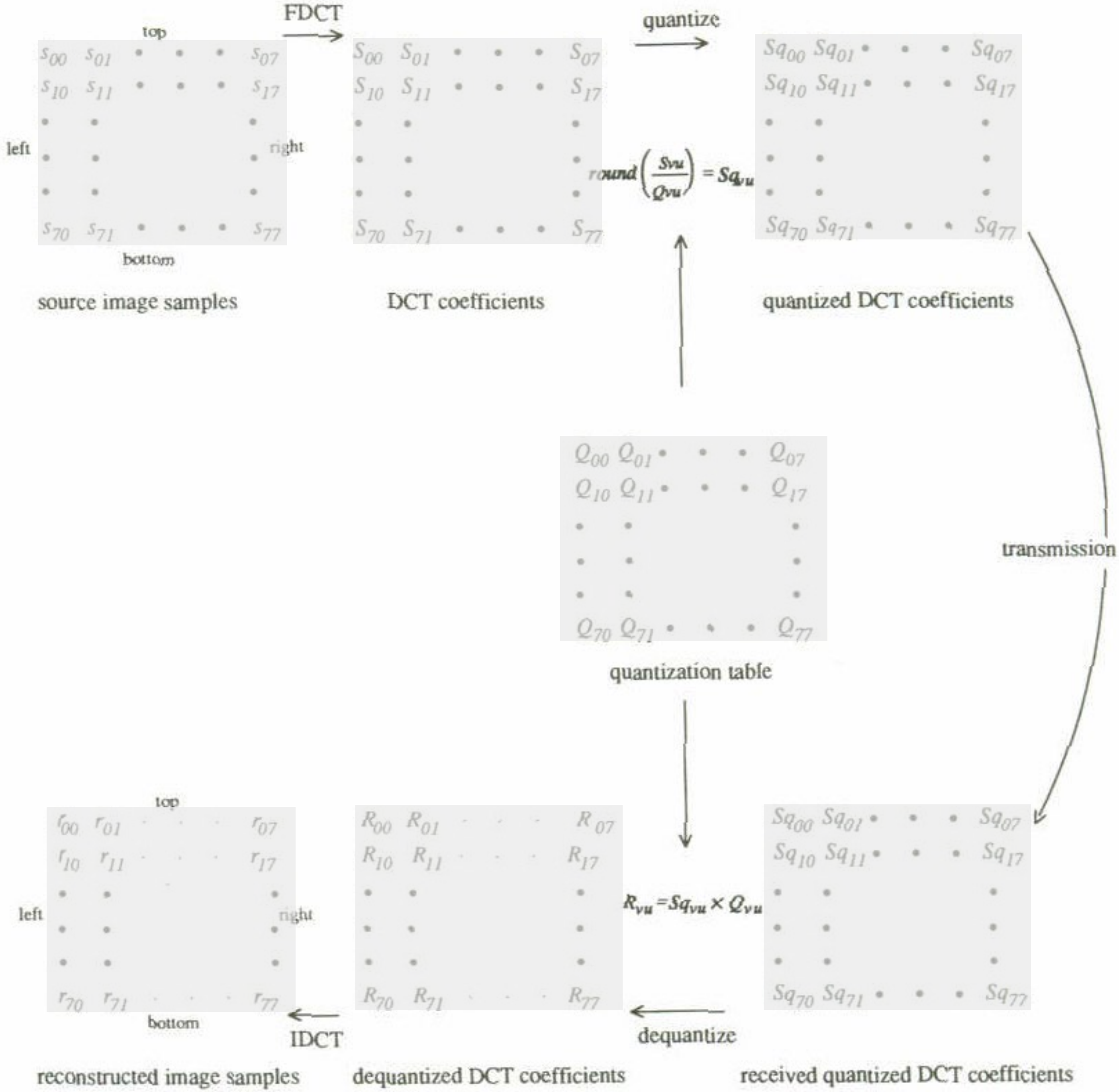
- In ISO 10918-1, the two tables are provided as examples only, but not as a requirement
- Based on psychovisual thresholding.
- Derived empirically using luminance and chrominance and 2:1 horizontal subsampling

Table K.1 Luminance quantization table

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

Table K.2 Chrominance quantization table

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


FDCT
quantize
top
left
right
bottom
round(Svu/Qvu) = Sqvu
source image samples
DCT coefficients
quantized DCT coefficients
transmission
quantization table
Rvu = Sqvu × Qvu
IDCT
dequantize
reconstructed image samples
dequantized DCT coefficients
received quantized DCT coefficients

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i, j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$\boldsymbol{F(u, v)}$

JPEG compression for a smooth image block

那些小數值的high freq部份做完decomposition就都被去掉了

$$\begin{matrix} 32 & 6 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$\hat{F}(u, v)$

$$\begin{matrix} 512 & 66 & -10 & 0 & 0 & 0 & 0 & 0 \\ -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 16 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$\tilde{F}(u, v)$

$$\begin{matrix} 199 & 196 & 191 & 186 & 182 & 178 & 177 & 176 \\ 201 & 199 & 196 & 192 & 188 & 183 & 180 & 178 \\ 203 & 203 & 202 & 200 & 195 & 189 & 183 & 180 \\ 202 & 203 & 204 & 203 & 198 & 191 & 183 & 179 \\ 200 & 201 & 202 & 201 & 196 & 189 & 182 & 177 \\ 200 & 200 & 199 & 197 & 192 & 186 & 181 & 177 \\ 204 & 202 & 199 & 195 & 190 & 186 & 183 & 181 \\ 207 & 204 & 200 & 194 & 190 & 187 & 185 & 184 \end{matrix}$$

$\tilde{f}(i, j)$

$$\begin{matrix} 1 & 6 & -2 & 2 & 7 & -3 & -2 & -1 \\ -1 & 4 & 2 & -4 & 1 & -1 & -2 & -3 \\ 0 & -3 & -2 & -5 & 5 & -2 & 2 & -5 \\ -2 & -3 & -4 & -3 & -1 & -4 & 4 & 8 \\ 0 & 4 & -2 & -1 & -1 & -1 & 5 & -2 \\ 0 & 0 & 1 & 3 & 8 & 4 & 6 & -2 \\ 1 & -2 & 0 & 5 & 1 & 1 & 4 & -6 \\ 3 & -4 & 0 & 6 & -2 & -2 & 2 & 2 \end{matrix}$$

$(i, j) = f(i, j) - \tilde{f}(i, j)$

JPEG compression for a smooth image block

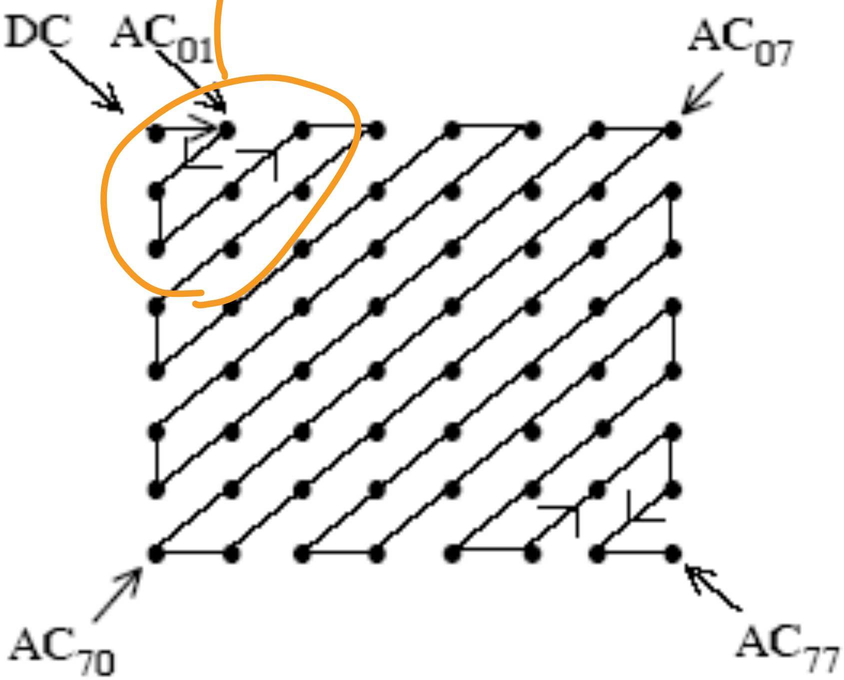
Entropy Coding of DCT Coefficients

- A two-step process:
  - The first step converts the zig-zag sequence of quantized coefficients into an intermediate sequence of symbols, turning the 8×8 matrix into a 64-vector. This increases the likelihood of creating a long run of zeros.
  - The second step converts the intermediate symbols into code stream
- The form and definition of the intermediate symbols is dependent on both the mode of operations and the entropy coding methods
- Entropy coding methods
  - Huffman coding
  - Arithmetic coding
- DC and AC coefficients are coded separately (but in a similar way) 特性不一定一樣，分開處理.

Zig-Zag Sequence

∵有值的多在左上角，剩下几乎都是0

∴用 zig-zag scan



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

Encoding AC Coefficients

- Each nonzero AC coefficient is represented in combination with the "runlength" (consecutive number) of preceding zero-valued AC coefficients

它前面的0有几個

non-zero coefficient

Symbol_1	**Symbol_2**
(RUNLENGTH, SIZE)	**(AMPLITUDE)**

- RUNLENGTH
  - Represents zero-runs of length 0 to 15
  - Zero-runs can be greater than 15. The value (15,0) for symbol-1 is used as an extension symbol with runlength=16.
  - There can be up to 3 consecutive (15,0) extensions before the terminating symbol_1 whose RUNLENGTH value completes the actual runlength.
  - The terminating symbol_1 is always followed by a single symbol_2, except for the case where the last run of zeros includes the last (63rd) AC coefficient.
  - The special symbol_1 value (0,0) means EOB (end of block), which denotes that the remaining coefficients in the block are zero.

後面都不用再encode了

Range of AC Coefficients

- SIZE
  - The number of bits needed to represent the coefficient
- AMPLITUDE
  - Bit pattern of the nonzero AC coefficient amplitude
- For 8-bit integer source samples
  - Range of input pixel value: $[-2^7, 2^7-1]$
  - Then range of the amplitude of quantized AC coefficient is $[-2^{10}, 2^{10}-1]$
  - So the symbol AMPLITUDE has 1 to 10 bits in length Hence, SIZE has values from 1 to 10.
  - RUNLENGTH represents values from 0 to 15, as discussed earlier
- Separate Huffman tables for luminance and chrominance

∵ luminance 和 chrominance
發生的机率不同, 所以理應用不同 table???

Structure of AC Code Table

* 用几個 bits ?

Symbol_1

		SIZE 0	1	2	. . .	9	10
RUN LENGTH	0	EOB					
	.	X			RUN-SIZE values		
	.	X					
	.	X					
	15	ZRL					

Entries marked "X" are undefined

Symbol_2

SIZE	AMPLITUDE
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

- The composite value (RUNLENGTH, SIZE) of symbol_1 is Huffman coded, since smaller sizes occurred more often.
- Each Huffman code is followed by additional bits representing SYMBOL_2 that specify the sign and exact magnitude of the coefficient (denote it by ZZ(K)).
- The AC code table consists of one Huffman code for each possible composite value. The maximum length is 16 bits, excluding the additional bits.
- The value of SIZE gives the number of additional bits required to specify the sign and precise magnitude of the coefficient ZZ(K). The additional bits are either the low-order SIZE bits of the coefficient if it is positive or the low-order SIZE bits of ZZ(K)-1 if it is negative.

Huffman Table for Luminance AC

Table K.5 Table for luminance AC coefficients

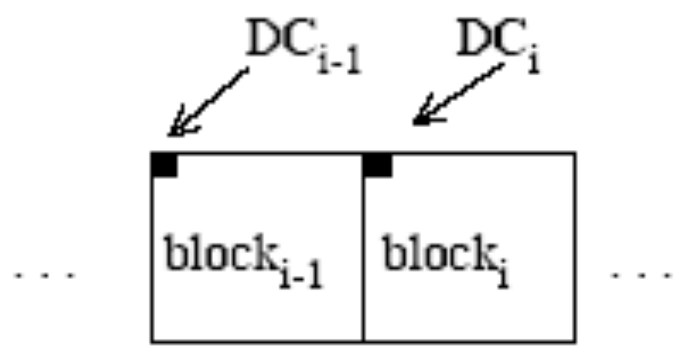
Run/Size	Code length	Code word
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
2/4	12	111111110100
2/5	16	1111111110001001
2/6	16	1111111110001010
2/7	16	1111111110001011
2/8	16	1111111110001100
2/9	16	1111111110001101
2/A	16	1111111110001110
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
3/4	16	1111111110001111
3/5	16	1111111110010000
3/6	16	1111111110010001
3/7	16	1111111110010010
3/8	16	1111111110010011
3/9	16	1111111110010100
3/A	16	1111111110010101
4/1	6	111011

⋮

Encoding DC Coefficients (1/4)

- Each DC coefficient is coded differentially, using the quantized DC value of the most recently coded 8x8 block.
- The coefficient DIFF is obtained as follows:

$$DIFF = DC_i - DC_{i-1}$$



Encoding DC Coefficients (2/4)

- Each DPCM coded DC coefficient is represented by two symbols

SIZE → Symbol_1　　AMPLITUDE → Symbol_2

  - SIZE: Number of bits for representing the coefficient
  - AMPLITUDE: Two's complement difference magnitude
  - Example (using the code from the table on the next page)

sign

150 → (8, 010010110),

−6 → (3, 1001)

第一個bit是sign
∴ symbol 1 的值是不含sign的

- SIZE is Huffman coded since smaller sizes occurred more often
- AMPLITUDE is not Huffman coded. Its value can change widely so Huffman coding has no appreciable benefit

Encoding DC Coefficients (3/4)

- The two's complement difference magnitudes are grouped into 12 categories indicated by SIZE
- A Huffman code is created for each of the 12 categories

16 8 4 2 1
1 1 0 ~~1~~ 0   0111
0 0 1 0 1   1000

SIZE	DIFF values
0	0
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
11	-2047..-1024,1024..2047

Table K.3 Table for luminance DC difference

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table K.4 Table for chrominance DC difference

Category	Code length	Code word
0	2	00
1	2	01
2	2	10
3	3	110
4	4	1110
5	5	11110
6	6	111110
7	7	1111110
8	8	11111110
9	9	111111110
10	10	1111111110
11	11	11111111110

Encoding DC Coefficients (4/4)

- For each category, except SIZE=0, an additional bit field is appended to the code word to uniquely specify which difference in that category actually occurred.
- The number of extra bits is given by SIZE
- When DIFF is positive, the SIZE low-order bits of DIFF are transmitted.
- When DIFF is negative, the SIZE low-order bits of (DIFF-1) are transmitted.
- The most significant bit of the appended bit sequence is 0 for negative differences and 1 for positive differences.

※若不是用2補數？那規則是啥？

1's complement？

Baseline Encoding Example (1/3)

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

(a) source image samples

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

(b) forward DCT coefficients

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

(c) quantization table

$\frac{235.6}{16}$

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(d) normalized quantized coefficients

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(e) denormalized quantized coefficients

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

(f) reconstructed image samples

Baseline Encoding Example (2/3)

- DC coefficient
  - Assuming previous DC coefficient = 12
  - 15 –12 = 3
  - So the intermediate representation of the differential DC is (2,3). That is, SIZE=2 and AMPLITUDE=3.
- AC coefficients
  - First nonzero coefficient = – 2; zero-run = 1
  - 2nd nonzero coefficient = – 1; zero-run = 0
  - 3rd nonzero coefficient = – 1; zero-run = 0
  - 4th nonzero coefficient = – 1; zero-run = 0
  - 5th nonzero coefficient = – 1; zero-run = 2

-2出現前有1個0

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(d) normalized quantized coefficients

- The intermediate sequence of symbols is

  (2,3), (1,2) (–2), (0,1) (–1), (0,1)(–1), (0,1) (–1), (2,1) (–1), (0,0)

  EOB

(2,3), (1,2) (−2), (0,1) (−1), (0,1)(−1), (0,1) (−1), (2,1) (−1), (0,0) → 這項要把它們變0和1

EOB

Baseline Encoding Example (3/3)

- Codes of the sequence of symbols
  - VLC for the differential DC 查表得011
    - $\boxed{(2) \rightarrow 011}$
  - VLC for luminance AC
    - (0,0) → 1010
    - (0,1) → 00
    - (1,2) → 11011 查表
    - (2,1) → 11100
  - Amplitude
    - 3 → 11
    - -2 → 01
    - -1 → 0

    by p21的那個像

    1's complement 的轉換
  - Bit-stream for the 8x8 block (31 bits)
    - 01111 1101101 000 000 000 111000 1010

    (2,3) (1,2) 2 (0,1)-1 (0,1)-1 (0,1)-1 (2,1)-1 (0,0)

p22的表

Table K.3 Table for luminance DC difference

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Table K.5 Table for luminance AC coefficients

Run/Size	Code length	Code word
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000

Two's complement examples

- 3 → 011
- 2 → 010
- 1 → 001
- 0 → 000
- -1 → 111
- - 2 → 110
- - 3 → 101

One's complement examples

- 3 → 011
- 2 → 010
- 1 → 001
- 0 → 000
- - 1 → 110
- - 2 → 101
- - 3 → 100

Four JPEG Modes

- Sequential Mode
  - The default JPEG mode, implicitly assumed in the discussions so far.
  - Each graylevel image or color image component is encoded in a single left-to-right, top-to-bottom scan.
- Progressive Mode
- Hierarchical Mode
- Lossless Mode

Progressive Mode

Progressive JPEG delivers low quality versions of the image quickly, followed by higher quality passes. Two ways:

1.**Spectral selection**: Takes advantage of the "spectral" (spatial frequency spectrum) characteristics of the DCT coefficients: higher AC components provide detail information.

先低頻再高頻

- Scan 1: Encode DC and first few AC components, e.g., AC1, AC2.
- Scan 2: Encode a few more AC components, e.g., AC3, AC4, AC5.
- ...
- Scan k: Encode the last few ACs, e.g., AC61, AC62, AC63.

Progressive Mode (Cont’d)

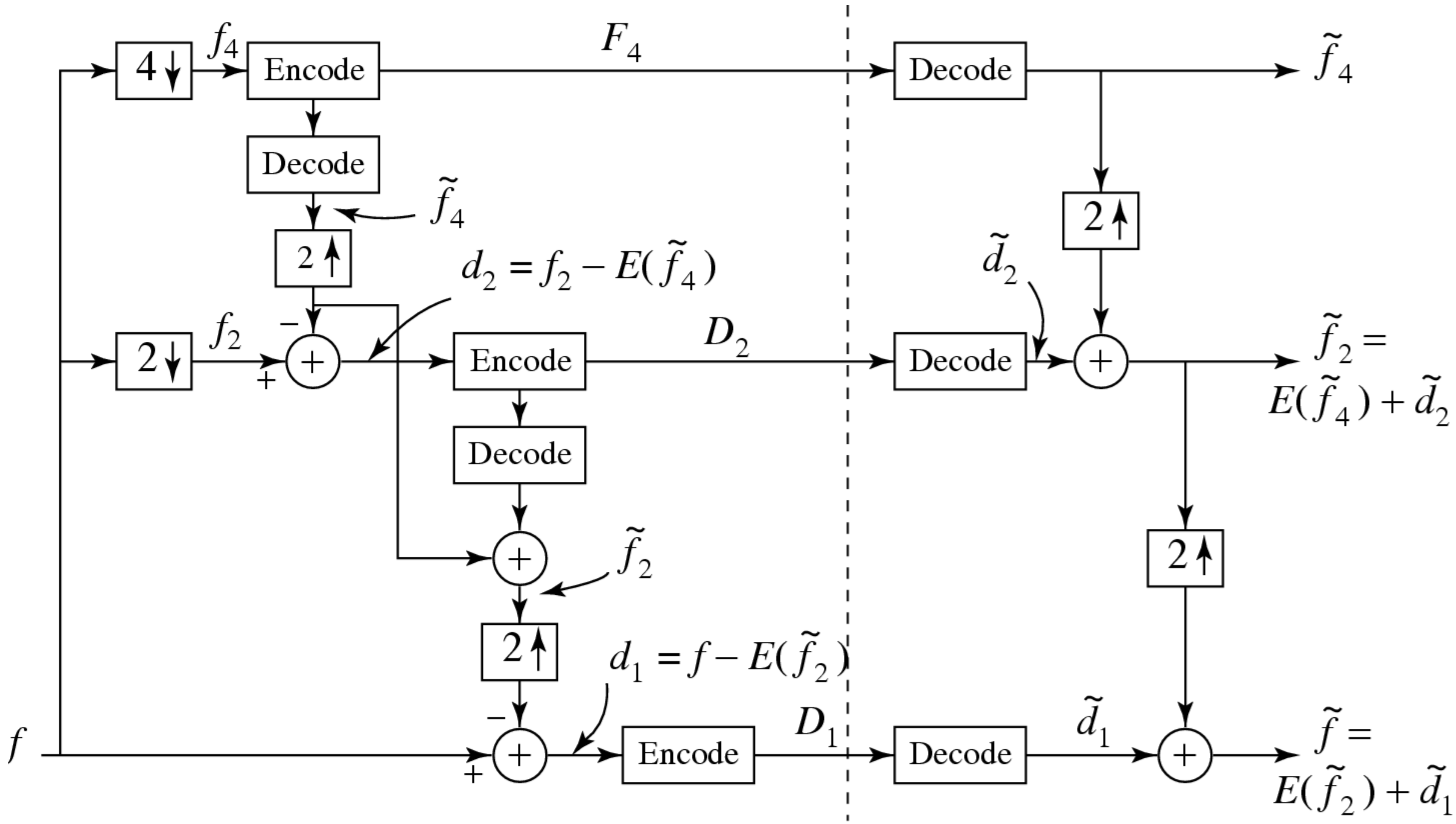
2. **Successive approximation**: Instead of gradually encoding spectral bands, all DCT coefficients are encoded simultaneously but with their most significant bits (MSBs) first. 從MSB開始

- Scan 1: Encode the first few MSBs, e.g., Bits 7, 6, 5, 4.
- Scan 2: Encode a few more less significant bits, e.g., Bit 3.
- ...
- Scan m: Encode the least significant bit (LSB), Bit 0.

Hierarchical Mode

- The encoded image at the lowest resolution is basically a compressed low-pass filtered image, whereas the images at successively higher resolutions provide additional details (differences from the lower resolution images).

- Similar to Progressive JPEG, the Hierarchical JPEG images can be transmitted in multiple passes to progressively improve image quality.



Block diagram of three-level hierarchical JPEG

Encoder for a Three-Level Hierarchical JPEG

1. Reduction of image resolution:
   - Reduce resolution of the input image f (e.g., 512×512) by a factor of 2 in each dimension to obtain f_2 (e.g., 256 × 256). Repeat this to obtain f_4 (e.g., 128 × 128).
2. Compress low-resolution image f_4:
   - Encode f_4 using any other JPEG method (e.g., Sequential, Progressive) to obtain F_4.
3. Compress difference image d_2:
   - Decode F_4 to obtain $\tilde{f}_4$. Use any interpolation method to expand $\tilde{f}_4$ to be of the same resolution as f_2 and call it $E(\tilde{f}_4)$.
   - Encode difference $d_2 = f_2 - E(\tilde{f}_4)$ using any other JPEG method (e.g., Sequential, Progressive) to generate D_2.
4. Compress difference image d_1:
   - Decode D_2 to obtain $\tilde{d}_2$; add it to $E(\tilde{f}_4)$ to get $\tilde{f}_2 = E(\tilde{f}_4) + \tilde{d}_2$ which is a version of f_2 after compression and decompression.
   - Encode difference $d_1 = f - E(\tilde{f}_2)$ using any other JPEG method (e.g., Sequential, Progressive) to generate D_1.

Three-Level Hierarchical JPEG Decoder

- Decompress the encoded low-resolution image F_4

  Decode F_4 using the same JPEG method as in the encoder to obtain . $\tilde{f}_4$

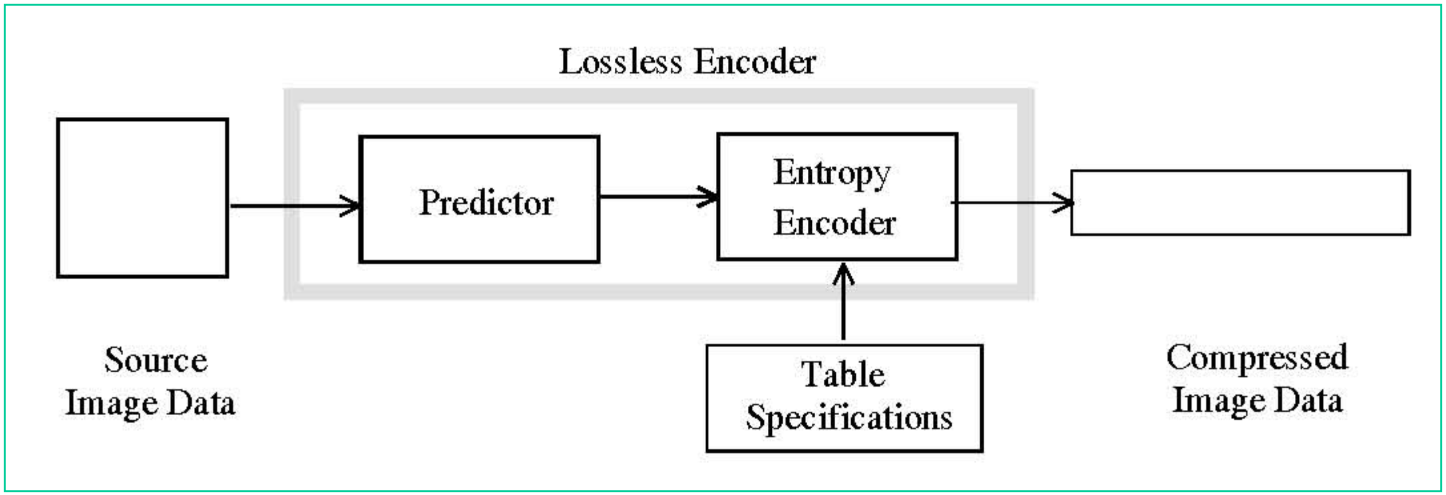
- Restore image $\tilde{f}_2$ at the intermediate resolution

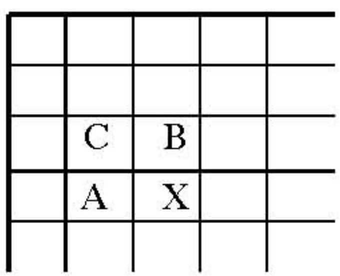
  Use $E(\tilde{f}_4) + \tilde{d}_2$ to obtain $\tilde{f}_2$.

- 3. Restore image $\tilde{f}$ at the original resolution

  Use $E(\tilde{f}_2) + \tilde{d}_1$ to obtain $\tilde{f}$.

Predictive Lossless Coding Mode





selection-value	prediction
0	no 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

9.1.3 JPEG Bitstream Structure

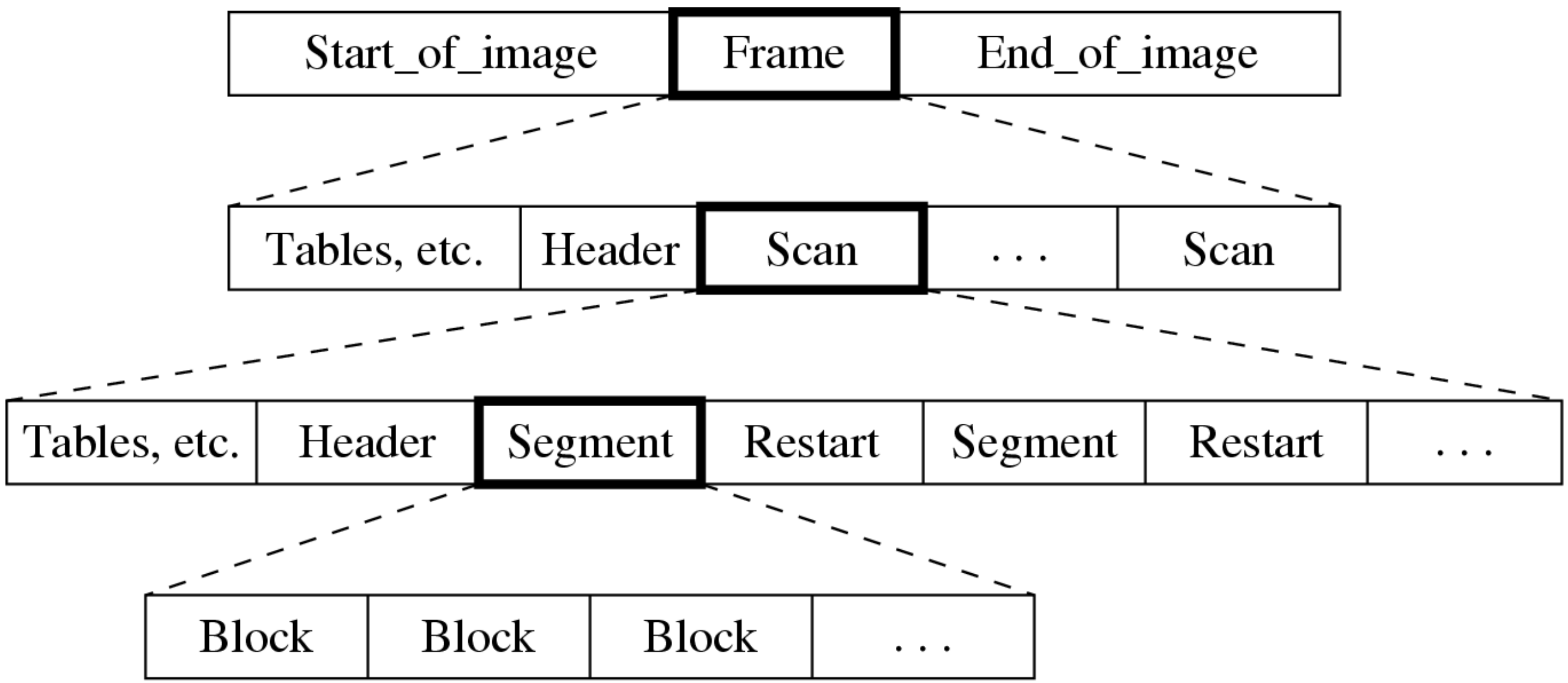


Fig. 9.6: JPEG bitstream.

Bitstream Syntax

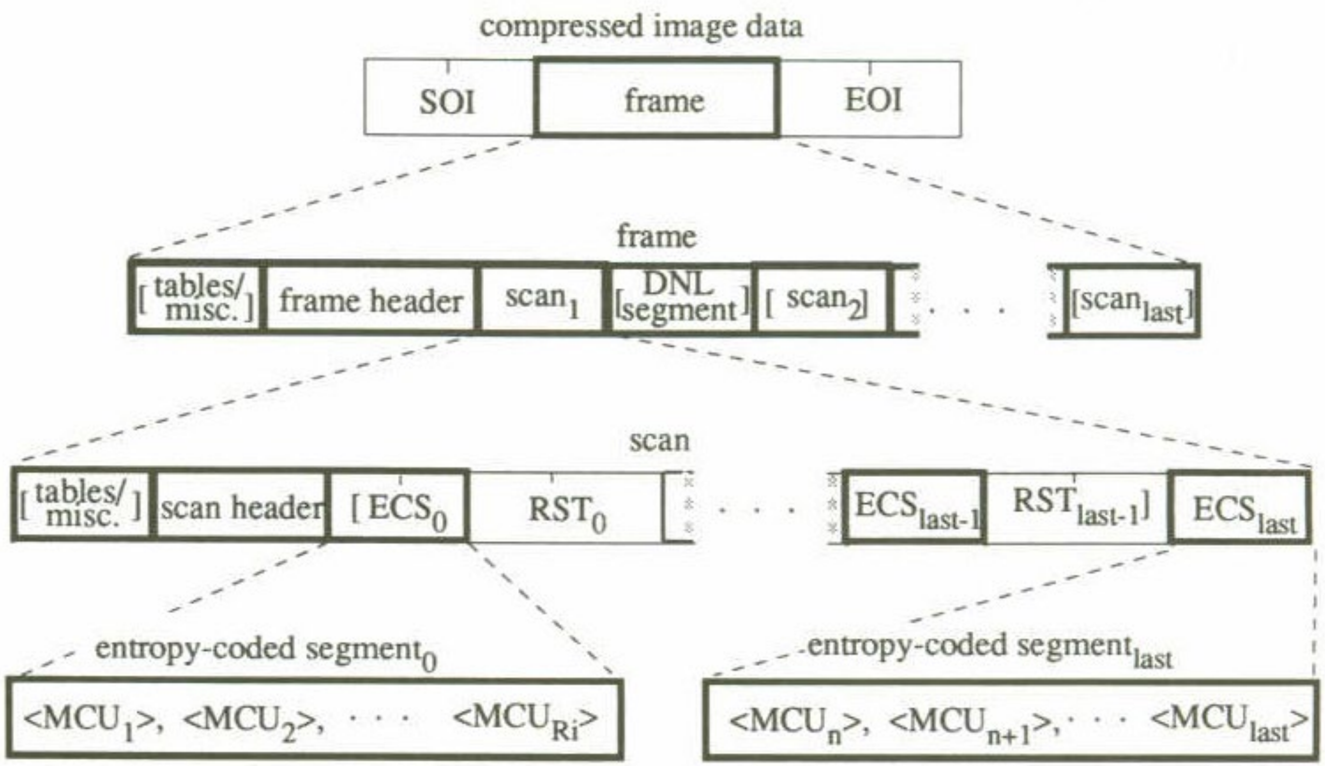


Figure B.2 - Syntax for sequential DCT-based, progressive DCT-based, and lossless modes of operation

SOI: Start of image
EOI: End of image
RST: Restart marker
ECS: Entropy-coded segment
DNL: Define number of lines
MCU: minimum coded unit

Syntax

The three markers shown in Figure B.2 are defined as follows:

SOI: start of image marker: marks the start of a compressed image represented in the interchange format.

EOI: end of image marker: marks the end of a compressed image represented in the interchange format.

RST$_m$: restart marker: an optional marker which is placed between entropy-coded segment only if restart is enabled. There are 8 unique restart markers (m=0-7) which repeat in sequence from 0 to 7 to provide a modulo 8 restart interval count.

A *frame* is a picture.

A *scan* is a pass through the pixels (eg. the red component).

A *segment* is a group of blocks.

A *block* consists of 8x8 pixels.

Picture Quality

Bit Rate (bits/pixel)	Quality
0.25-0.5	Moderate to good
0.5-0.75	Good to very good
0.75-1.5	Excellent
1.5-2.0	Visually indistinguishable from the original

JPEG Changed Our World

-
- *12.12.14* - **JPEG is the image format we use the most in the world, be it in our computers, smartphones or digital cameras. But it is actually more than that. JPEG is also an international group of experts, which recently elected as its head an EPFL Professor: Touradj Ebrahimi. The new President sheds light on the JPEG adventure and gives us an overview of new features in development.**

Four letters, over twenty years in existence and a virtual world monopoly in image compression. The JPEG standard, which is used more than one billion times each day on social networks alone, recently elected as its Head the EPFL professor Touradj Ebrahimi. At its core, what is the significance of JPEG? Touradj Ebrahimi gives us an overview of this popular standard that has revolutionized our lives.

JPEG Changed Our World

- **JPEG is everywhere. The basis of this project, however, was laid out over 30 years ago...**
  This is true. JPEG is an image format created to address a major problem in the digital age. In the early 80s, no technology existed to copy or transmit electronic images. Minitel, invented by the French, only allows you to send text and simple graphics. We had to find a way to reduce the size of image files. It was the international standardization groups and telecommunications industry that provided the impetus for the creation of JPEG. In 1982, they brought world experts in image compression to the table to form the "Joint Photographic Experts Group (JPEG)." The JPEG format was created in 1992.
- **How does JPEG reduce image file sizes more that other formats?**
  Basically, JPEG relies on a lossy compression algorithm. When an image is compressed in JPEG, a portion of its contents is destroyed, but practically without being noticeable to the human eye. This technique makes it possible to reduce the image file size, so that it can be easily stored, copied and transmitted. Over the years, other formats with various technologies have been developed, such as JPEG2000, which can operate with or without loss. Lossless compression is used mainly in the medical settings, where it is crucial to preserve image details. The JPEG group is also working on a new standard called JPEG XT, which will allow efficient compression of High-Dynamic-Range images with a wide color gamut.

JPEG Changed Our World

- **Why has JPEG become a global standard?**
  At the time of the emergence of Internet and digital devices, JPEG was the only international standard that was free and accessible to all. While other commercial compression formats were offered by private companies (e.g.: Kodak), JPEG was the only partially open format, suitable for all devices and software, requiring no royalties. Another reason for this success is that new JPEG formats, such as JPEG XT, are developed so as to be always readable with an old version of JPEG. It's important not to rush the consumers, because they do not like abrupt changes. The proof is that after 22 years of existence, JPEG has strengthened and is increasingly popular.
- **If everything is open and free, what's the point of working on the development of standards such as JPEG?**
  There are several. When a format is free, developers and consumers are more inclined to use it. If it is also open source, it means that everybody can access the details of the algorithm, and try to improve it. For scientists, it is therefore easier to analyse the strengths and weaknesses of the approach, and to propose new ideas to improve it.

JPEG Changed Our World

- **What are your next challenges as President of JPEG?**
  Before the end of 2015, we plan to finalize the new JPEG XT format, for an efficient compression of High-Dynamic-Range (HDR) images, which are increasingly widespread. JPEG XT contains enough information to correct the shot when a picture taken by a digital camera is under-exposed or over-exposed resulting in a too dark or too bright image. Then, for the years 2015-2020, we will launch the basis of a new version of JPEG called JPEG Pleno, in reference to the term "plenoptic." The idea is to correct the focus of a picture after it has been taken, and to make any object in the image sharper even if it appears blurred in the initial picture. It will also be possible to look at any object in the picture, even from a different perspective than what the camera that captured it..
- **Why is image transmission so important to human beings?**
  To have value and meaning, an image must be shared and copied. Since the dawn of time, humans have sought to transmit images, whether by painting on cave walls, by copying illustrations manually or by mechanically reproducing them with the invention of the printing press. The instrument of JPEG is part of this mission of transmission, but this time in digital form. It has changed our world. Today, almost everyone has used JPEG, whether via computer, smartphone, tablet or camera. And yet, most of the time, people don't really know what it is.

JPEG Changed Our World

- **How do you predict our relationship with images will evolve in the future?**
  I hope JPEG Pleno can last for 20 to 30 years. Then, in my opinion, we will progress towards a point at which it's no longer necessary to go through the eye to access images. Images could for example be injected directly into the brain!