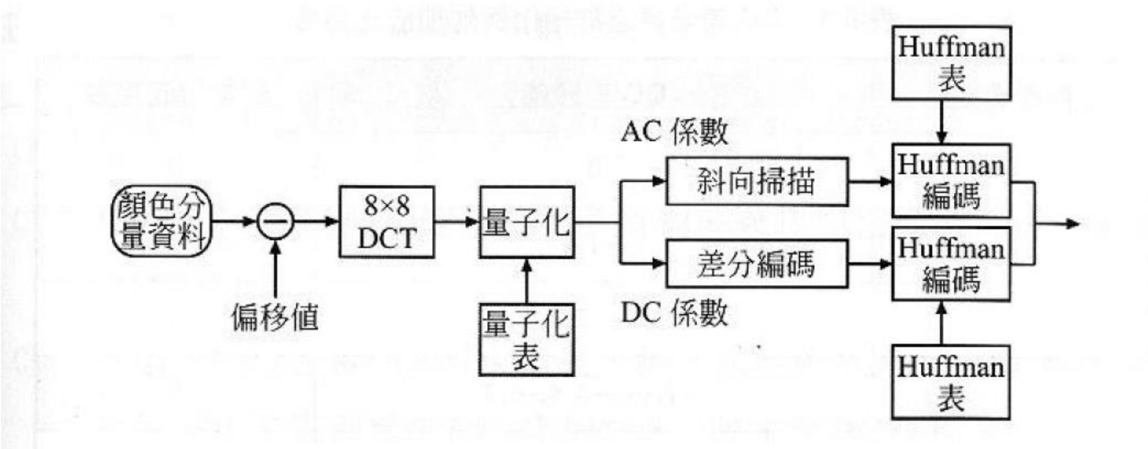


圖 9.2　JPEG 編碼器之基本組成

合適參考資料

[1] 酒井善則，吉田俊之原著，原島博監修，白執善編譯，“影像壓縮術”，全華印行, 2004. (建議最先讀，淺顯易懂)

[2] https://idea2ic.com/File_Formats/CCITT.pdf (官方標準)

[3] 戴顯權，“資料壓縮 Data Compression,” 旗標出版社, 2007. (同樣也是淺顯易懂的參考資料，有提到視訊壓縮)

(1) 顏色分量資料是將輸入的 R, G, B 先變成 Y, Cb, Cr

Y = 0.299R +0.587 +0.114G B

Cb = -0.169R - 0.331G + 0.5B

Cr = 0.5R -0.419G - 0.081B

(2) 接著，再根據 page 35, 圖 1.19 的方法

將 Cb, Cr 每兩個 rows 只選一個 row

每兩個 columns 只選一個 column

例如 Cb1[m, n] = Cb[2m, 2n]

(3) 接著，再對 Cb, Cr 值各加上 127.5，

讓 Cb, Cr 值皆為正的

Note:

第 (1)-(3) 步若要還原, 則將 127.5 加回來

再把刪去的 rows 和 columns 加回來

Cb[2m, 2n] = Cb1[m, n]

Cb[2m+1, 2n] =(Cb[2m, 2n] +Cb[2m+2, 2n])/2
Cb[m, 2n+1] = (Cb[m, 2n]+Cb[m, 2n+2])/2

若要從 YCbCr 還原至 RGB 的座標
可以算
[0.299, 0.114,0.587;
-0.169, -0.331, 0.5;
0.5, -0. 419, -0. 081] 的反矩陣

(4) 將 Y, Cb, Cr 分成多個 8x8 blocks

(5) 對每一個 8x8 blocks 做 DCT
假設 f(i, j) (i, j = 0, 1, 2, 3, 4, 5, 6, 7) 為其中一個 8x8 blocks
根據 Eqs. (5.99), (5.100) 來做 8x8 DCT

$$F(u,v)=\frac{2C(u)C(v)}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}f(i,j)\cdot\cos\frac{(2i+1)u\pi}{2N}\cos\frac{2j+1)v\pi}{2N} \tag{5.100}$$

$$f(i,j)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cdot\cos\frac{(2i+1)u\pi}{2N}\cos\frac{(2j+1)v\pi}{2N} \tag{5.101}$$

$$0\le i,j,u,v\le N-1，\quad C(n)=\begin{cases}1/\sqrt{2} & (n=0)\\ 1 & (n\neq 0)\end{cases}$$

值得注意的是，以上的式子可以用 matrix multiplication 來表示，若將

f(i, j) (i, j = 0, 1, 2, 3, 4, 5, 6, 7)

F(u, v) (u, v = 0, 1, 2, 3, 4, 5, 6, 7)

用 8x8 matrices **f**, **F** 來表示，則

$$\mathbf{F}=\mathbf{C}_1\mathbf{f}\mathbf{C}_2$$

其中

$$\mathbf{C}_1[u,i]=\sqrt{\frac{2}{N}}C(u)\cos\left(\frac{(2i+1)u}{2N}\pi\right) \qquad C(n)=\begin{cases}1/\sqrt{2} & n=0\\ 1 & n\neq 0\end{cases}$$

$$\mathbf{C}_2[j,v]=\sqrt{\frac{2}{N}}C(v)\cos\left(\frac{(2j+1)v}{2N}\pi\right)$$

我們可以驗證

$$\mathbf{C}_1 = \mathbf{C}_2^{\mathrm{T}}, \qquad \mathbf{C}_1^{-1} = \mathbf{C}_1^{\mathrm{T}} = \mathbf{C}_2$$

所以 $\mathbf{F} = \mathbf{C}_1 \mathbf{f} \mathbf{C}_2$ 可以改寫成

$$\mathbf{F} = \mathbf{C}_1 \mathbf{f} \mathbf{C}_1^{\mathrm{T}}$$

(Note): 這一步要做 inverse 可以用

$$\mathbf{f} = \mathbf{C}\mathbf{F}_1^{\mathrm{T}} \mathbf{C}_1$$

(Step 5 加速方法)

若原影像為 $\mathbf{A} = \begin{bmatrix} \mathbf{f}_{1,1} & \mathbf{f}_{1,2} & \mathbf{f}_{1,3} & \cdots & \mathbf{f}_{1,N} \\ \mathbf{f}_{2,1} & \mathbf{f}_{2,2} & \mathbf{f}_{2,3} & \cdots & \mathbf{f}_{2,N} \\ \mathbf{f}_{3,1} & \mathbf{f}_{3,2} & \mathbf{f}_{3,3} & \cdots & \mathbf{f}_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{f}_{M,1} & \mathbf{f}_{M,2} & \mathbf{f}_{M,3} & \cdots & \mathbf{f}_{M,N} \end{bmatrix}$ 每個 $\mathbf{f}_{m,n}$ 是一個 8×8 block

DCT output 為 $\mathbf{B} = \begin{bmatrix} \mathbf{F}_{1,1} & \mathbf{F}_{1,2} & \mathbf{F}_{1,3} & \cdots & \mathbf{F}_{1,N} \\ \mathbf{F}_{2,1} & \mathbf{F}_{2,2} & \mathbf{F}_{2,3} & \cdots & \mathbf{F}_{2,N} \\ \mathbf{F}_{3,1} & \mathbf{F}_{3,2} & \mathbf{F}_{3,3} & \cdots & \mathbf{F}_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{M,1} & \mathbf{F}_{M,2} & \mathbf{F}_{M,3} & \cdots & \mathbf{F}_{M,N} \end{bmatrix}$ $\mathbf{F}_{m,n}$ 是 $\mathbf{f}_{m,n}$ 的 8x8 DCT

可以採用

$$\begin{bmatrix} \mathbf{C}_1\mathbf{f}_{m,1} & \mathbf{C}_1\mathbf{f}_{m,2} & \mathbf{C}_1\mathbf{f}_{m,3} & \cdots & \mathbf{C}_1\mathbf{f}_{m,N} \end{bmatrix} = \mathbf{C}_1 \begin{bmatrix} \mathbf{f}_{m,1} & \mathbf{f}_{m,2} & \mathbf{f}_{m,3} & \cdots & \mathbf{f}_{m,N} \end{bmatrix}$$

m = 1, 2, …, M (將 A 每 8 個 rows 為一組，前面乘上 $\mathbf{C}_1$)

$$\begin{bmatrix} \mathbf{F}_{1,n} \\ \mathbf{F}_{2,n} \\ \mathbf{F}_{3,n} \\ \vdots \\ \mathbf{F}_{M,n} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1\mathbf{f}_{1,n} \\ \mathbf{C}_1\mathbf{f}_{2,n} \\ \mathbf{C}_1\mathbf{f}_{3,n} \\ \vdots \\ \mathbf{C}_1\mathbf{f}_{M,n} \end{bmatrix} \mathbf{C}_1^{\mathrm{T}}$$ (每 8 個 columns 為一組，後面乘上 $\mathbf{C}_1^{\mathrm{T}}$)

n = 1, 2, …, N

(6) 做 quantization，可使用 Eqs. (9.2), (9.3) 的 quantization matrices

$$F_q[m,n] = round\left(\frac{F[m,n]}{Q_Y[m,n]}\right)$$ for Y channel

$$F_q[m,n] = round\left(\frac{F[m,n]}{Q_C[m,n]}\right)$$ for Cb and Cr channels

其中 round 是指四捨五入，直接用 round 的指令即可，

$$Q_Y(u,v)=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \qquad (9.2)$$

$$Q_C(u,v)=\begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \qquad (9.3)$$

(Note): 這一步要做 inverse 可以用

$F[m,n]=F_q[m,n]Q_Y[m,n]$ for Y channel

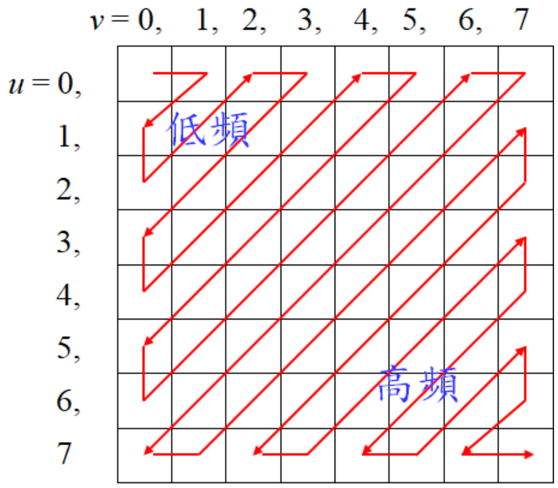
$F[m,n]=F_q[m,n]Q_C[m,n]$ for Cb and Cr channels

(7) For DC terms, calculate the DC difference.

$DC[i,j]-DC[i,j\text{-}1]$

$DC[i,j]$ means the DC value (the value of $F_q[0,0]=round\left(\dfrac{F[0,0]}{Q_{Y\ or\ C}[0,0]}\right)$) at the $[i,j]^{\text{th}}$ block.

(8) For AC terms, first, perform zigzag scanning



Please see page 6-14 and Eq. (6-15).

這一步建議用 table lookup, 告訴電腦 8x8 matrix 上的每一個點在 zigzag 之後排序為第幾個點，會比較省事

	1	5	6	14	15
2	4	7	13	16	
3	8	12	17		
9	11	18			
10	19				
20					

(9) Then, the zero-run-length method as in Eq. (6-16) is applied.

格式：(前面有幾個 0； 值)

EOB 代表 end of block, 後面將全部為 0

(10) DC AC 的編碼，要參考 CCITT Rec. T.81 這本書

DC Difference Table: Page 149

AC Table: Page 150-157

Luminance: Y; Chrominance: Cb, Cr

關於 DC Difference 編碼的部分，要注意的是，我們並不是直接對 DC difference 來編碼，而是將 DC difference 先做分類再編碼

當中的 category 是指這個 DC difference 被分為第幾群

Category 0 是 DC difference = 0

Category 1 是 DC difference = +-1

Category 2 是 DC difference = +-2, +-3

Category 3 是 DC difference = +-4, +-5, +-6, +-7

:

:

Category k 是 DC difference = +-2^(k-1), +-2^(k-1)+1, +-2^(k-1)+2, ..., +-2^k -1

編碼範例

若經過 quantization 之後，Y channel 相鄰兩個 8x8blocks 的 DC term 分別為 100, 95
則 DC difference 為-5,
-5 屬於 Category 3 當中的 負值，又是負值當中的第 2 個值
(-4, -5, -6, -7 當中的第 2 個值)
根據 Table K.3，Category 3 編碼為 100
接著，我們以 '1' 代表正，以'0'代表負
接著，由於 -5 是 4 個值當中的第二個值，所以用 '01' 來表示
(-4, -5, -6, -7 分別用 00, 01, 10, 11 來表示)
所以，-5 的編碼結果為
'100''0''01'

(11) 關於 AC Term 編碼的部分，
Y 的 AC term 要用 page 150-153 頁的 table
Cb, Cr 的 AC term 要用 page 154-157 頁的 table
要注意的是，我們並不是直接對 AC term 來編碼，
而是先轉在 zero-run-length 之後(如 Eq. (6-16)) 先做分類再編碼
Pages 150-157 當中的 Size 是指這個 DC difference 被分為第幾群
Size 0 是 AC value = 0

Size 1 是 AC value = +-1

Size 2 是 AC value = +-2, +-3

Size 3 是 AC value = +-4, +-5, +-6, +-7

:

:

Size k 是 AC value = +-2^(k-1), +-2^(k-1)+1, +-2^(k-1)+2, ..., +-2^k -1

編碼範例

根據 Eq. (6-15) 的例子，前面兩個 zero-run-length 值為 (0, -2), (3, 6) ,
首先，關於 (0, -2) 的部分
-2 屬於 Size 2 當中的 負值，又是負值當中的第 1 個值

(2, 3 當中的第 1 個值)
根據 Table K.5，zero-run = 0, Size = 2 編碼為 01
接著，我們以 '1' 代表正，以'0'代表負
接著，由於 -2 是 2 個值當中的第一個值，所以用 '0' 來表示
(-2,-3 分別用 0, 1 來表示)
所以，(0, -2) 的編碼結果為
'01''0''0'
其次，關於 (3, 6) 的部分
6 屬於 Size3 當中的 正值，又是正值當中的第 3 個值
(4，5，6，7 當中的第 3 個值)
根據 Table K.5，zero-run = 3, Size =3 編碼為 111111110101
接著，我們以 '1' 代表正，以'0'代表負
接著，由於 6 是 4 個值當中的第 3 個值，所以用 '10' 來表示
(4, 5, 6, 7 分別用 00, 01, 10, 11 來表示)
所以，(3, 6) 的編碼結果為
'111111110101''1''10'

(12) AC term 的補充
(i) End-of-block 就相當於 (0, 0)
(ii) Pages 150-157 當中，zero run 最多只有 F (15) 個
若超過 15 個 0，需要用 ZRL

例子：如果有 20 個 0 再接 1 (20，1)
可以變成 (F, 0), (4, 1)
(也就是 15 個 0，再加 1 個 0，再加 4 個 0，再加 1)
(F, 0)對應 11111111001
(4, 1) 對應 '111011''1' (1 代表正號)
所以編碼結果為
'11111111001''111011''1'

(13) 計算 bpp (bit per pixel)

$$bpp = \frac{total\ number\ of\ bits}{MN}$$

其中 M ×N 是影像的大小
(沒經過任何壓縮的影像，bpp = 24，因為有 RGB 三個 channels，每個 channel 的每一點需要 8 個 bits)

(14) 計算 PSNR

$$PSNR = 10\log_{10} \frac{255^2}{\frac{1}{3MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{c=r,g,b}\left(y[m,n,c]-x[m,n,c]\right)^2}$$

參考：

對於 Lena512c.bmp, bpp = 0.7228, PSNR = 31.92dB

對於 Baboon1.bmp, bpp = 1.5348, PSNR = 24.7228dB

對於 Baboon1.bmp, bpp = 0.7857, PSNR = 29.3088dB