

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;

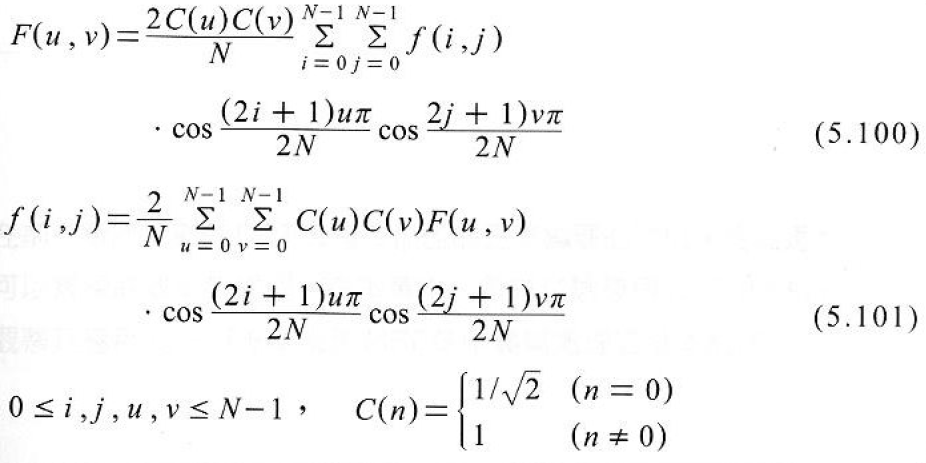
0.5, -0. 419, -0. 081] 的反矩陣

(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



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



其中



我們可以驗證



所以  可以改寫成



(Note): 這一步要做 inverse 可以用



(Step 5 加速方法)

若原影像為  每個 **f***m*,*n* 是一個 8×8 block

DCT output 為  **F***m*,*n* 是 **f***m*,*n* 的 8x8 DCT

可以採用



m = 1, 2, …, M (將 A 每8個 rows 為一組，前面乘上 **C1**)

 (每8個 columns 為一組，後面乘上 **C1T**)

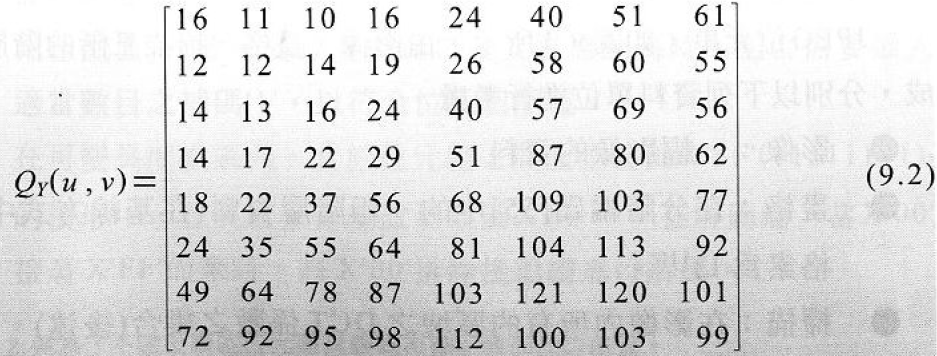
n = 1, 2, …, N

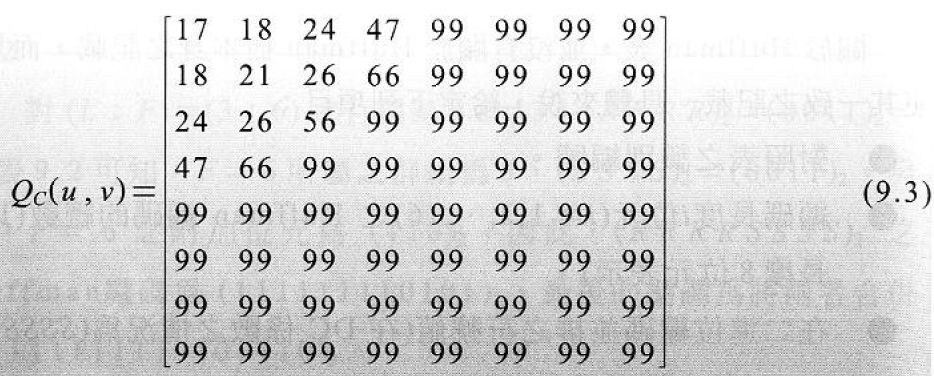
(6) 做 quantization，可使用 Eqs. (9.2), (9.3) 的 quantization matrices

 for Y channel

 for Cb and Cr channels

其中 round 是指四捨五入，直接用 round 的指令即可，





(Note): 這一步要做 inverse 可以用

 for Y channel

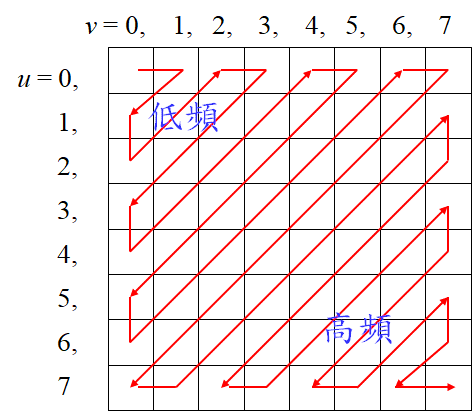
 for Cb and Cr channels

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

*DC*[*i*, *j*] – *DC*[*i*, *j*-1]

*DC*[*i*, *j*] means the DC value (the value of ) at the [*i*, *j*]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



(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)



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



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