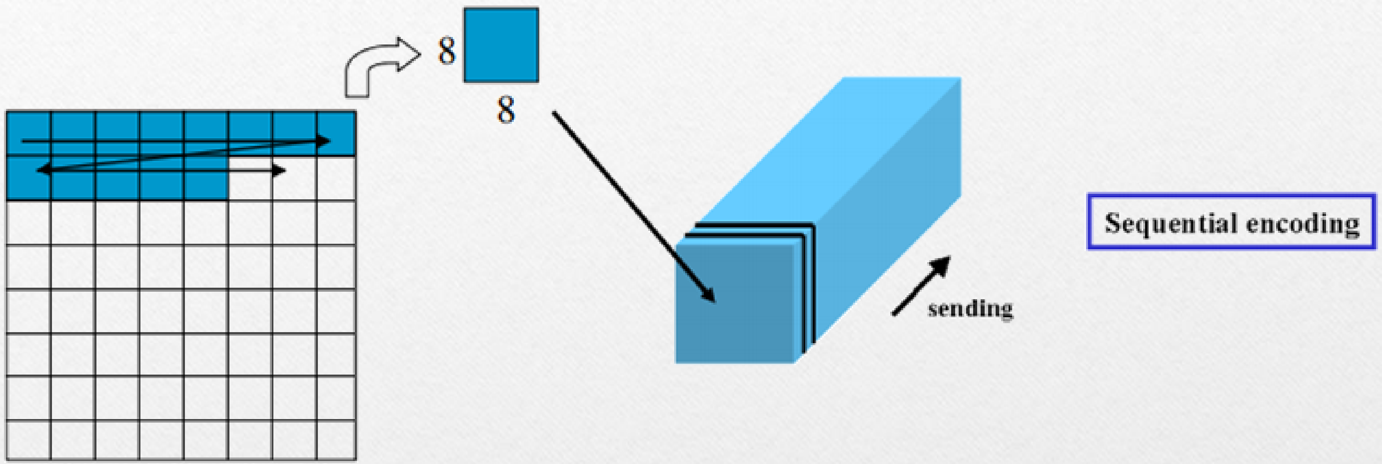
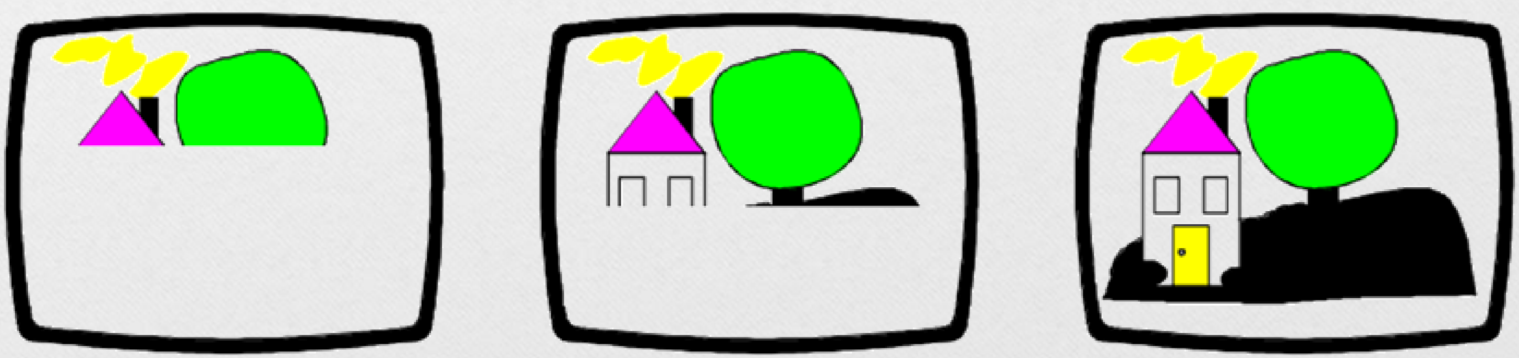
Jpeg Decoder

Baseline Sequential DCT-based

Baseline Sequential DCT-based





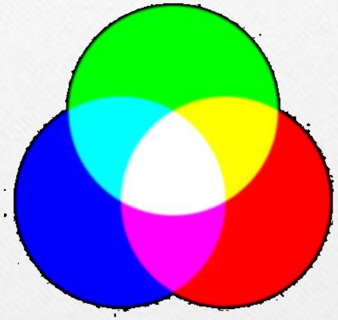
Baseline Sequential DCT-based Encoding Process

- Color Space Conversion
- Subsampling
- Partition
- Encoding Flow Control
  - Discrete Cosine Transform (DCT)
  - Quantization
  - Entropy Encoding (Huffman)

Decoding

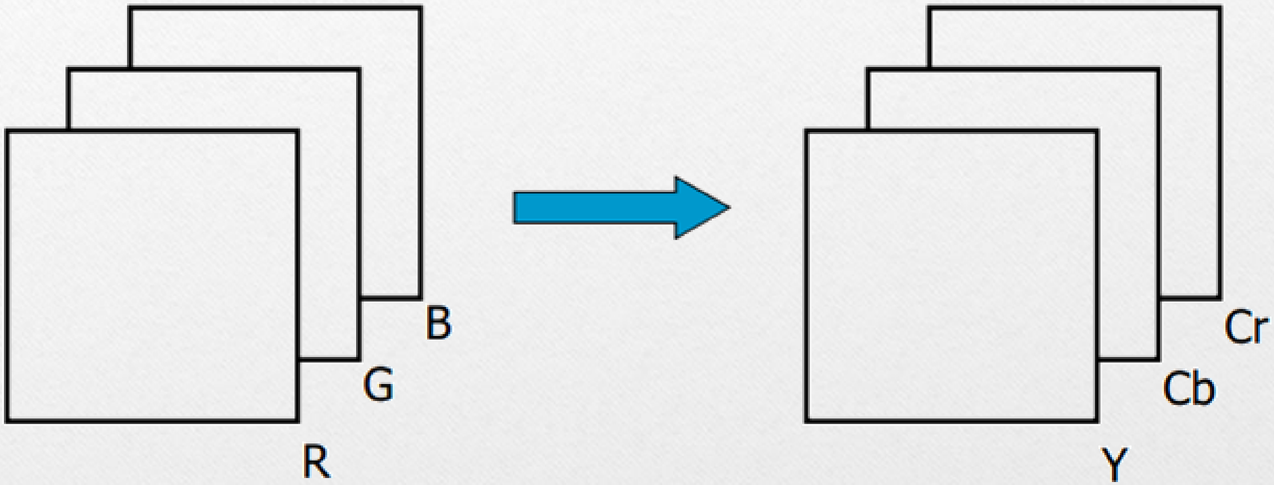
Color Space Conversion

RGB



YCbCr

Color Space Conversion



Y – Luminance(亮度) Domain Cb,Cr – Chrominance(色差) Domain

$$Y = 0.299R + 0.587G + 0.144B \qquad R = Y + 1.402 * (Cr - 128)$$

$$Cb = -0.168R - 0.331G - 0.449B \qquad G = Y - 0.34414 * (Cb - 128) - 0.71414 * (Cr - 128)$$

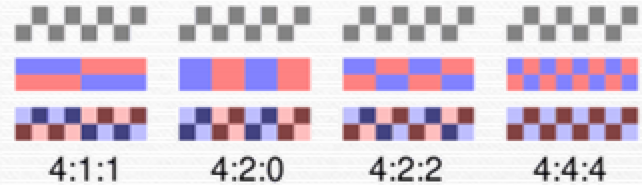
$$Cr = 0.500R - 0.419G - 0.081B \qquad B = Y + 1.772 * (Cb - 128)$$

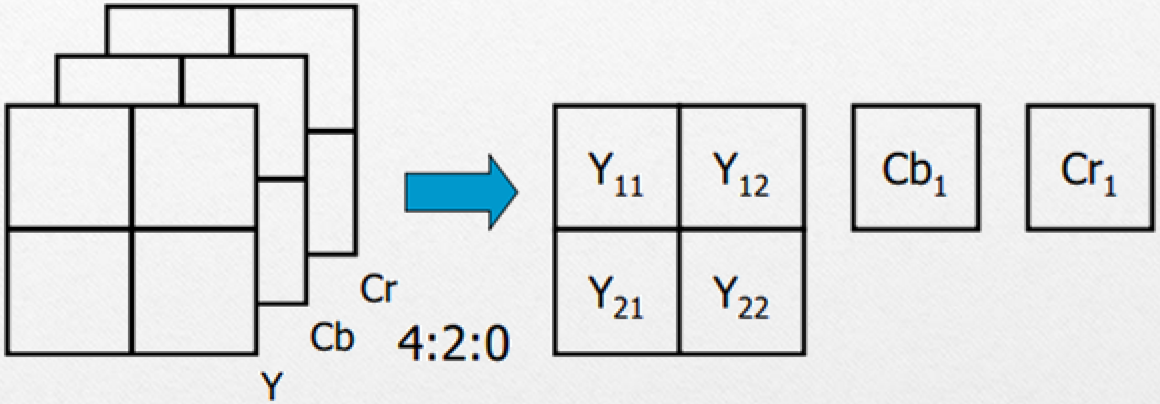
Baseline Sequential DCT-based Encoding Process

- Color Space Conversion
- Subsampling
- Partition
- Encoding Flow Control
  - Discrete Cosine Transform (DCT)
  - Quantization
  - Entropy Encoding (Huffman)

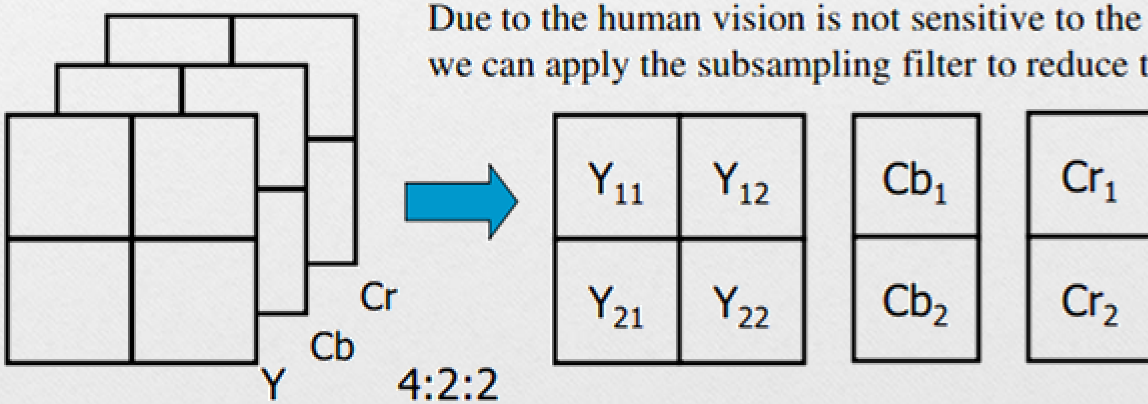
Decoding

Subsampling





Due to the human vision is not sensitive to the chrominance domain, we can apply the subsampling filter to reduce the data size.



Note: If there is no subsampling, we called this 4:4:4 mode

Baseline Sequential DCT-based Encoding Process

- Color Space Conversion
- Subsampling
- Partition
- Encoding Flow Control
  - Discrete Cosine Transform (DCT)
  - Quantization
  - Entropy Encoding (Huffman)

Decoding

Partition

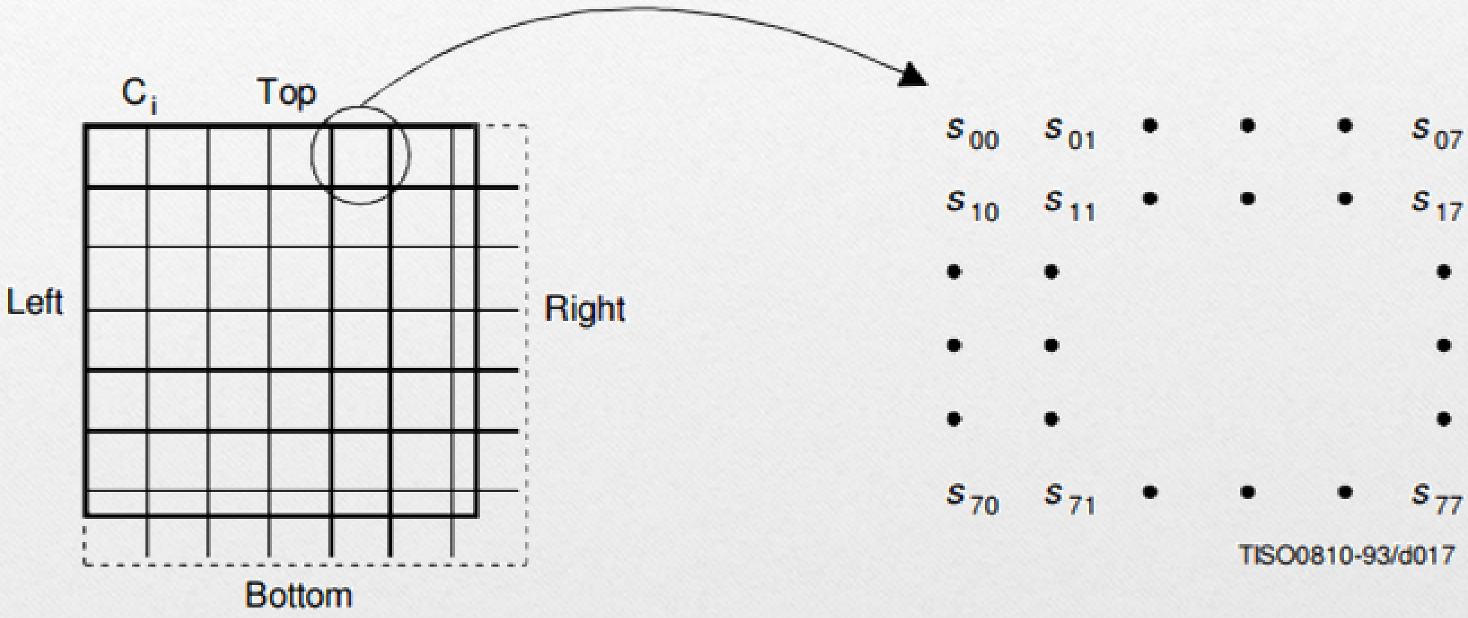


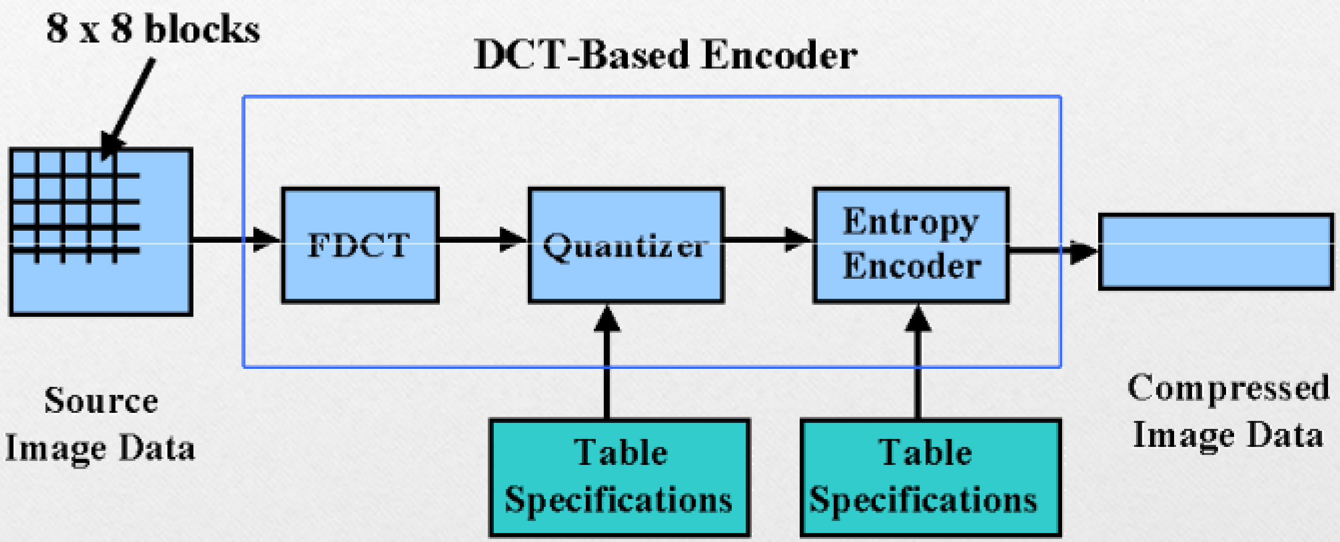
Figure A.4 – Partition and orientation of 8 x 8 sample blocks

Baseline Sequential DCT-based Encoding Process

- Color Space Conversion
- Subsampling
- Partition
- Encoding Flow Control
  - Discrete Cosine Transform (DCT)
  - Quantization
  - Entropy Encoding (Huffman)

Decoding

Encoding Flow control



DCT-Based Encoder Processing Steps

Quantization

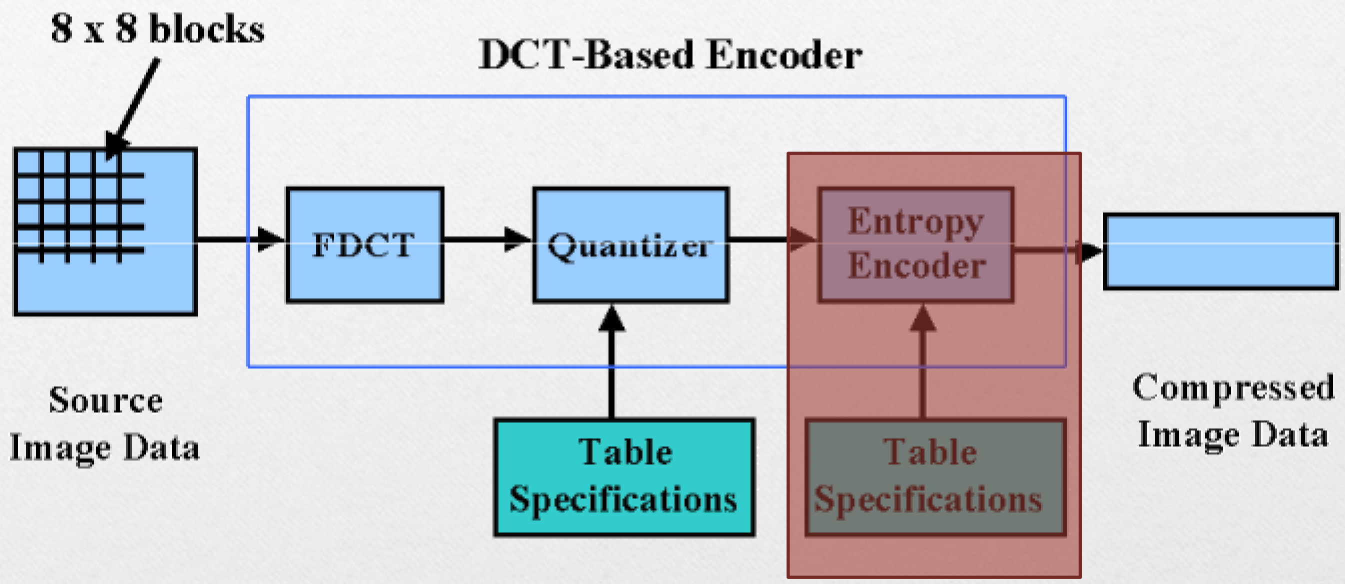
Table K.1 – Luminance quantization table

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

Encoding Flow control



DCT-Based Encoder Processing Steps

Huffman Encoding

- DC Diff
- AC Run-Length pair

DC Encoding, AC Scanning

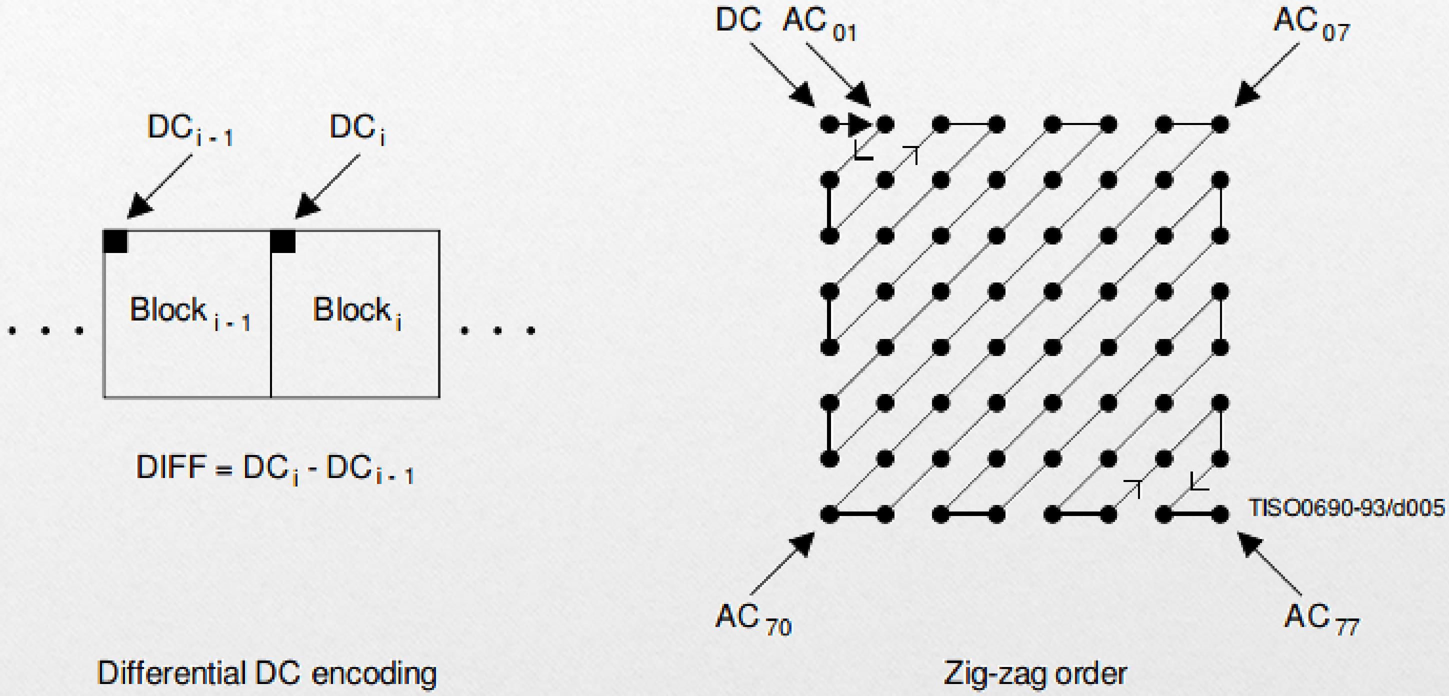


Figure 5 – Preparation of quantized coefficients for entropy encoding

DC Diff

(ITU-T81 Annex F)

SSSS	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	–1 023..–512,512..1 023
11	–2 047..–1 024,1 024..2 047

1. Decoding codeword with *Huffman decoding* for getting SSSS
2. Using following SSSS bits as index for looking up DIFF value
3. Plus the DC value of last block

DC Huffman Decoding

$T = \mathrm{DECODE}$

$\mathrm{DIFF} = \mathrm{RECEIVE}(T)$

$\mathrm{DIFF} = \mathrm{EXTEND}(\mathrm{DIFF}, T)$

Example: 1011111…………

1. Decode and get $101_2 \rightarrow T = 4$

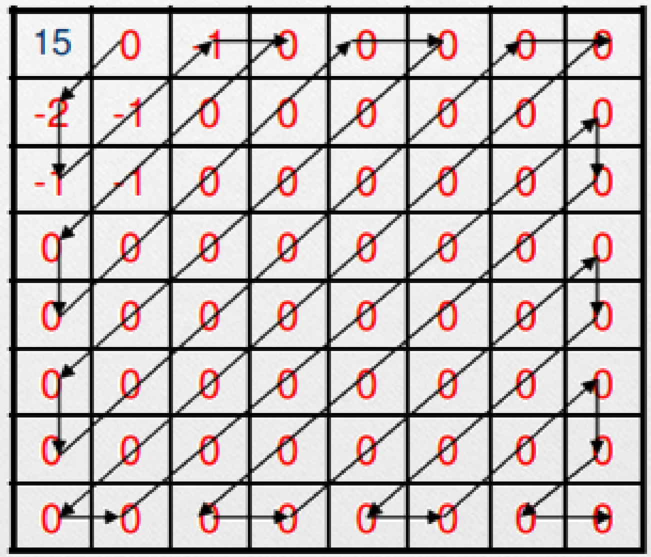
Huffman decoding

SSSS	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
….	….

2. Get T bits $\rightarrow$ Get $1111_2 \rightarrow$ DIFF = 1111_2
3. Extent DIFF $\rightarrow$ Get 15

AC Run-length encoding

- Run-length encoding



15 0 –2 –1 –1 –1 0 0 –1 0 0 …. 0

<1,-2> <0,-1> <0,-1> <0,-1> <2,-1> <EOB>

AC Huffman Encoding

SSSS	AC coefficients
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8,15
5	-31..-16,16..31
6	-127..-64,64..127
….	….

AC coefficient magnitude category

RRRRSSSS

Run/Size	Code length	Code word
0 / 0 (EOB)	4	1010
0 / 1	2	00
..	..	..
1 / 2	5	11011
..	..	..
2 / 1	5	11100
….	….	….

AC Huffman Table

AC code word = Run/Size code word + AC coefficient code word

<1,-2>	<0,-1>	<0,-1>	<0,-1>	<2,-1>	<EOB>
11011 01	00 0	00 0	00 0	11100 0	1010

AC Huffman Decoding

Example: 1101101......

		SSSS						
		0	1	2	.	.	13	14
RRRR	0	EOB	COMPOSITE VALUES					
	.	N/A						
	.	N/A						
	.	N/A						
	15	ZRL						

1. Decode and get $11011_2 \rightarrow$ RS = 1/2 $\rightarrow$ S=2

Huffman decoding

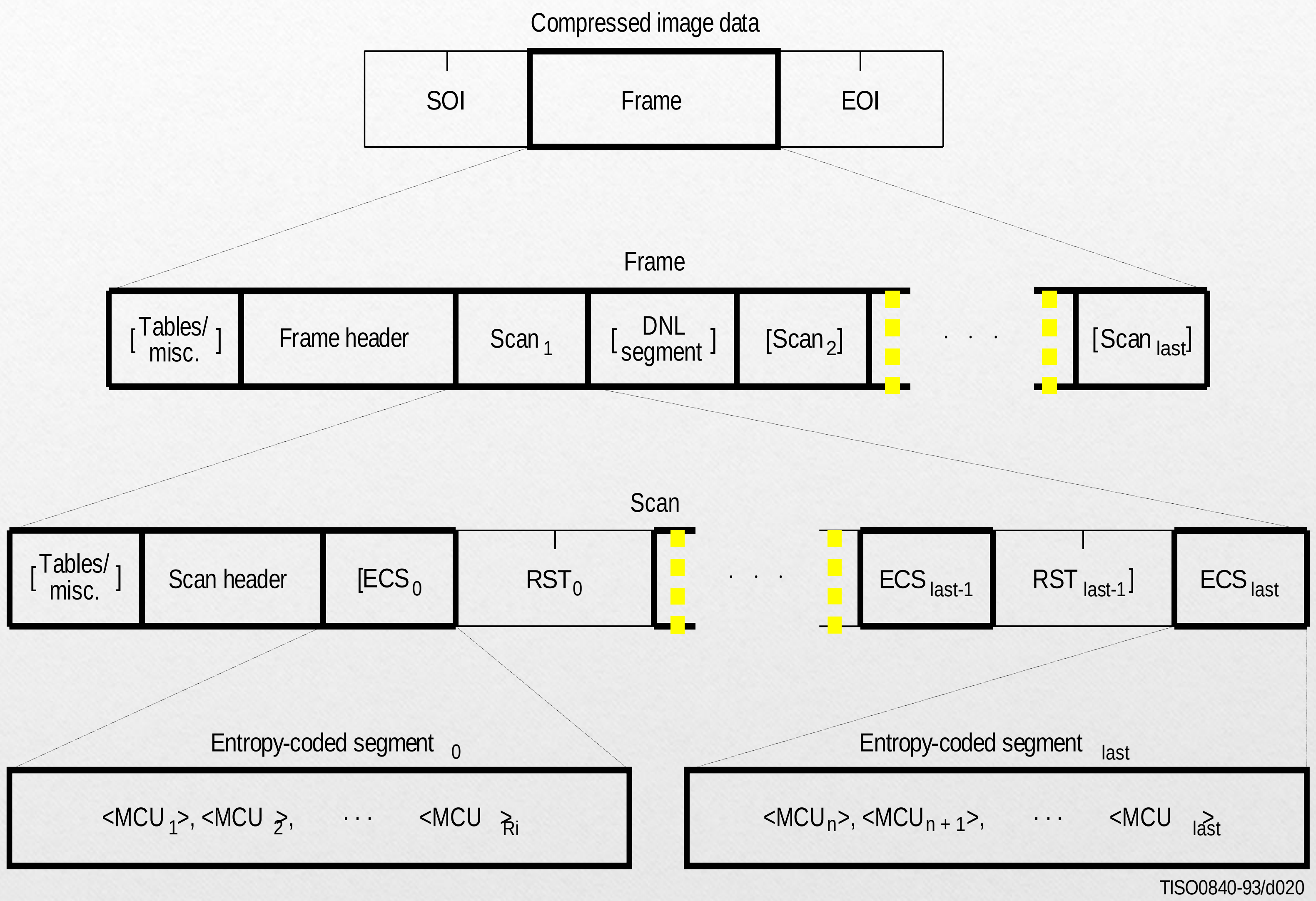
2. Get S bits $\rightarrow$ SSSS = 01_2
3. Extent SSSS $\rightarrow$ SSSS = -2_{10}

SSSS	AC coefficients
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8,15
5	-31..-16,16..31
6	-127..-64,64..127
....	....

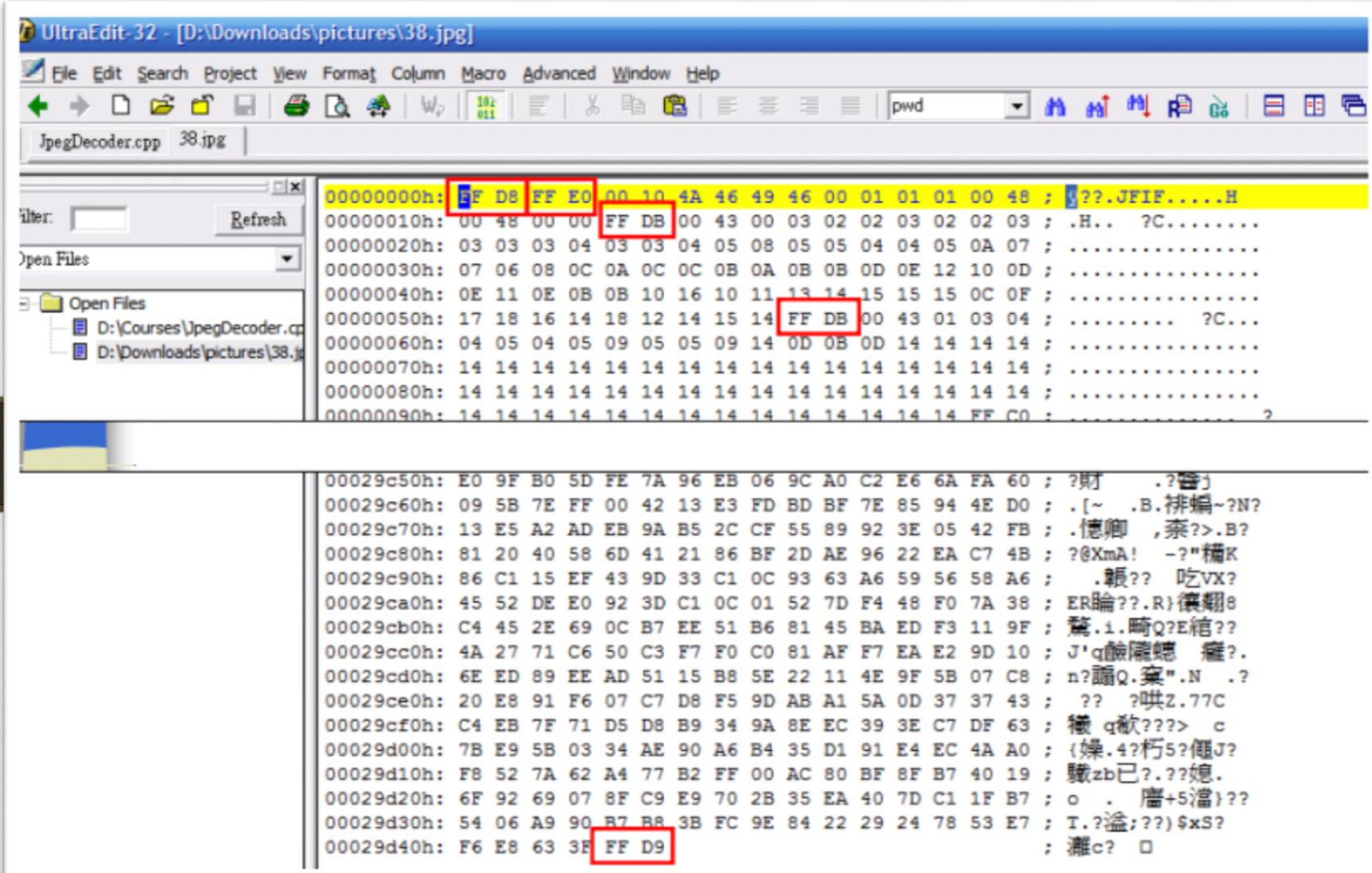
4. Finally, (Run, Length) = (R,SSSS) = (1,-2)

Header Processing

Important header


Compressed image data
SOI
Frame
EOI
Frame
[Tables/ misc.]
Frame header
Scan 1
[DNL segment]
[Scan 2]
· · ·
[Scan last]
Scan
[Tables/ misc.]
Scan header
[ECS 0
RST 0
· · ·
ECS last-1
RST last-1]
ECS last
Entropy-coded segment 0
<MCU 1>, <MCU 2>, · · · <MCU Ri>
Entropy-coded segment last
<MCU n>, <MCU n + 1>, · · · <MCU last>
TISO0840-93/d020

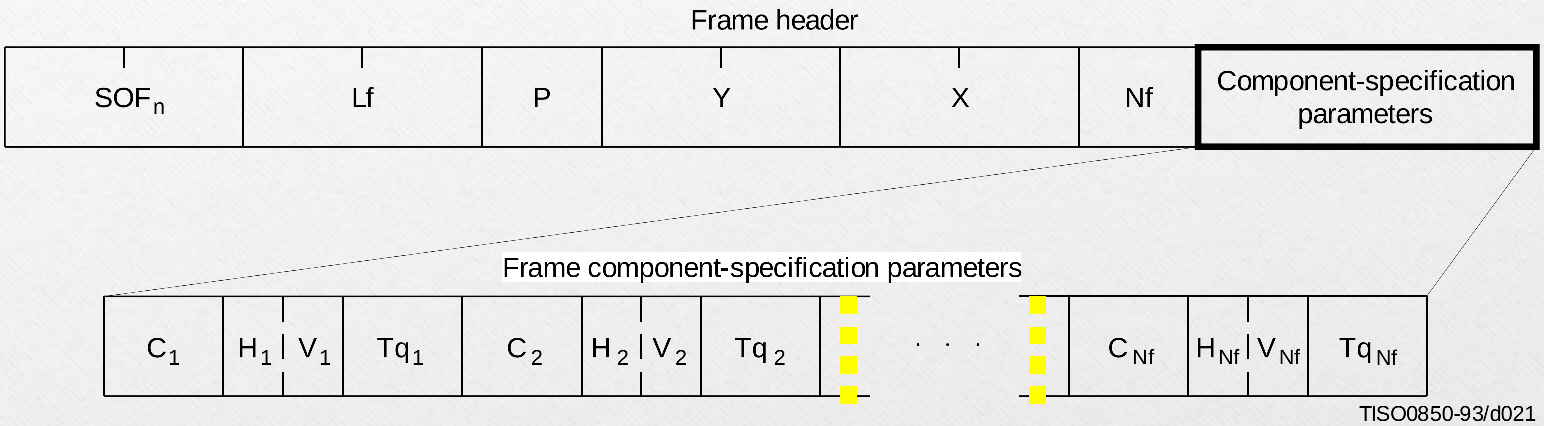
Code Assignment	Symbol	Description
Start Of Frame markers, non-differential, Huffman coding		
X'FFC0'	SOF0	Baseline DCT
Huffman table specification		
X'FFC4'	DHT	Define Huffman table(s)
Restart interval termination		
X'FFD0' through X'FFD7'	RSTm*	Restart with modulo 8 count "m"
Other markers		
X'FFD8'	SOI*	Start of image
X'FFD9'	EOI*	End of image
X'FFDA'	SOS	Start of scan
X'FFDB'	DQT	Define quantization table(s)
X'FFDD'	DRI	Define restart interval
X'FFE0' through X'FFEF'	APPn	Reserved for application segments


UltraEdit-32 - [D:\Downloads\pictures\38.jpg]
File Edit Search Project View Format Column Macro Advanced Window Help
pwd
JpegDecoder.cpp
38.jpg
Filter:
Refresh
Open Files
Open Files
D:\Courses\JpegDecoder.cp
D:\Downloads\pictures\38.j
00000000h: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 48 ; ??.JFIF.....H
00000010h: 00 48 00 00 FF DB 00 43 00 03 02 02 03 02 02 03 ; .H.. ?C........
00000020h: 03 03 03 04 03 03 04 05 08 05 05 04 04 05 0A 07 ; ....
00000030h: 07 06 08 0C 0A 0C 0C 0B 0A 0B 0B 0D 0E 12 10 0D ; ....
00000040h: 0E 11 0E 0B 0B 10 16 10 11 13 14 15 15 15 0C 0F ; ....
00000050h: 17 18 16 14 18 12 14 15 14 FF DB 00 43 01 03 04 ; .... ?C...
00000060h: 04 05 04 05 09 05 05 09 14 0D 0B 0D 14 14 14 14 ; ....
00000070h: 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 ; ....
00000080h: 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 ; ....
00000090h: 14 14 14 14 14 14 14 14 14 14 14 14 14 14 FF C0
00029c50h: E0 9F B0 5D FE 7A 96 EB 06 9C A0 C2 E6 6A FA 60
00029c60h: 09 5B 7E FF 00 42 13 E3 FD BD BF 7E 85 94 4E D0
00029c70h: 13 E5 A2 AD EB 9A B5 2C CF 55 89 92 3E 05 42 FB
00029c80h: 81 20 40 58 6D 41 21 86 BF 2D AE 96 22 EA C7 4B
00029c90h: 86 C1 15 EF 43 9D 33 C1 0C 93 63 A6 59 56 58 A6
00029ca0h: 45 52 DE E0 92 3D C1 0C 01 52 7D F4 48 F0 7A 38
00029cb0h: C4 45 2E 69 0C B7 EE 51 B6 81 45 BA ED F3 11 9F
00029cc0h: 4A 27 71 C6 50 C3 F7 F0 C0 81 AF F7 EA E2 9D 10
00029cd0h: 6E ED 89 EE AD 51 15 B8 5E 22 11 4E 9F 5B 07 C8
00029ce0h: 20 E8 91 F6 07 C7 D8 F5 9D AB A1 5A 0D 37 37 43
00029cf0h: C4 EB 7F 71 D5 D8 B9 34 9A 8E EC 39 3E C7 DF 63
00029d00h: 7B E9 5B 03 34 AE 90 A6 B4 35 D1 91 E4 EC 4A A0
00029d10h: F8 52 7A 62 A4 77 B2 FF 00 AC 80 BF 8F B7 40 19
00029d20h: 6F 92 69 07 8F C9 E9 70 2B 35 EA 40 7D C1 1F B7
00029d30h: 54 06 A9 90 B7 B8 3B FC 9E 84 22 29 24 78 53 E7
00029d40h: F6 E8 63 3F FF D9

Header Structure (ITU-T81 Annex B)

Marker	Number of components	Information

SOF

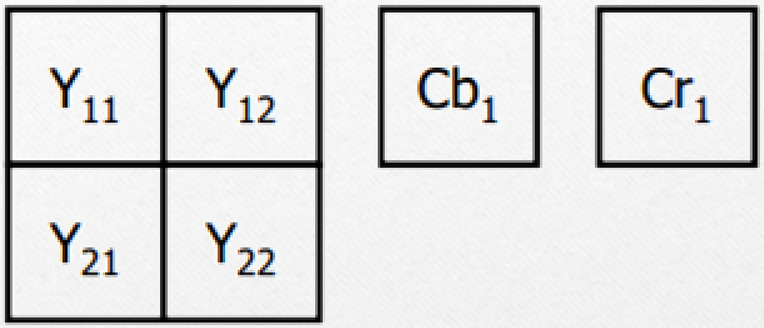


Y: Number of lines
X: Number of samples per line

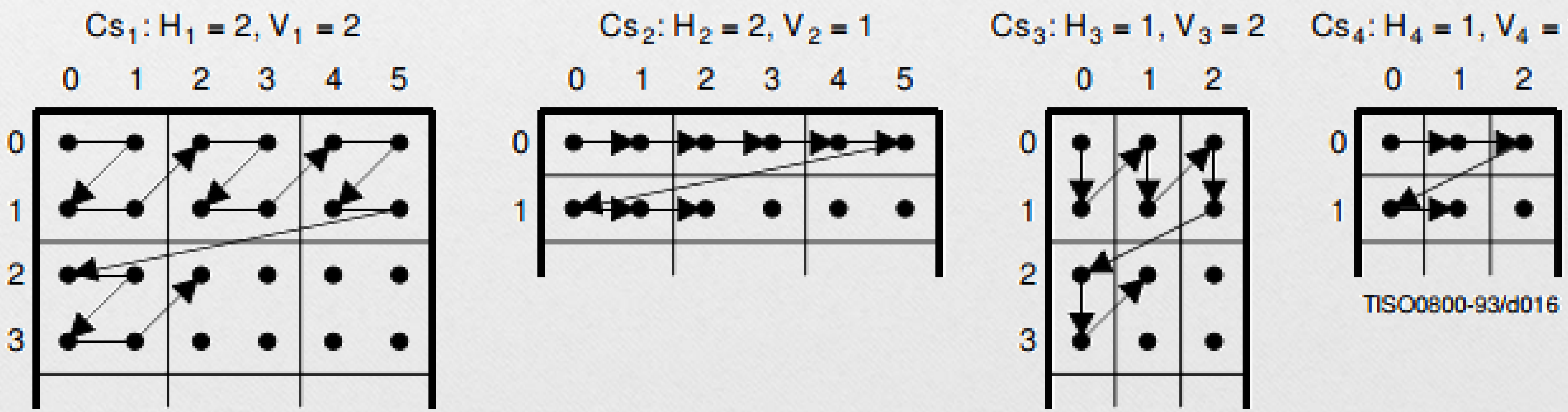
Ci: Component identifier
Hi: Horizontal sampling factor
Vi: Vertical sampling factor

Parameter	Size (bits)	Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Lf	16	8 + 3* Nf			
P	8	8	8, 12	8, 12	2-16
Y	16	0-65535			
X	16	1-65535			
Nf	8	1-255	1-255	1-4	1-255
C_i	8	0-255			
H_i	4	1-4			
V_i	4	1-4			
Tq_i	8	0-3	0-3	0-3	0-1

MCU – Minimum Coded Unit, which is comprised by blocks from each component



In 4:2:0 mode, there are 6 blocks (2*2+1*1+1*1) in an MCU.



SOF

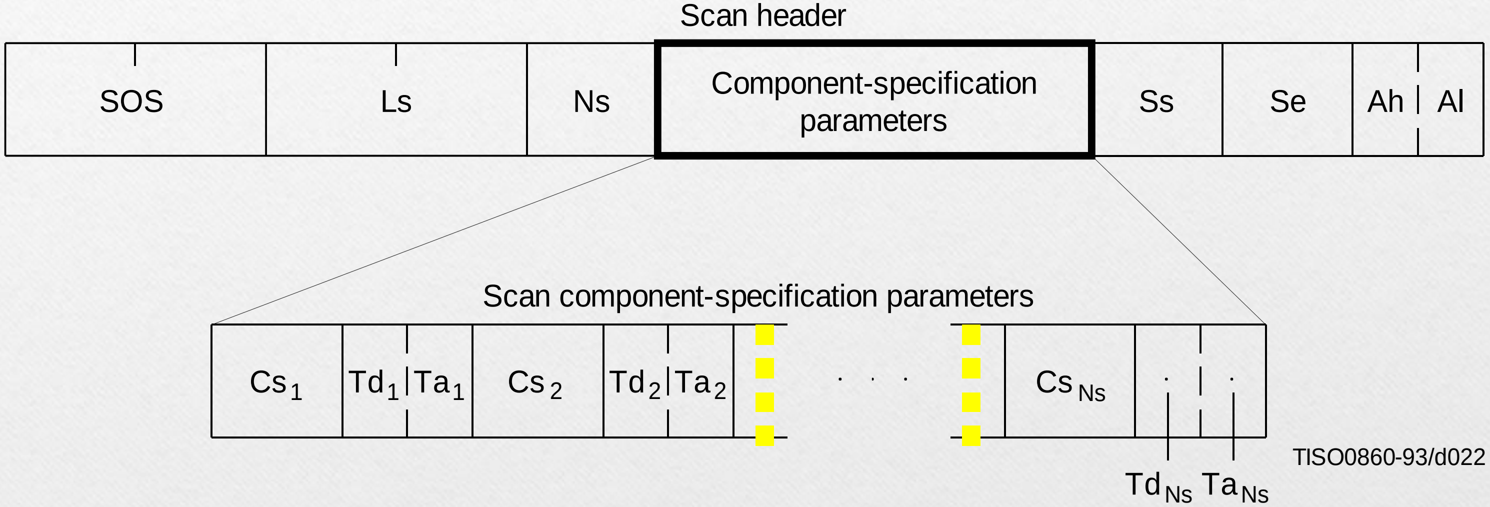
- Each component's size will be calculated by the following equation:

$$x_i = X \times \frac{H_i}{H_{\max}} \text{ and } y_i = Y \times \frac{V_i}{V_{\max}}$$

- Ex: X=512,Y=512, H_{max}=2, V_{max}=2

Component 0	H_0=2,V_0=2	→	x_0=512, y_0=512
Component 1	H_1=1,V_1=1	→	x_1=256, y_1=256
Component 2	H_2=1,V_2=1	→	x_2=256, y_2=256

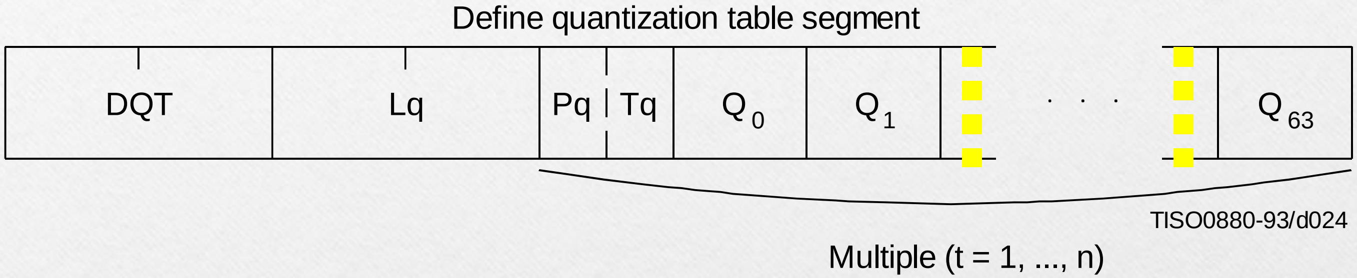
SOS



Ns: Number of image components in scan
Tdj: DC entropy coding table destination selector
Taj: AC entropy coding table destination selector

		Values			
Parameter	Size (bits)	Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Ls	16	6 + 2 * Ns			
Ns	8	1-4			
Csj	8	0-255			
Tdj	4	0-1	0-3	0-3	0-3
Taj	4	0-1	0-3	0-3	0
Ss	8	0	0	0-63	1-7
Se	8	63	63	Ss-63	0
Ah	4	0	0	0-13	0
Al	4	0	0	0-13	0-15

DQT



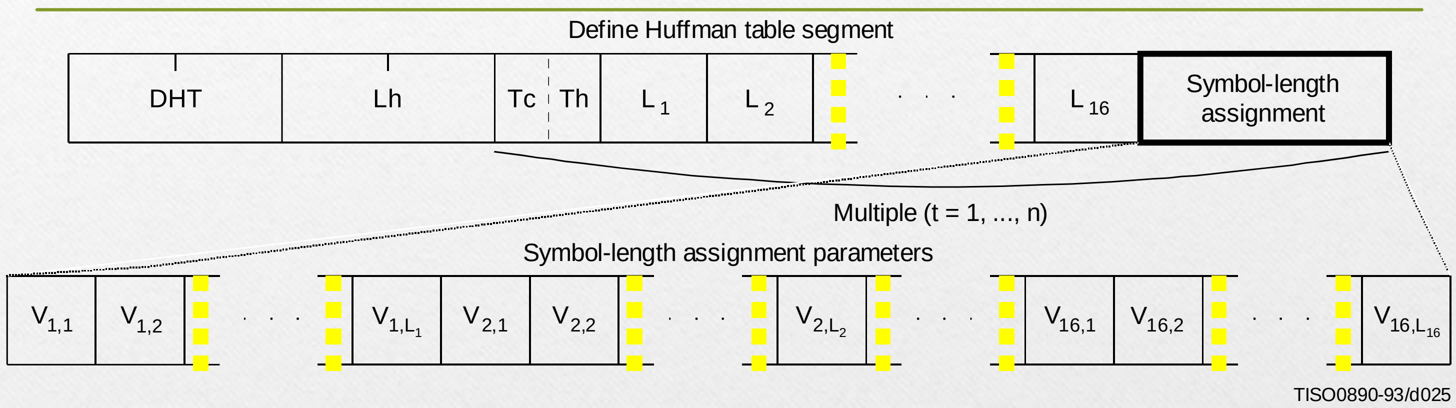
Tq: Quantization table destination identifier

- o for DC 1 for AC

Q_k: Quantization table element

		Values			
Parameter	Size (bits)	Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Lq	16				Undefined
Pq	4	0	0, 1	0, 1	Undefined
Tq	4	0-3			Undefined
Q_k	8, 16	1-255, 1-65535			Undefined

DHT



(codeword)

Tc: Table class, DC or AC?
Th: Huffman table destination identifier

(length)
Li: Number of Huffman codes of length i
Vi,j: Value associated with each Huffman code
(symbol)

$$C_1 = 00\cdots0$$

$$C_{i+1} = (C_i + 1) * 2^{p-q}$$

and

$$C_n = 11\cdots1$$

where p and q are the codeword lengths for s_i and s_{i+1}

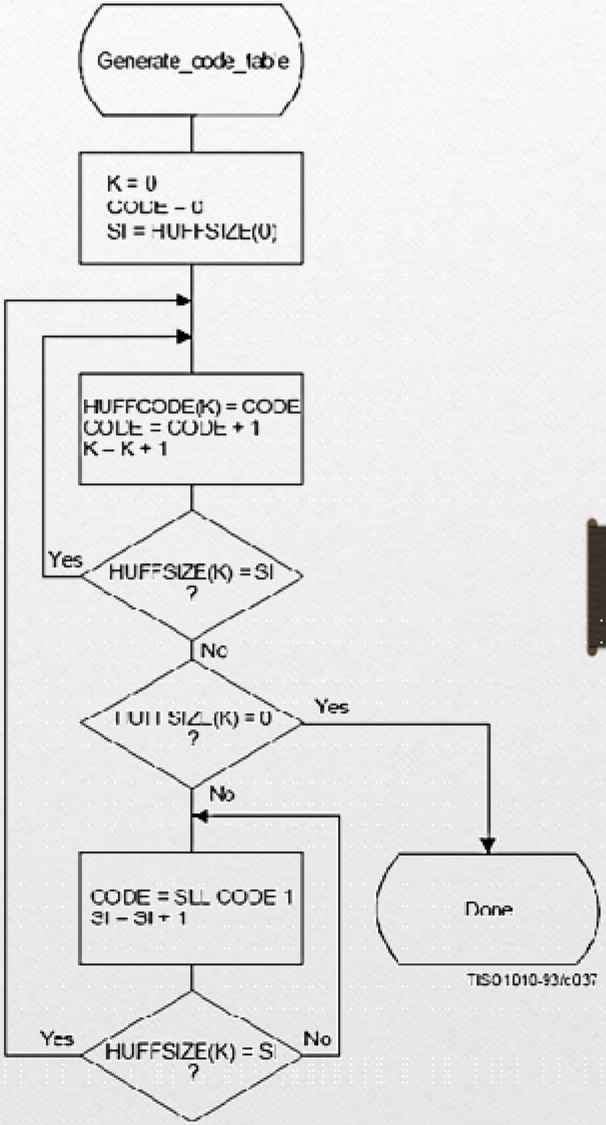
Example

Tc: 0 Th: 0

Li: 0 1 5 1 1 1 1 1 1 0 0 0 0 0 0 0

Vi,j: 0 1 2 3 4 5 6 7 8 9 10 11

Size	Codeword	Value
2	00	0
3	010	1
3	011	2
3	100	3
3	101	4
3	110	5
4	1110	6
5	11110	7
6	111110	8
7	1111110	9
8	11111110	10
9	111111110	11



(ITU-T81 Annex C)

Figure C.2 – Generation of table of Huffman codes

In common case, there are four huffman tables:

– Luminance’ s DC huffman table

– Luminance’ s AC huffman table

– Chrominance’ s DC huffman table

– Chrominance’ s AC huffman table

		Values			
Parameter	Size (bits)	Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Lh	16				
Tc	4	0, 1			0
Th	4	0, 1	0-3		
Li	8	0-255			
Vi, j	8	0-255			

DRI

- Ri – Specifies the number of MCU in the restart interval.
- After the number of MCU reaches Ri, a RST marker is inserted and the MCU counter is reset.
- You can resynchronize to the next RST marker when bit error or packet loss occurred during image transmission and the number of lost MCU can be known from the difference of RST id.

Other Marker

- APP, COM, DNL, DHP, EXP …
- Skip it by using the length field after the marker