

# JPEG Decoder

---

Baseline Sequential DCT-based

# 目錄

I. 壓縮過程

II. 文件格式

III. 解碼過程

# 目錄

I. 壓縮過程

II. 文件格式

III. 解碼過程

# Encoding Process

1. Color Space Conversion

2. Subsampling

3. Partition

4. Encoding Flow Control

Discrete Cosine Transform(DCT)

Quantization

Entropy Encoding(Huffman)



Decoding

# Encoding Process

1. Color Space Conversion

2. Subsampling

3. Partition

4. Encoding Flow Control

Discrete Cosine Transform(DCT)

Quantization

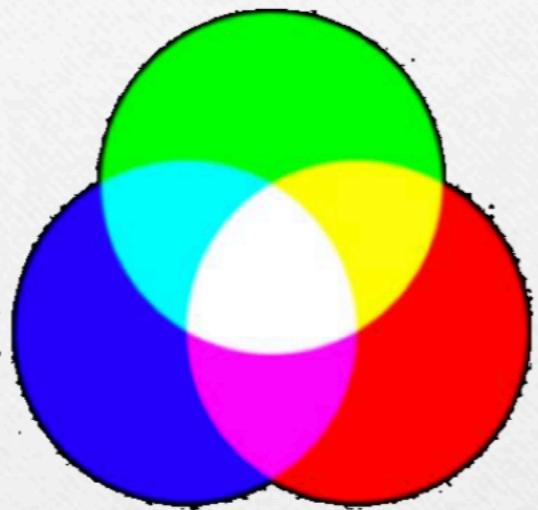
Entropy Encoding(Huffman)



Decoding

# Color Space Conversion

RGB

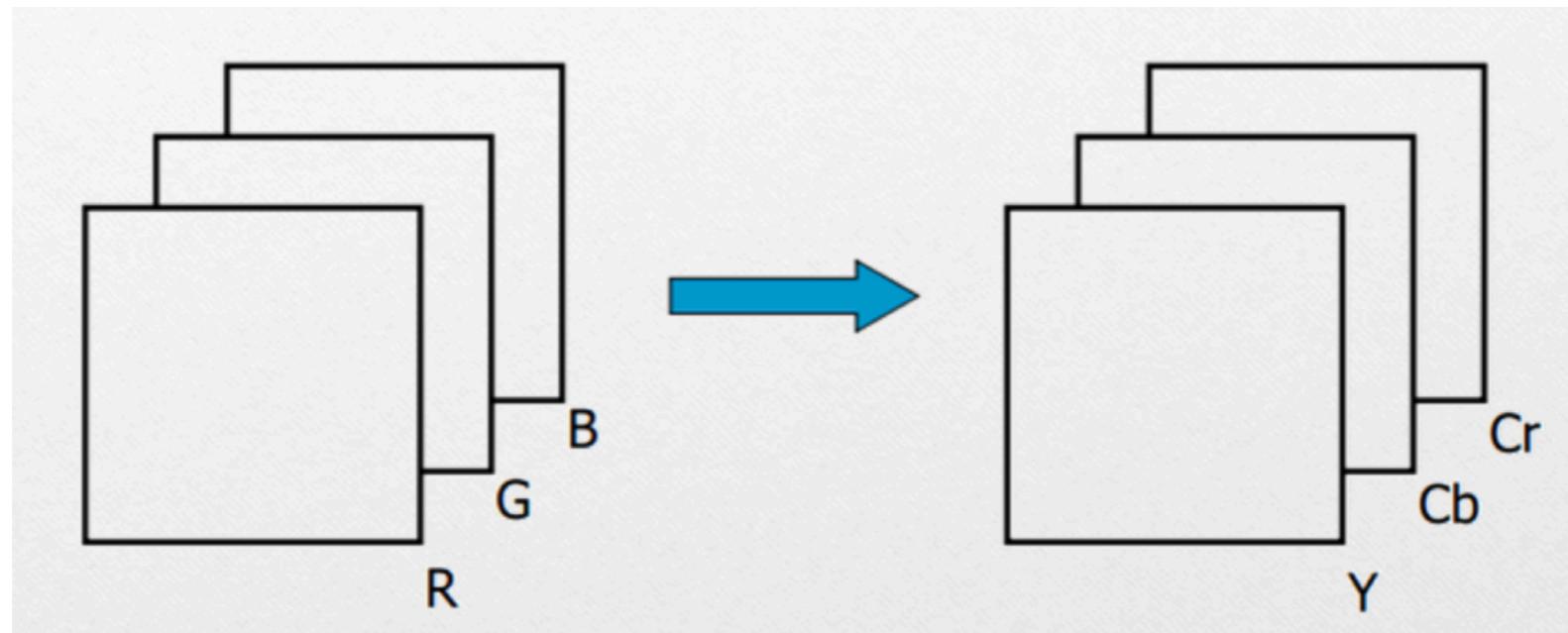


YCbCr



# Color Space Conversion

- 人類對於亮度 (Luminance) 較為敏感，而 $Y\text{C}_b\text{C}_r$ 的 $Y$ 代表了亮度；而對於彩度 (Chrominance) 比較不敏感，而 $\text{C}_b\text{C}_r$ 則代表了彩度。



$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ C_b = -0.168R - 0.331G + 0.499B \\ C_r = 0.5R - 0.419G - 0.081B \end{cases}$$

# Encoding Process

1. Color Space Conversion

2. Subsampling

3. Partition

4. Encoding Flow Control

Discrete Cosine Transform(DCT)

Quantization

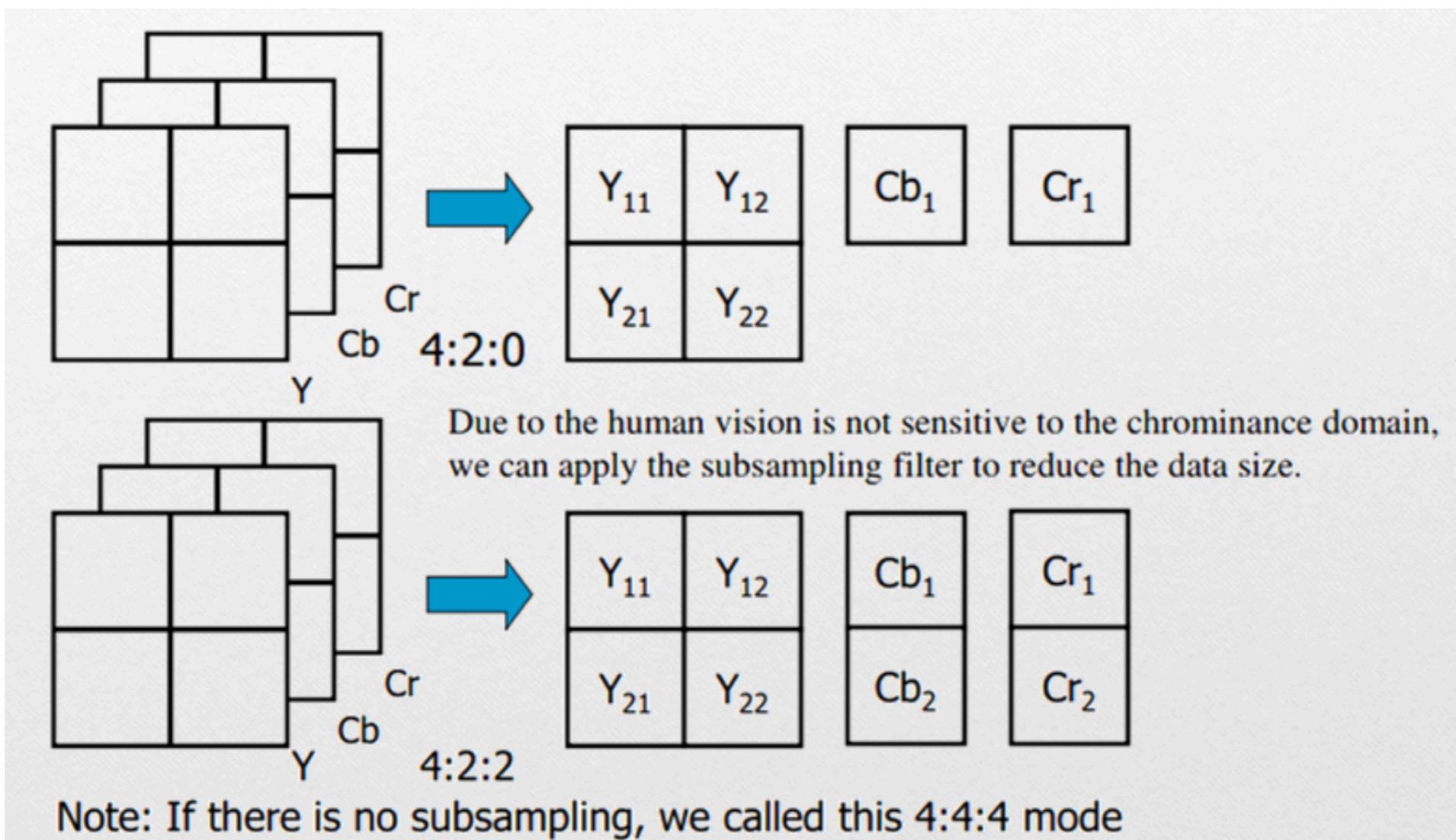
Entropy Encoding(Huffman)



Decoding

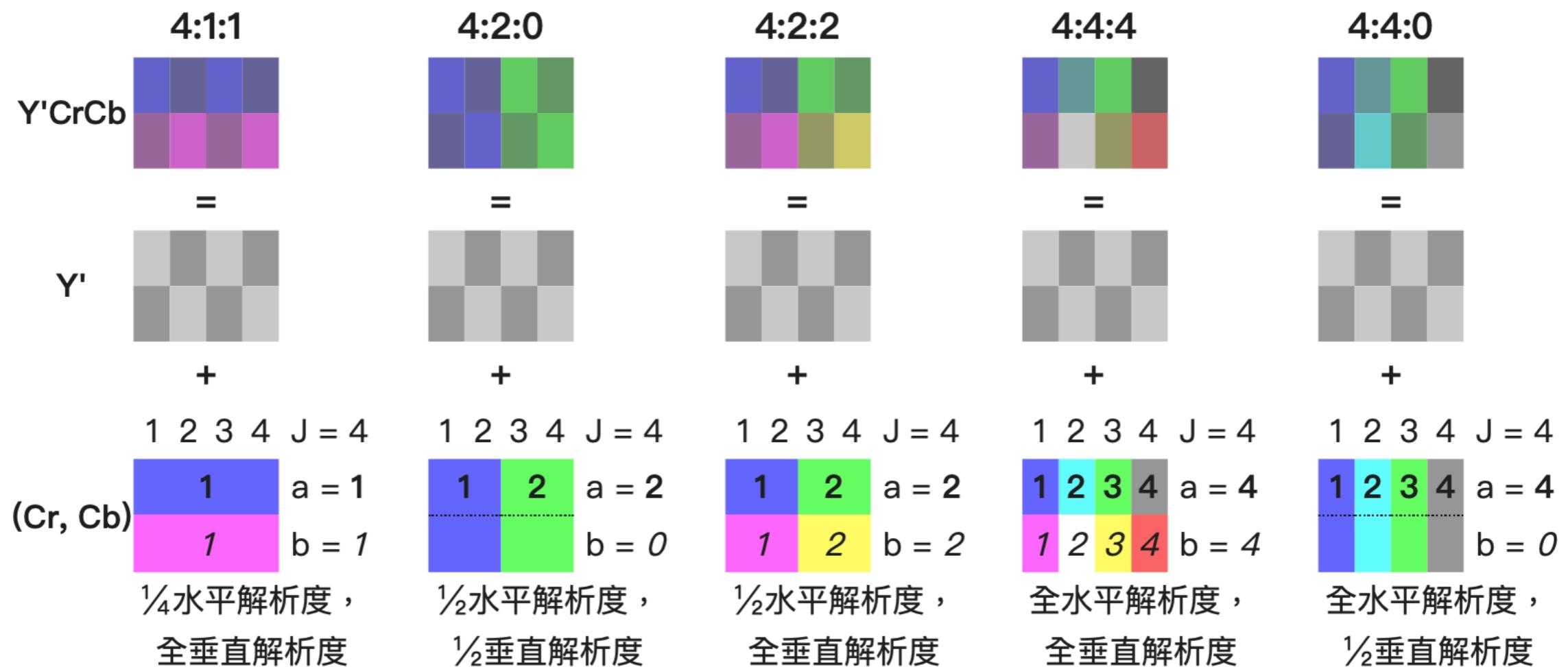
# Subsampling

- 因為人類視覺對彩度 (Chrominance) 較不敏感，可以透過 subsampling來減少data size
- 常見的有4:4:4、4:1:1、4:2:2、4:2:0



# Subsampling

- J : 水平抽樣參照（概念上區域的寬度）。通常為4。
  - a : 在J個像素第一行中的色度抽樣數目 (Cr, Cb)。
  - b : 在J個像素第二行中的額外色度抽樣數目 (Cr, Cb)。



# Encoding Process

1. Color Space Conversion

2. Subsampling

3. Partition

4. Encoding Flow Control

Discrete Cosine Transform(DCT)

Quantization

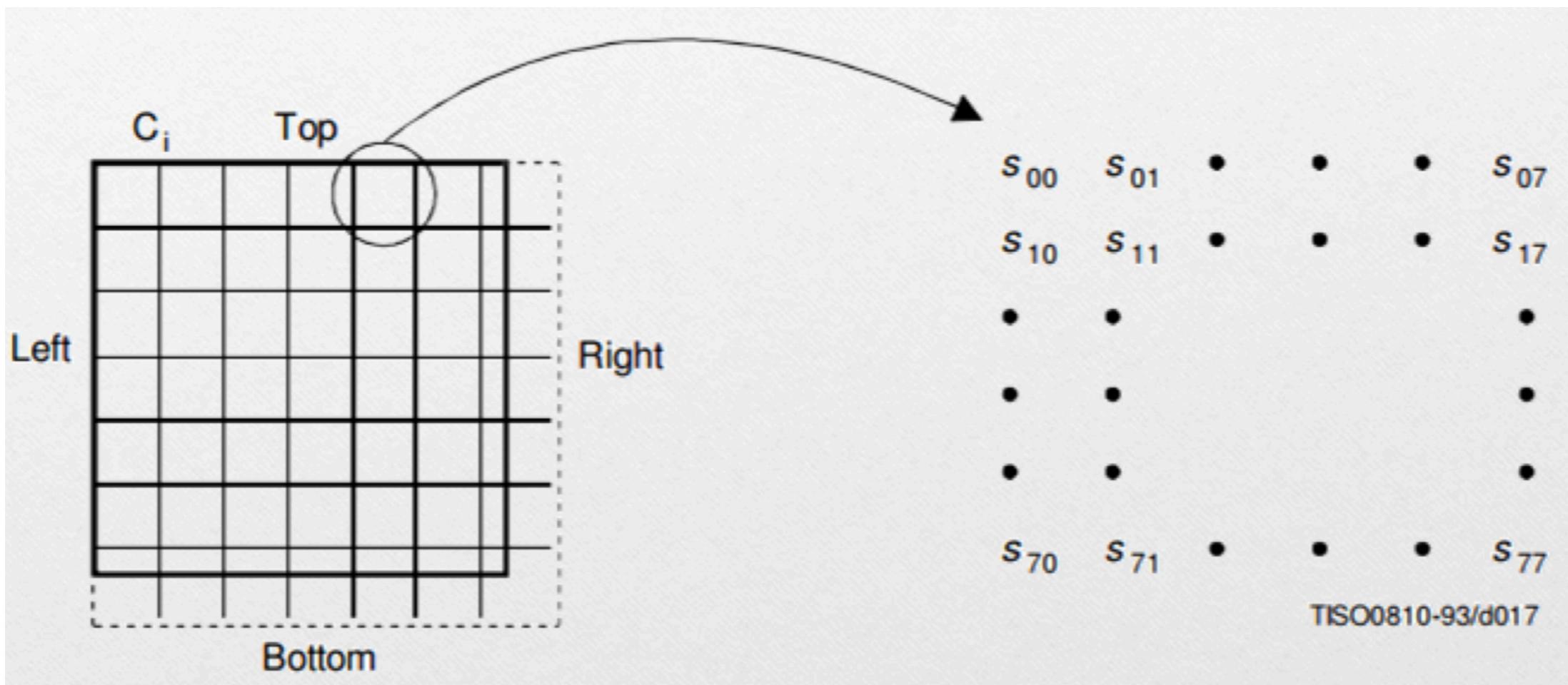
Entropy Encoding(Huffman)



Decoding

# Partition

- 因為JPEG是採用Block Coding的方式來完成的，因此經過 subsampling後，將影像切割成一個個8x8大小的區塊。



# Encoding Process

1. Color Space Conversion

2. Subsampling

3. Partition

4. Encoding Flow Control

Discrete Cosine Transform(DCT)

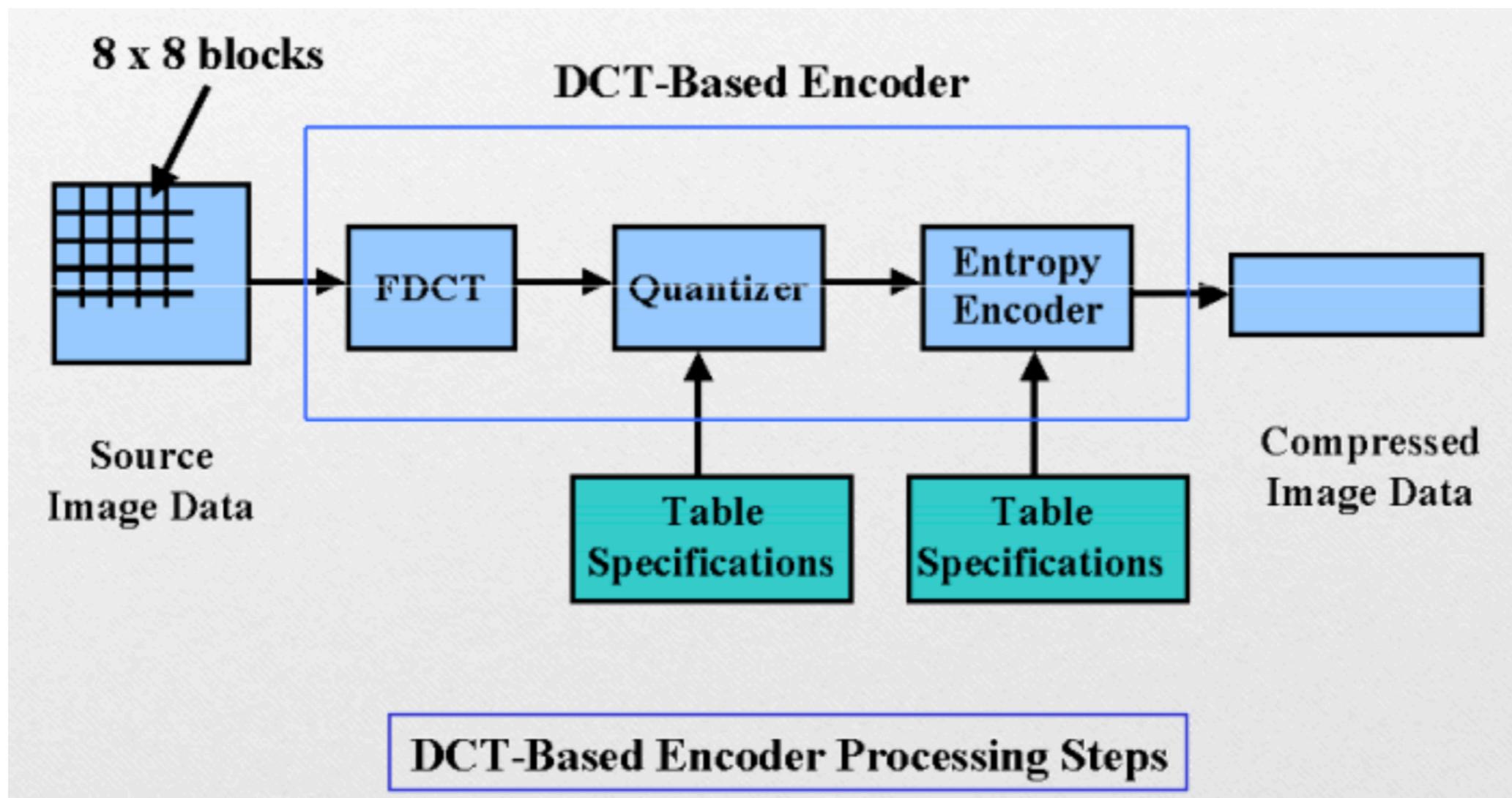
Quantization

Entropy Encoding(Huffman)

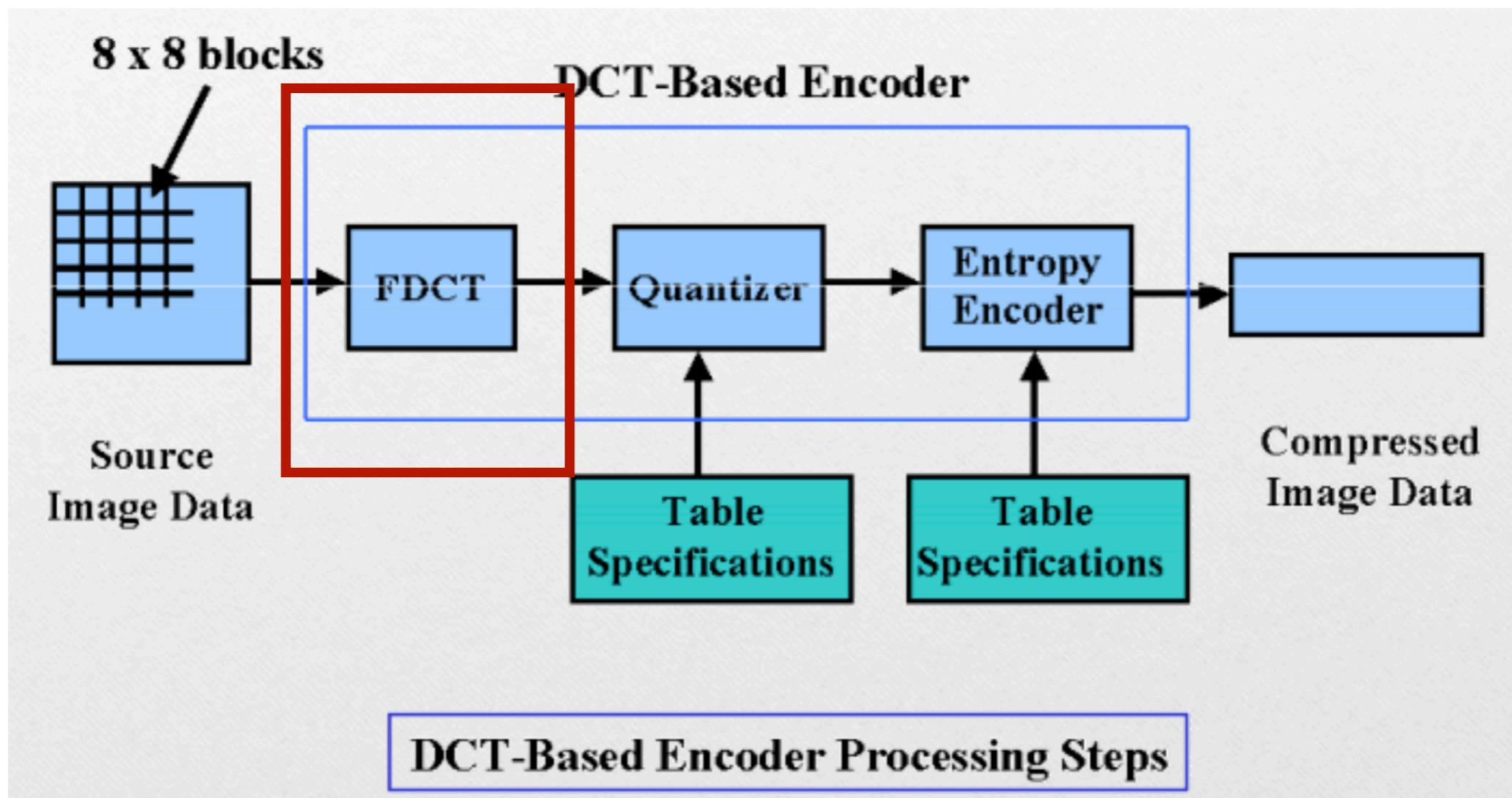


Decoding

# Encoding Flow Control



# Encoding Flow Control



# Forward Discrete Cosine Transform

- 將像素、空間域數位影像資料轉換為頻率域的值
- 因為人類對於低頻的訊號較為敏感，在下一步中會將轉換好的值進行量化

$$X_{k_1, k_2} = \frac{2}{N} c(k_1) c(k_2) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x_{n_1, n_2} \cdot \cos \frac{(2n_1 + 1)k_1\pi}{2N} \cos \frac{(2n_2 + 1)k_2\pi}{2N}$$

$$n_1, n_2, k_1, k_2 = 0, 1, \dots, N-1$$

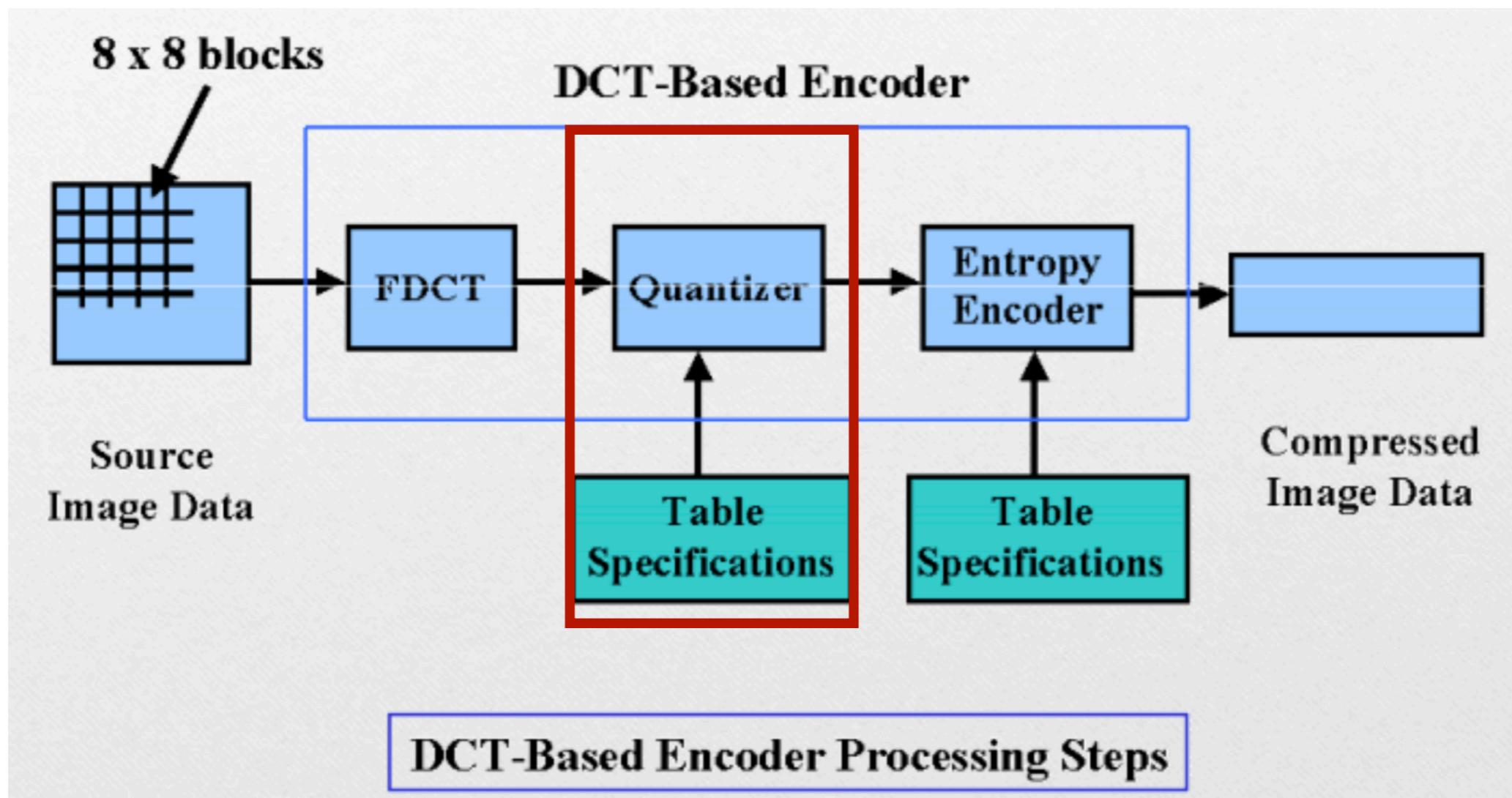
where  $c(0) = \frac{1}{\sqrt{2}}$  and  $c(n) = 1$  for  $n \neq 0$

$$x_{n_1, n_2} = \frac{2}{N} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} c(k_1) c(k_2) X_{k_1, k_2} \cdot \cos \frac{(2n_1 + 1)k_1\pi}{2N} \cos \frac{(2n_2 + 1)k_2\pi}{2N}$$

$$n_1, n_2, k_1, k_2 = 0, 1, \dots, N-1$$

where  $c(0) = \frac{1}{\sqrt{2}}$  and  $c(n) = 1$  for  $n \neq 0$

# Encoding Flow Control



# Quantization

- 希望透過量化手法，在影像品質能接受的情況下，將不重要的訊息予以忽略
- 因為人類對於低頻的訊號較為敏感，此程序會將高頻值大幅量化為0，盡量只保留低頻的值。

$$X_{k_1, k_2} = \frac{2}{N} c(k_1) c(k_2) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x_{n_1, n_2} \cdot \cos \frac{(2n_1+1)k_1\pi}{2N} \cos \frac{(2n_2+1)k_2\pi}{2N}$$

$$n_1, n_2, k_1, k_2 = 0, 1, \dots, N-1$$

where  $c(0) = \frac{1}{\sqrt{2}}$  and  $c(n) = 1$  for  $n \neq 0$

$$x_{n_1, n_2} = \frac{2}{N} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} c(k_1) c(k_2) X_{k_1, k_2} \cdot \cos \frac{(2n_1+1)k_1\pi}{2N} \cos \frac{(2n_2+1)k_2\pi}{2N}$$

$$n_1, n_2, k_1, k_2 = 0, 1, \dots, N-1$$

where  $c(0) = \frac{1}{\sqrt{2}}$  and  $c(n) = 1$  for  $n \neq 0$

# Quantization

- 希望透過量化手法，在影像品質能接受的情況下，將不重要的訊息予以忽略
- 因為人類對於低頻的訊號較為敏感，此程序會將高頻值大幅量化為0，盡量只保留低頻的值。

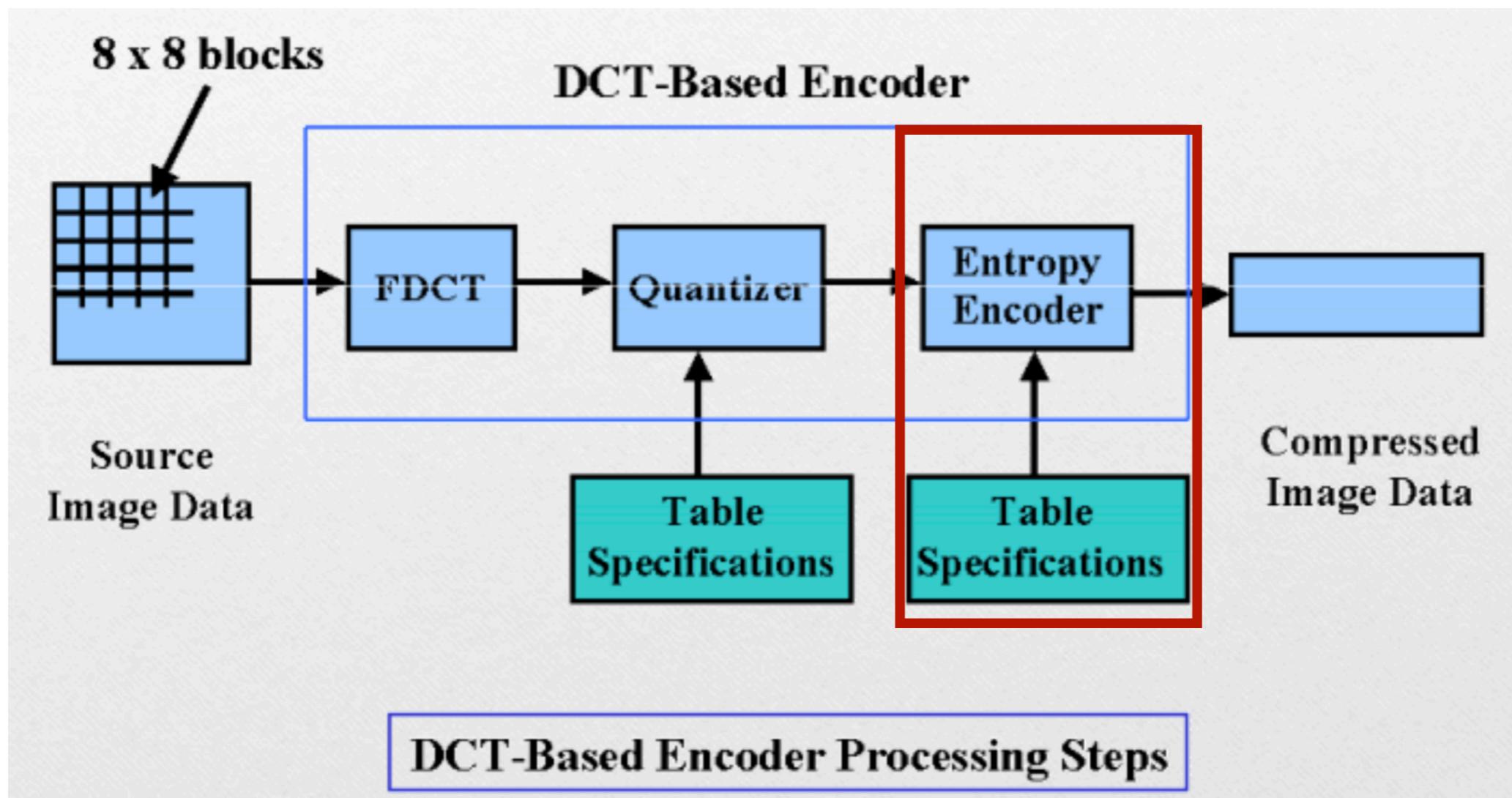
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	67	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Luminance 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

Chrominance Quantization Table

# Encoding Flow Control



# Entropy Encoder

經過量化後，可將信號分成DC值與AC值。DC值為8x8區塊中最左上角的值。AC值為剩下的63個。

- DC — Differential Pulse Code Modulation : DPCM
- AC — Run-Length pair  
—> Huffman encoding

$$X_{k_1, k_2} = \frac{2}{N} c(k_1) c(k_2) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x_{n_1, n_2} \cdot \cos \frac{(2n_1+1)k_1\pi}{2N} \cos \frac{(2n_2+1)k_2\pi}{2N}$$
$$n_1, n_2, k_1, k_2 = 0, 1, \dots, N-1$$

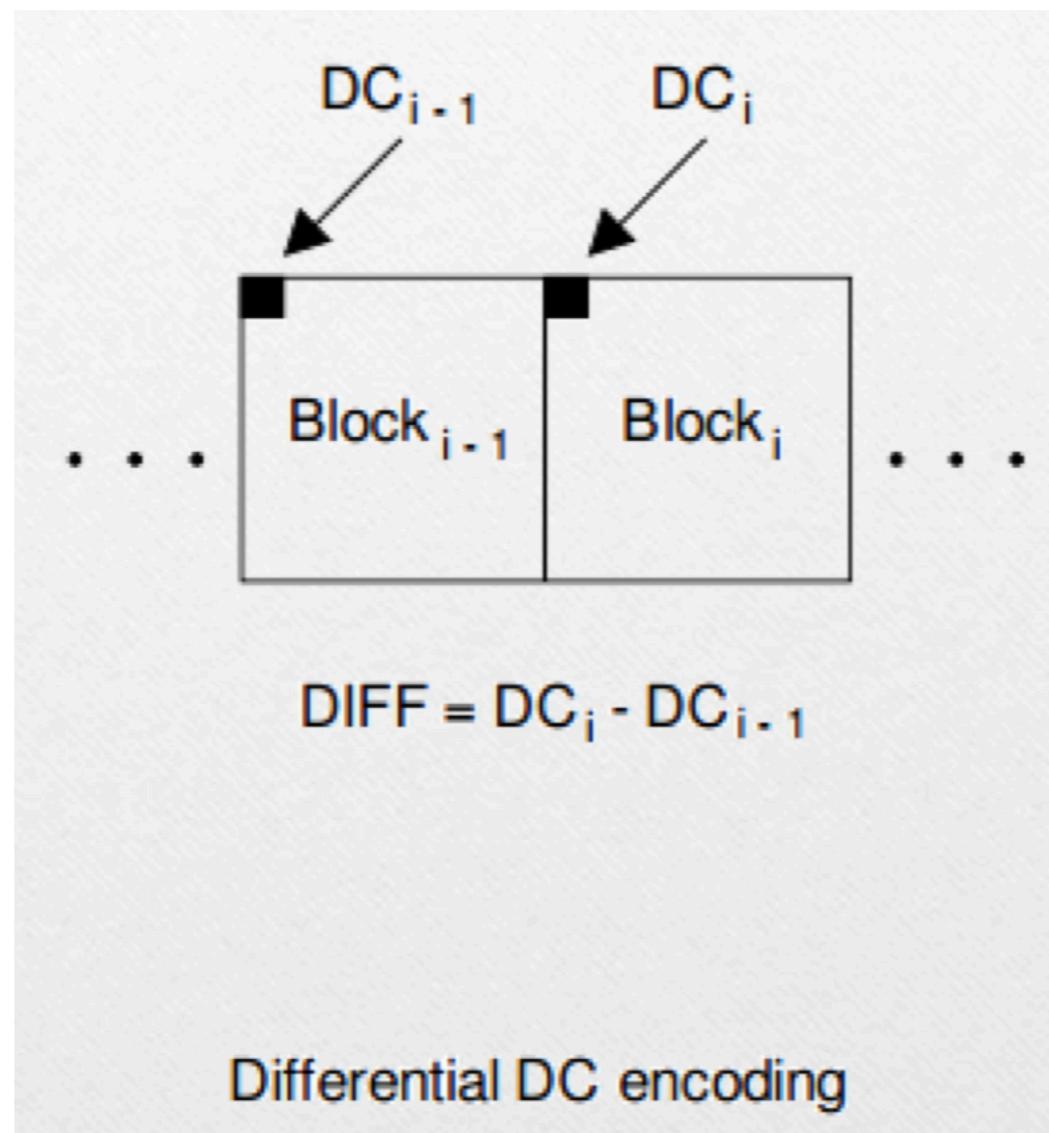
$$\text{where } c(0) = \frac{1}{\sqrt{2}} \text{ and } c(n) = 1 \text{ for } n \neq 0$$

$$x_{n_1, n_2} = \frac{2}{N} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} c(k_1) c(k_2) X_{k_1, k_2} \cdot \cos \frac{(2n_1+1)k_1\pi}{2N} \cos \frac{(2n_2+1)k_2\pi}{2N}$$
$$n_1, n_2, k_1, k_2 = 0, 1, \dots, N-1$$

$$\text{where } c(0) = \frac{1}{\sqrt{2}} \text{ and } c(n) = 1 \text{ for } n \neq 0$$

# DPCM

- 由於每個DC值都很接近，因此，與其存所有DC值，不如存前後兩兩DC值的差值。



# DPCM

- Huffman encoding

SSSS	Value
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
⋮	⋮
⋮	⋮
⋮	⋮

Category	Code Word
0	00
1	010
2	011
3	100
4	101
⋮	⋮

# DPCM

- Huffman encoding

Ex:Diff = 3

SSSS	Value
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
:	:
:	:
:	:

Category	Code Word
0	00
1	010
2	011
3	100
4	101
.....	.....

# DPCM

- Huffman encoding

Ex:Diff = 3

SSSS	Value
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
:	:
:	:
:	:

Category	Code Word
0	00
1	010
2	011
3	100
4	101
.....	.....

# DPCM

- Huffman encoding

Ex:Diff = 3

SSSS	Value
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
:	:
:	:
:	:

Category	Code Word
0	00
1	010
2	011
3	100
4	101
.....	.....

# DPCM

- Huffman encoding

Ex:Diff = 3

SSSS	Value
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
:	:
:	:
:	:

Category	Code Word
0	00
1	010
2	011
3	100
4	101
.....	.....

# DPCM

- Huffman encoding

Ex:Diff = 3

SSSS	Value
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
:	:
:	:
:	:

Category	Code Word
0	00
1	010
2	01111
3	100
4	101
.....	.....

# DPCM

- Huffman decoding

Example: 1011111.....

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

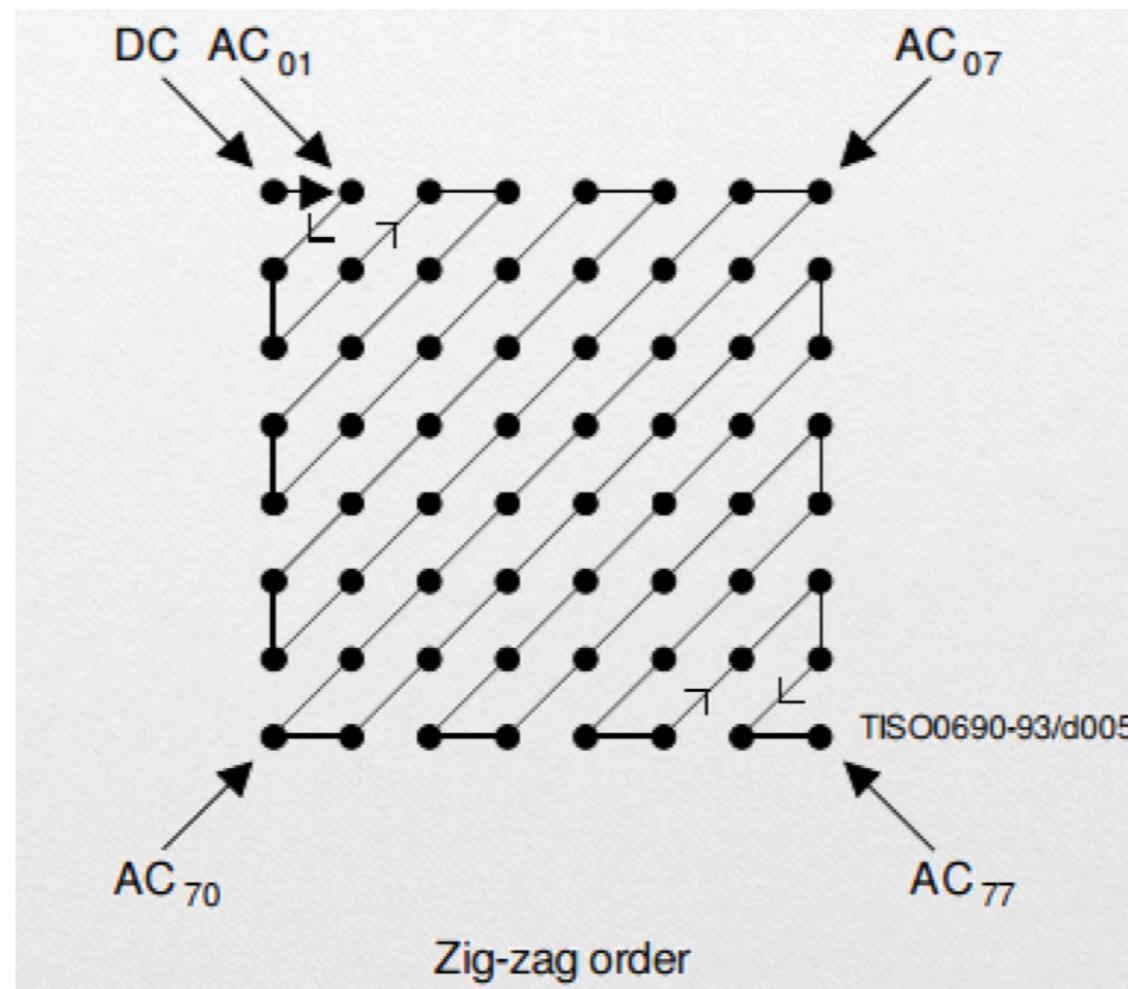
SSSS	DIFF values
0	0
1	-1, 1
2	-3, -2, 2, 3
3	-7..-4, 4..7
4	-15..-8, 8, 15
5	-31..-16, 16..31
....	....

Huffman decoding

2. Get T bits  $\rightarrow$  Get  $1111_2 \rightarrow \text{DIFF} = 1111_2$
3. Extend DIFF  $\rightarrow$  Get 15

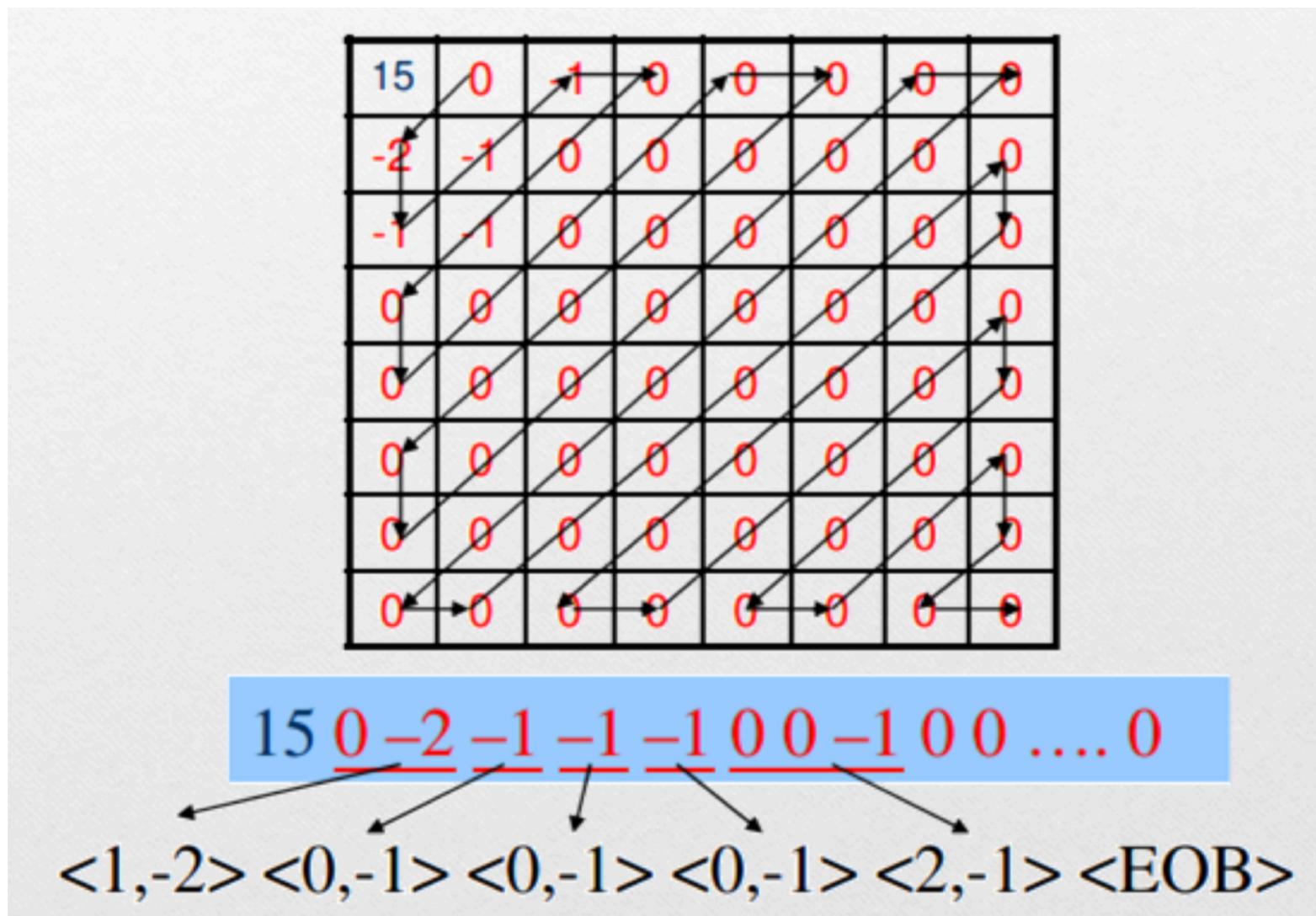
# Run-Length pair

- 因為經過FDCT與量化後，訊號從左上角到右下角呈現的頻率會越來越高，而Zig-Zag掃描方式則會從低頻逐漸掃描至高頻。
- 而越高頻的訊號，值為零的機率越高。因此可透過Zig-Zag掃描方式讓零連續出現，再利用run-length pair達到最高的壓縮率。



# Run-Length pair

- Run-length coding的編碼方式為： $(R, L) = (\text{出現}0\text{的數目}, \text{非}0\text{的值})$



# Run-Length pair

- Huffman Coding

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

$$\text{AC code word} = \text{Run/Size code word} + \text{AC coefficient code word}$$

$\langle 1, -2 \rangle$	$\langle 0, -1 \rangle$	$\langle 0, -1 \rangle$	$\langle 0, -1 \rangle$	$\langle 2, -1 \rangle$	$\langle \text{EOB} \rangle$
11011 01	00 0	00 0	00 0	11100 0	1010

# Run-Length pair

- Huffman Coding

Example: 1101101.....

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

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

2. Get S bits  $\rightarrow SSSS = 01_2$

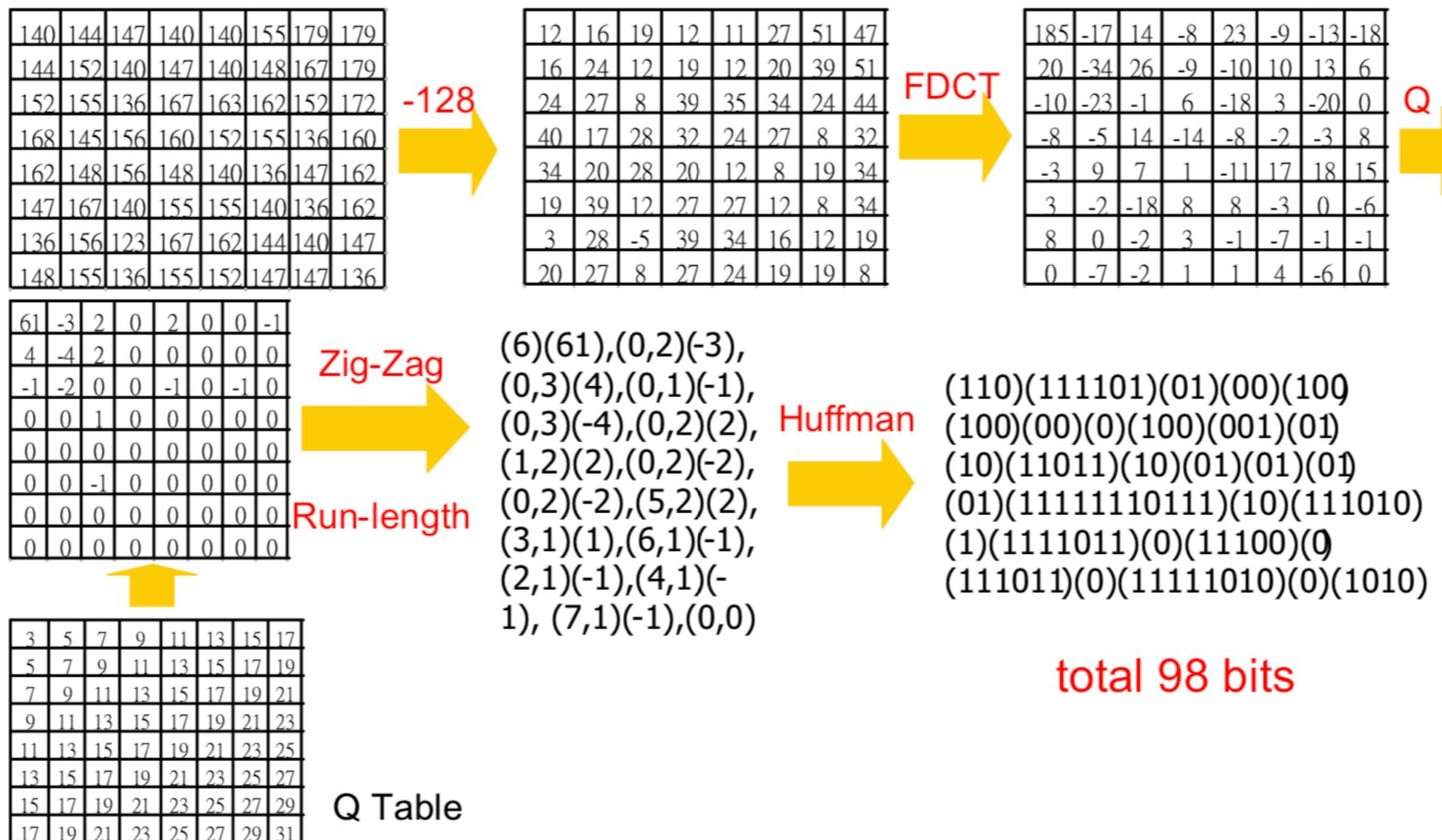
Huffman decoding

3. Extend 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)

# Summary



# 目錄

I. 壓縮過程

II. 文件格式

III. 解碼過程

# 文件格式

JPEG文件大致方為兩個部分

## 1.Marker

為兩個字節所構成，第一個字節固定為0xFF，可以在標記碼的前面加入不限數量的0xFF，但只算一次，用來代表後一個字節的起始。而後一字節則根據不同的意義有不同的數值。

## 2.Segment

標記碼所對應的壓縮數據，記載關於文件的訊息。

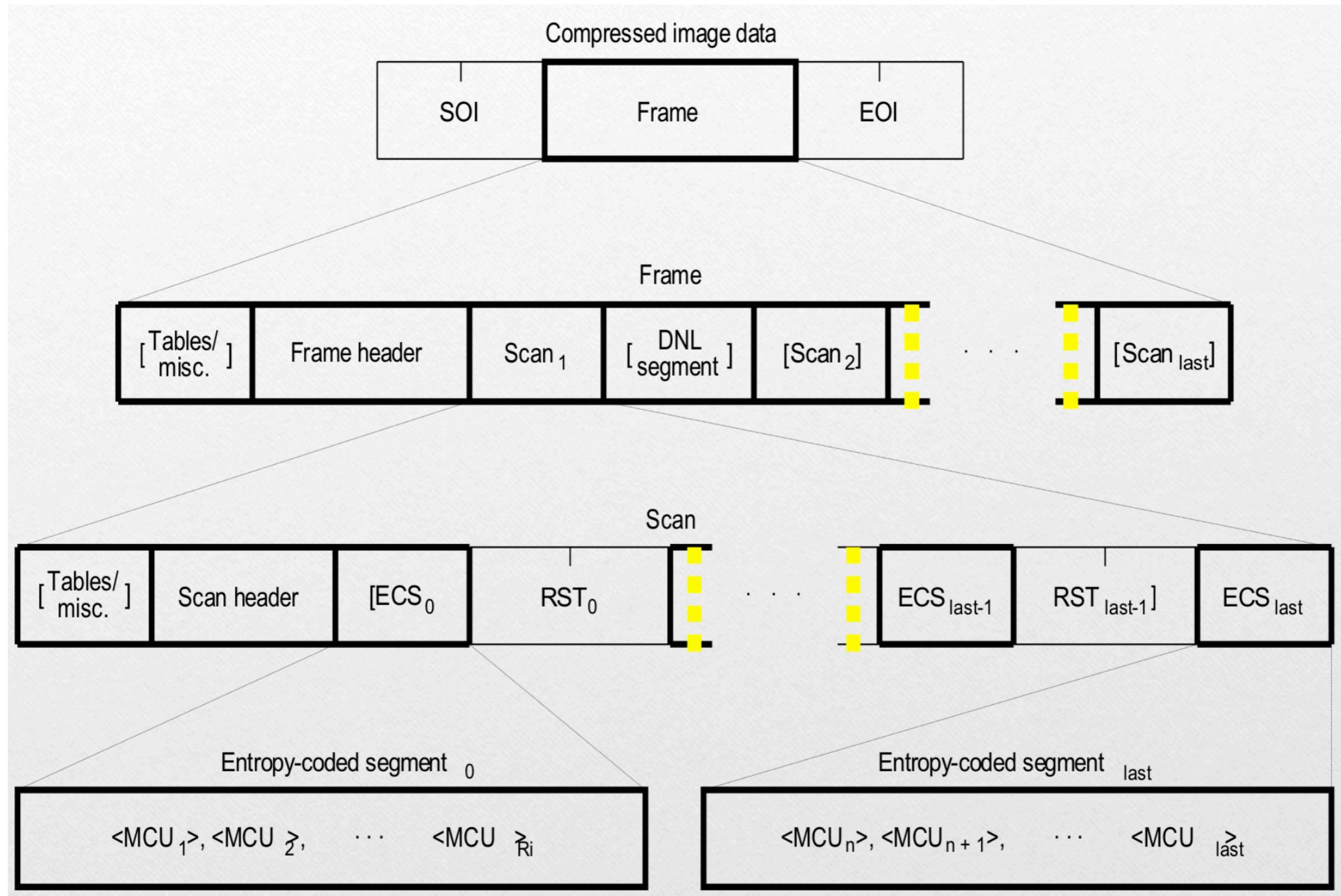
# Header Structure



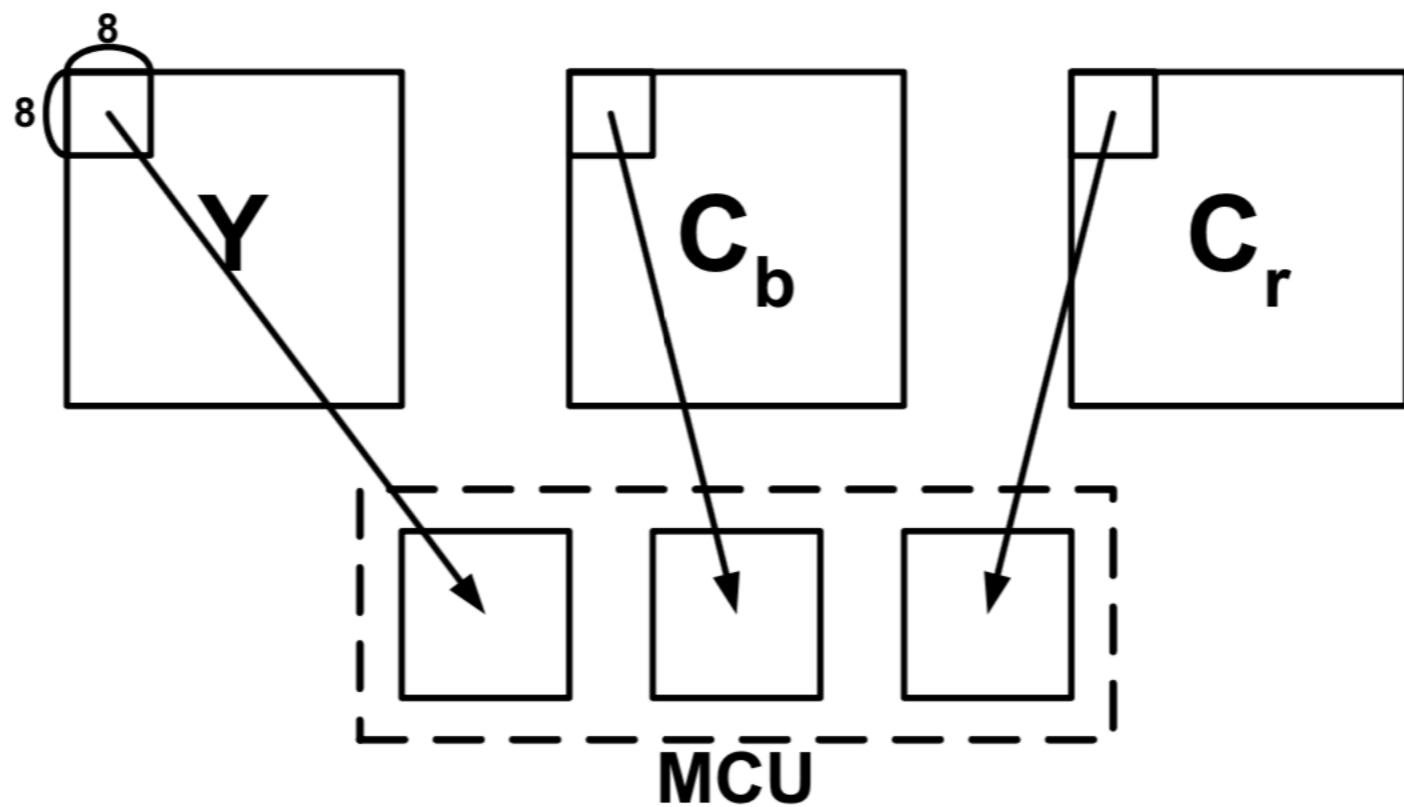
# 文件格式

- 1.SOI(Start of Image，圖像開始)
- 2.APP0(Application，應用程序保留標記0)
- 3.APPn(Application，應用程序保留標記n，n可為1~15)
- 4.DQT(Define Quantization Table，定義量化表)
- 5.SOF0(Start of Frame，Frame圖像開始)
- 6.DHT(Define Huffman Table，定義Hoffman table)
- 7.DRI(Define Restart Interval，定義差分編碼累計復位間隔)
- 8.SOS(Start of Scan，掃描開始十二字節)
- 9.EIO(End of Image，圖像結束，兩個字節)

# 文件格式



# MCU



# 目錄

I. 概觀

II. 文件格式

III. 解碼過程

# SOI

Tag：兩個字節，固定為0xFFD8。

# APPO

Tag：兩個字節，固定為0xFFE0。

包含九個具體字段：

1. 數據長度，2個字節，代表1~9的總長度。
2. 標示符，5個字節，固定為0x4A46494600。
3. 版本號，2個字節。
4. X和Y的密度單位，1個字節，只有三個可選（0:無單位、1:英吋、2:釐米）。
5. X方向像素密度，2個字節。
6. Y方向像素密度，2個字節。
7. 縮圖水平像素數目，1個字節。
8. 縮圖垂直像素數目，1個字節。
9. 縮圖RGB位圖。

# DQT

Tag：兩個字節，固定為0xFFDB

包含九個具體字段：

1. 數據長度，2個字節。

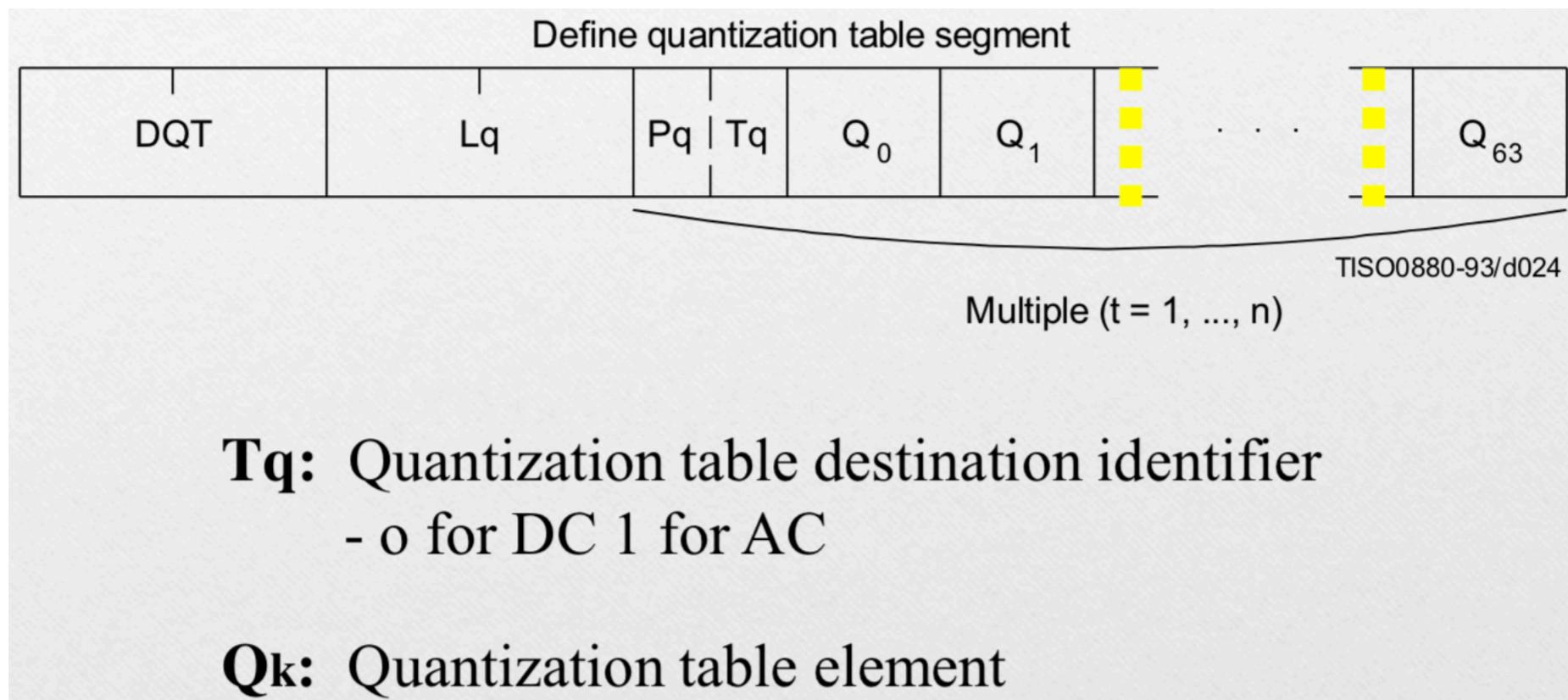
2. 量化表，2個字節：

a. 精度及量化表ID，高四位為精度（只有0:8位、1:16位）、低四位為ID

b. 表項

量化表可重複出現。

# DQT



# DQT

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

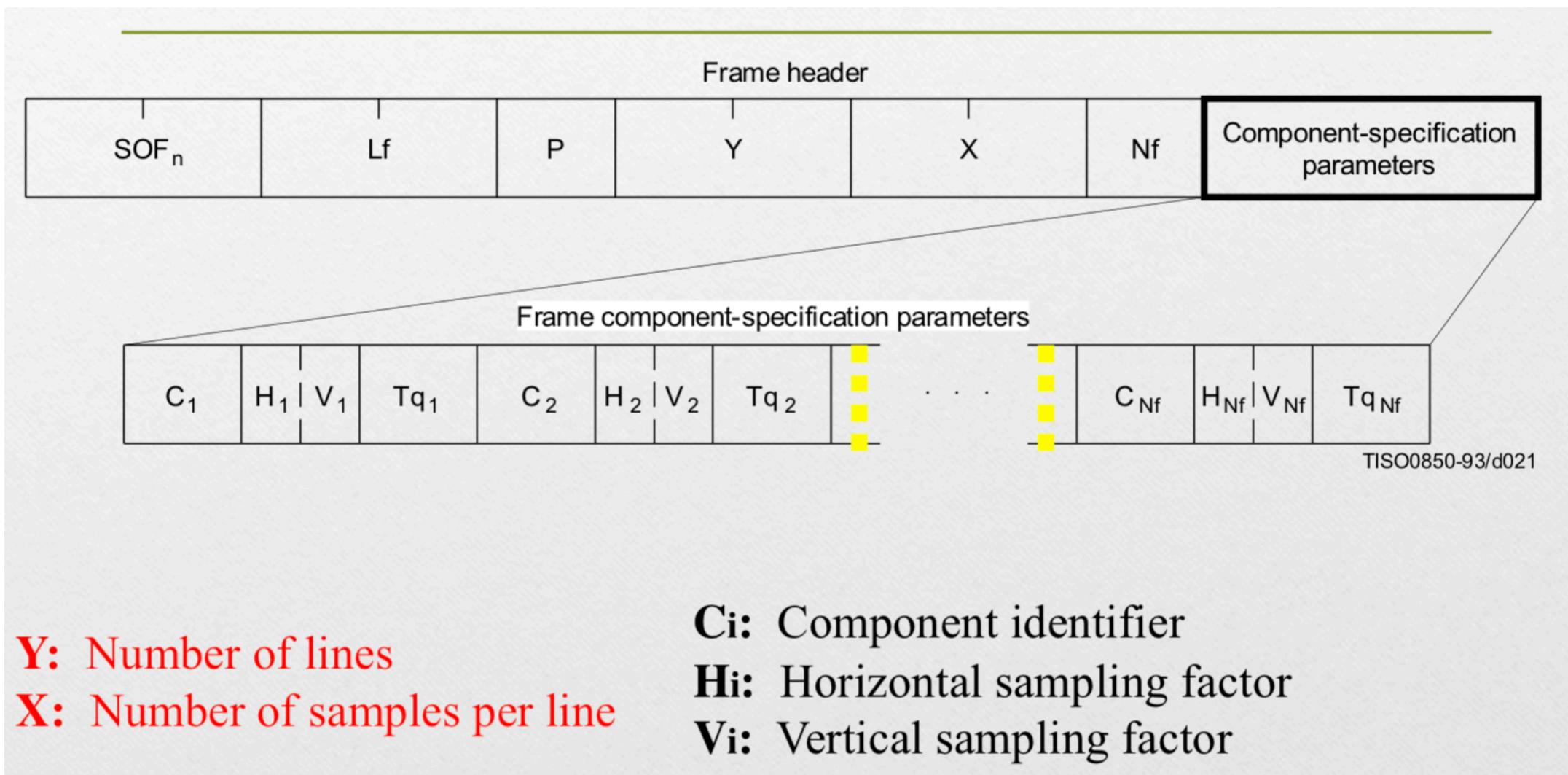
# SOFO

Tag：兩個字節，固定為0xFFC0

包含九個具體字段：

1. 數據長度，2個字節，代表1~9的總長度。
  2. 精度，1個字節，各個數據樣本的位數。
  3. 圖像高度，2個字節，單位為像素。
  4. 圖像寬度，2個字節，單位為像素。
  5. 顏色分量數，1個字節，只有三種可選（1:灰度值、3:YCrCb或YIQ、4:CMYK）。
  6. 顏色分量訊息，分量數\*3個字節：
    - a. 顏色分量ID，1個字節。
    - b. 水平/垂直採樣因子，1個字節，高四位為水平、低四位為垂直。
    - c. 量化表，1個字節，當前使用的量化表ID。
- 此處，6會重複出現，一般為三次。

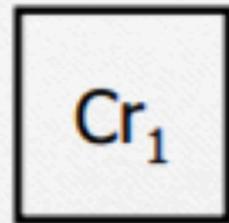
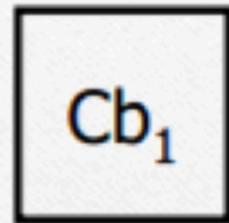
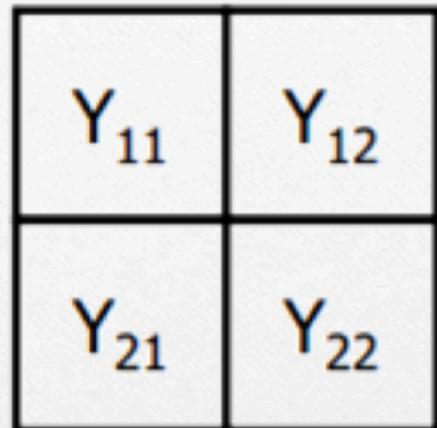
# SOFO



# SOFO

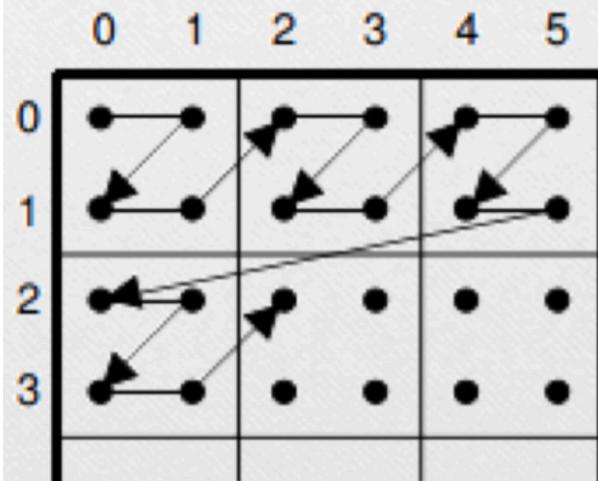
Parameter	Size (bits)	Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Lf	16	$8 + 3 * Nf$			
P	8	8	8, 12	8, 12	2-16
Y	16	0-65535			
X	16	1-65535			
Nf	8	1-255	1-255	1-4	1-255
Ci	8	0-255			
Hi	4	1-4			
Vi	4	1-4			
Tqi	8	0-3	0-3	0-3	0-1

# SOFO



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

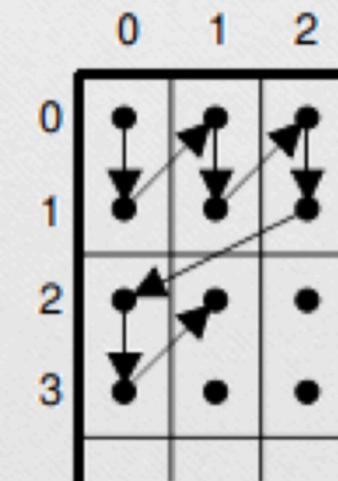
$Cs_1: H_1 = 2, V_1 = 2$



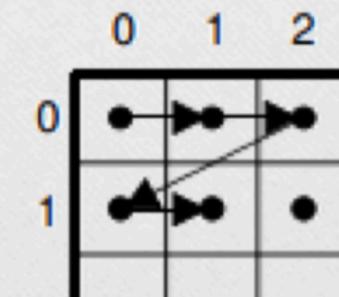
$Cs_2: H_2 = 2, V_2 = 1$



$Cs_3: H_3 = 1, V_3 = 2$



$Cs_4: H_4 = 1, V_4 = 1$



TISO0800-93/d016

# DHT

Tag：兩個字節，固定為0xFFC4

包含兩個具體字段：

1. 數據長度，2個字節，代表1~2的總長度。

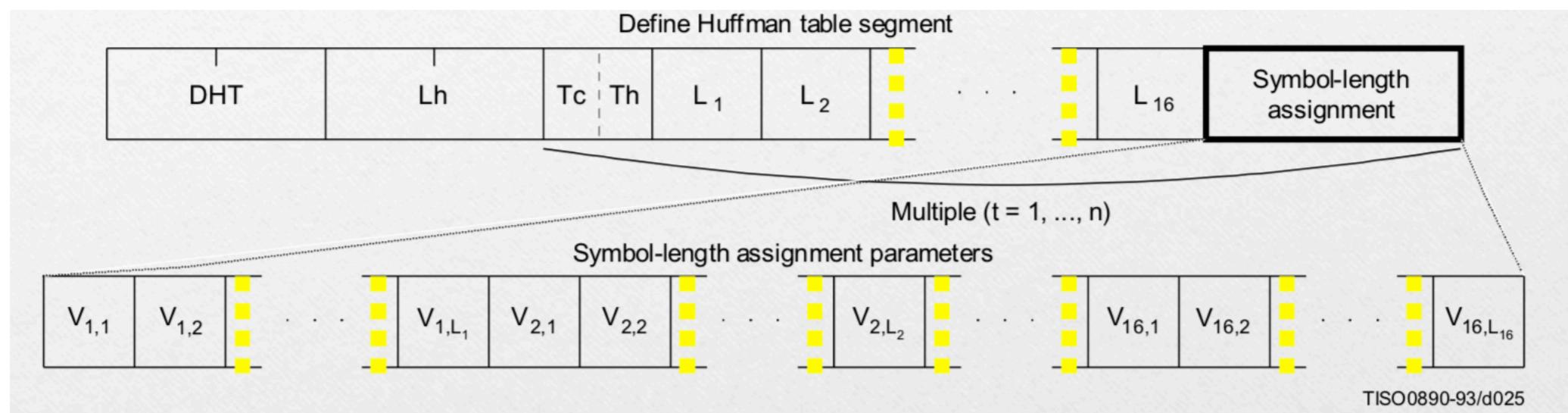
2. Hoffman Table：

a. 表ID、表類型，1個字節，高四位為類型（0:DC、1:AC） 、低四位為ID。

b. 不同位數的碼字數量，16字節。

c. 編碼內容，字節為16個不同位數的碼字數量之和。

# DHT



# DRI

Tag：兩個字節，固定為0xFFDD

包含兩個具體字段：

1. 數據長度，2個字節，代表1~2的總長度。

2. MCU塊的單元中的重新開始間隔，2個字節，若設為n則代表每n個MCU塊就有一個RSTn標記。

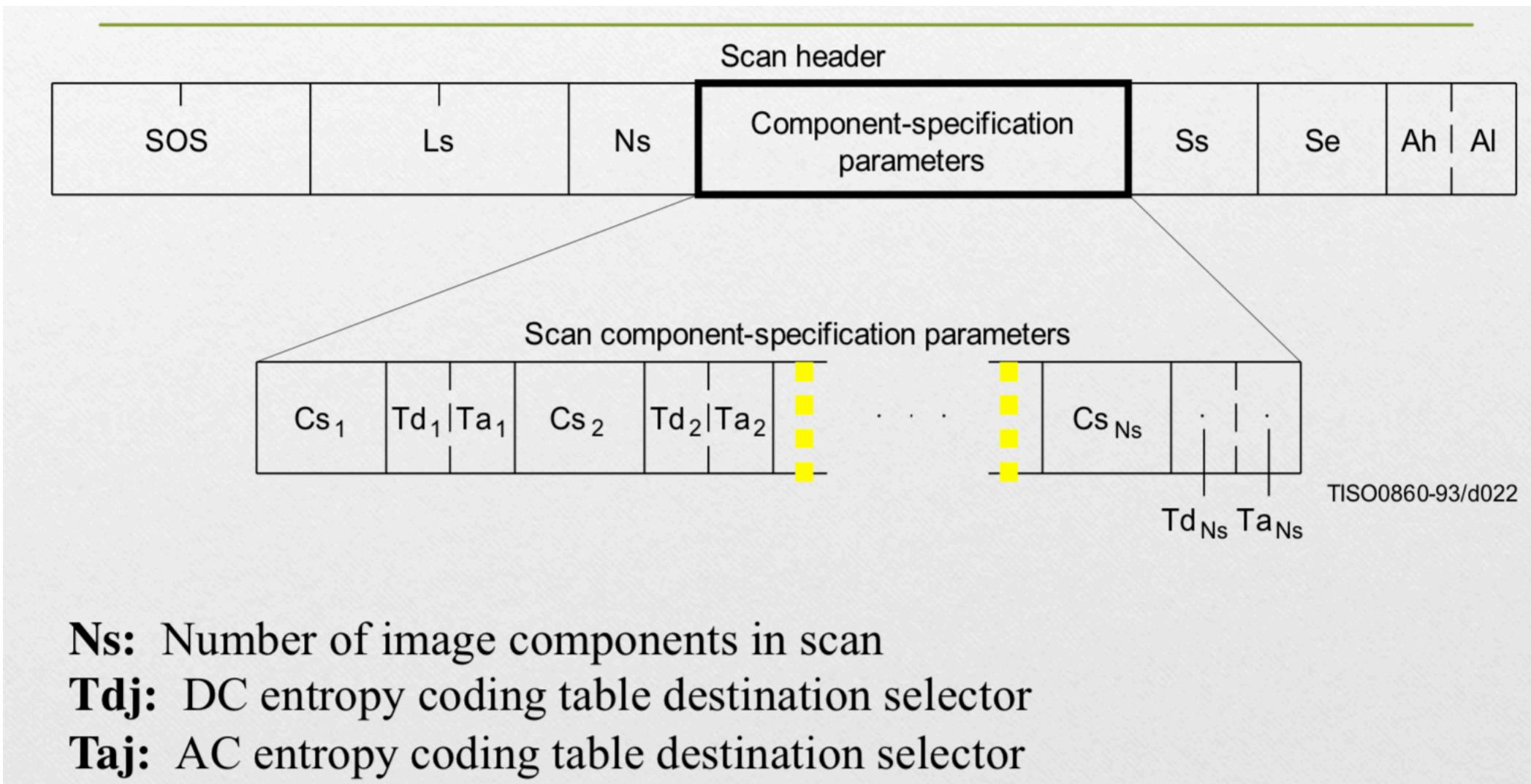
# SOS

Tag：兩個字節，固定為0xFFDA

包含具體字段：

1. 數據長度，2個字節，代表1~4的總長度。
2. 顏色分量數，1個字節，有三個可選（1:灰度值、3:YCrCb或YIQ  
4:CMYK）。
3. 顏色分量訊息
  - a. 顏色分量ID，1個字節。
  - b. DC/AC係數表號，1個字節，高四位為DC用的Hoffman tree ID、低四圍AC用的Hoffman tree ID。
4. 壓縮圖像數據
  - a. 開始，1個字節，固定值0x00。
  - b. 結束，1個字節，固定值0x3F。
  - c. 選擇，1個字節，基本JPEG總為90。

# SOS



# SOS

Parameter	Size (bits)	Values			
		Sequential DCT		Progressive DCT	Lossless
		Baseline	Extended		
Ls	16			$6 + 2 * N_s$	
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

EIO

Tag：兩個字節，固定為0xFFD9

# Decoding Process

1. Color Space Conversion

2. Subsampling

3. Partition

4. Encoding Flow Control

Discrete Cosine Transform(DCT)

Quantization

Entropy Encoding(Huffman)



Decoding