

朱式幸福

2013年11月1日 星期五

Huffman Coding for DC & AC Coefficients of Grayscale JPEG Image

Joint Photographic Expert Group 是一種利用餘弦轉換 (Cosine Transform)的國際標準壓縮方法，詳細的方法可以參考 [1]www.w3.org/Graphics/JPEG/itu-t81.pdf

比較困難的部份並不是餘弦轉換，而是轉換後的AC值與DC值的壓縮編碼。今天我們利用Matlab來實作一遍吧！

◎影像切塊成8乘8(partition into 8x8 blocks)

聽起來這一步應該很簡單，唯一的問題是影像的尺寸如果不是8的倍數時，要如何處理？最簡單的方法就是在影像的最右邊及最下邊補灰度值為128的像素，讓寬度與高度都是8的倍數。

↓利用padarray將區塊補足至8X8

```
A =  
  
79 111 79 57 64 69 86  
70 68 90 83 68 98 80  
66 71 70 65 90 71 61  
85 57 68 90 79 53 43  
72 83 101 105 66 51 46  
88 89 97 89 66 54 56  
  
>> B = padarray(A,[2,1],128,'post')  
  
B =  
  
79 111 79 57 64 69 86 128  
70 68 90 83 68 98 80 128  
66 71 70 65 90 71 61 128  
85 57 68 90 79 53 43 128  
72 83 101 105 66 51 46 128  
88 89 97 89 66 54 56 128  
128 128 128 128 128 128 128 128  
128 128 128 128 128 128 128 128
```

◎平移128 (shift by 128)

將區塊的每一個灰度值減去128，由於灰度影像的class是uint8，像素值介於0與255之間，在減去128之前，需要影像格式轉為double。

↓轉換格式後再減去128



標籤

- 319鄉 (3)
- 工程數學 (29)
- 公費留考 (1)
- 心得 (3)
- 指考 (18)
- 研討會 (45)
- 海外遊 (30)
- 特招 (15)
- 高中數學 (67)
- 高普考 (38)
- 高職數學 (81)
- 國小數學 (2)
- 國中數學 (70)
- 國內遊 (51)
- 基測 (24)
- 教檢 (2)
- 單車 (37)
- 統計 (3)

```
>> C = double(B)-128

C =

-49 -17 -49 -71 -64 -59 -42 0
-58 -60 -38 -45 -60 -30 -48 0
-62 -57 -58 -63 -38 -57 -67 0
-43 -71 -60 -38 -49 -75 -85 0
-56 -45 -27 -23 -62 -77 -82 0
-40 -39 -31 -39 -62 -74 -72 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

◎餘弦轉換(cosine transform)

由於Matlab 已有餘弦轉換的函式，直接套用就可以了

↓利用函式dct2作餘弦轉換

```
>> D = dct2(C)

D =

-280.2500 -25.0373 46.5936 -73.3596 49.7500 -36.8107 28.8668 -14.8188
-116.7935 -31.7484 27.6545 -11.3168 1.5102 -20.9354 7.7955 -4.1314
100.6004 -5.8794 5.3943 38.8620 -53.1055 6.7996 -18.4268 6.1113
-27.2858 21.1516 22.2765 -31.2395 -10.0935 -1.6924 -4.4046 -7.0030
-5.7500 13.7562 6.0188 -9.8431 8.7500 -0.8000 -8.9874 -3.1807
15.5493 3.4246 8.7387 20.4926 6.5468 -7.3138 -14.6543 8.9978
-38.6935 4.8812 18.0732 -7.4203 4.4081 -19.5806 -8.3943 7.7015
23.0701 0.5536 -7.1645 -0.4108 -14.0260 -9.2384 -10.3621 3.8017
```

◎量化餘弦轉換後的值(quantization)

餘弦轉換後的值為實數，不利壓縮。所以將其量化，其實就是除以一個整數，並四捨五入成為一個整數。在JPG標準文件中，量化矩陣Q(其實就是除數)定義為

↓量化矩陣Q

```
Q =

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 36 55 64 81 194 113 92
49 64 78 87 103 121 120 101
72 92 95 98 112 100 103 99
```

↓餘弦轉換後的8X8區塊除以矩陣Q，再四捨五入

```
>> T = round(D./Q)

T =

-18 -2 5 -5 2 -1 1 0
-10 -3 2 -1 0 0 0 0
7 0 0 2 -1 0 0 0
-2 1 1 -1 0 0 0 0
0 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
-1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

◎儲存量化區塊

量化區塊的值分成兩類，一個叫DC值，也就是左上角第一個值，其它的數值稱之為AC值。

DC值只有一個，所以將每個區塊的DC值依掃描線順序排列儲存。每個區塊有63個AC值，以Z字形(zigzag)順序儲存。

- 統測 (63)
- 微分方程 (3)
- 微積分 (20)
- 會考 (7)
- 路跑 (11)
- 運動績優 (7)
- 電腦管理 (17)
- 臺澎金馬 (6)
- 論文徵稿 (2)
- 學力鑑定 (19)
- 學測 (9)
- 應用數學 (1)
- 警專 (14)
- DIY (41)
- GeoGebra (4)
- GIMP (1)
- LaTeX (4)
- matlab (17)
- octave (23)
- python (8)
- R (1)
- Scratch 程式設計 (7)

熱門文章

103年國中會考數學詳解

106年大學學測數學詳解

104年國中會考數學詳解

105年國中教育會考數學詳解

107年國中教育會考數學詳解

網誌存檔

- 2019 (6)
- 2018 (123)
- 2017 (49)
- 2016 (89)
- 2015 (29)
- 2014 (65)
- ▼ 2013 (83)
 - 十二月 (4)
 - ▼ 十一月 (7)
 - 102年國中基測數學詳解
 - 木柵貓空&清龍宮的艱苦騎行
 - 國中數學：難搞的因式分解
 - 國中數學：求相離兩圓內公切線的夾角
 - 更新 WOctave

Z字形順序就是將區塊順時鐘轉45度，再以平常的掃描線方式循環排列。

↓zigzag在8X8區塊的順序

```

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

```

在matlab中，我們先以人工的方式建立8x8 zigzag序列的一維座標，再套入二維區塊中。以上圖為例，序號1在一維座標是9，序號2的一維座標是2，序號3的一維座標是3，序號4的一維座標是10，依此類推。matlab的二維座標轉一維座標是先算y軸，再算x軸，也就是 $A(y,x) = A((x-1)*8+y)$ 。EX. $A(1,2)=A((2-1)*8+1)=A(9)$ 。

↓建立ZIGZAG的一維座標

```

>> zz=[...
    9 2 3 10 17 25 18 11 4 5 12 19 26 ...
    33 41 34 27 20 13 6 7 14 21 28 35 ...
    42 49 57 50 43 36 29 22 15 8 16 23 ...
    30 37 44 51 58 59 52 45 38 31 24 32 ...
    39 46 53 60 61 54 47 40 48 55 62 63 56 64];

```

↓將量化後的矩陣T以zigzag的順序列出

```

>> T(zz)

ans =

Columns 1 through 17

    -2    -10     7     -3     5     -5     2     0     -2     0     1     0    -1

Columns 18 through 34

     1     1     1    -1     0     0    -1    -1     0     1     0     0     0

Columns 35 through 51

     0     0     0     0     0     0     0     0     0     0     0     0     0

Columns 52 through 63

     0     0     0     0     0     0     0     0     0     0     0     0

```

◎DC值編碼

由於影像被切成許多8X8區塊，每一個區塊有一個DC值，這個值在餘弦轉換佔有重要地位，解壓縮影像的畫值與其有密切關係。

將每個DC值排成一維陣列，假設是DC1, DC2, ... DCn。第一個DC值不動，計算第2個DC值與第1個DC值的差，再計算第3個DC值與第2個DC值的差，...，依此類推。求得 DC1, (DC2-DC1), (DC3-DC2), ..., (DCn-DCn-1)。

由於JPG是個國際標準的壓縮方式，對DC值的Huffman Code已有現成表格可供使用。對於每一個DC差值，看它的範圍落在哪個Category，找出對應的Code word來使用。

另一個小細節：第一個DC值(不是差值)，仍然使用與差值相同的方法來編碼！

↓DC差值的Huffman編碼(取自[1]第149頁)

至1.3.2 版本

國中數學：圓
內角與圓外
角

Huffman Coding
for DC & AC
Coefficients
of Grayscale

- ▶ 十月 (8)
- ▶ 九月 (5)
- ▶ 八月 (8)
- ▶ 七月 (8)
- ▶ 六月 (6)
- ▶ 五月 (6)
- ▶ 四月 (9)
- ▶ 三月 (5)
- ▶ 二月 (9)
- ▶ 一月 (8)

▶ 2012 (60)

▶ 2011 (2)

總瀏覽量



1,306,918

關於我自己

C.-H. Chu

檢視我的完整簡介

Table K.3 – Table for luminance DC coefficient differences

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

為了容易理解，把DC的差值加入上表中，可得下表

Category	DC差值範圍	Huffman Code	編碼長度
0	0	00	2
1	-1, 1	010	3
2	-3, -2, 2, 3	011	3
3	-7~-4, 4~7	100	3
4	-15~-8, 8~15	101	3
5	-31~-16, 16~31	110	3
6	-63~-32, 32~63	1110	4
7	-127~-64, 64~127	11110	5
8	-255~-128, 128~255	111110	6
9	-511~-256, 256~511	1111110	7
10	-1023~-512, 512~1023	11111110	8
11	-2047~-1023, 1023~2047	111111110	9

因此DC差值(假設為DIFF)的編碼流程如下：

如果 DIFF=0 則輸出為 00

如果 DIFF>0 則

依上表找出對應的Huffman Code(假設為HCode)，並求出DIFF的二進位碼(假設為BDIFF)，輸出 [HCode BDIFF]；

如果DIFF<0 則

依上表找出對應的Huffman Code(假設為HCode)，並求出DIFF的**絕對值**的二進位碼(假設為BDIFF)，輸出 [HCode ~BDIFF] (~代表0變1, 1變0, 也就是**補數**)。

舉例：

- DIFF = 2 → Category=2→HCode = 011, BDIFF = 10, 所以輸出為[011 10] = 01110
- DIFF = 12 →Category=4→HCODE = 101, BDIFF = 1100, 所以輸出為[101 1100] = 1011100
- DIFF = -4 →Category=3→HCODE = 100, BDIFF = 100, ~BDIFF = 011, 所以輸出為[100 001]= 100011
- DIFF = -100→Category=7→HCODE = 11110, BDIFF = 1100100, ~BDIFF = 0011011, 所以輸出為[100 001]= 111100011011

↓matlab 的程式碼如下

```

1 function DCCODE = getDC(dc)
2     Hcode = {'00', '010', '011', '100', '101', '110', '1110', ...
3             '11110', '111110', '1111110', '11111110'};
4
5     if dc==0
6         mycode = {'00'};
7     elseif dc>0
8         index = floor(log2(dc));
9         mycode = strcat(Hcode(index+2), dec2bin(dc));
10    else
11        index = floor(log2(abs(dc)));
12        bdc = dec2bin(abs(dc));
13        for x=1:length(bdc)
14            if(bdc(x)=='0') bdc(x)='1';
15            else bdc(x)='0';
16        end
17    end
18    mycode = strcat(Hcode(index+2), bdc);
19 end
20 DCCODE = mycode{1};

```

↓執行看看...結果與上述相同

```

>> getDC(2)

ans =

01110

>> getDC(12)

ans =

1011100

>> getDC(-4)

ans =

100011

>> getDC(-100)

ans =

111100011011

```

◎AC值編號

在一個8X8的區塊中，DC值編碼與另外63個AC值編碼略有不同。利用AC值本身並不是AC值的差值來編號，另外，由於AC值有許多零，所以將零的個數納入編碼的因素，採用 run-length及Huffman Code的方式來編碼。

因此決定編碼的索引值就是**AC值與有幾個零**！AC值的編碼與DC差值編碼使用的方法完全一樣，在JPG標準中，以Size來代表AC值的範圍，其範圍如下表：

↓AC值的Size 與DC值的Category意義相同

Size	AC值範圍
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

$$10=A \quad -1023 \sim -512, 512 \sim 1023$$

上表與DC差值的對應表幾乎一樣，一個是沒有Size=0，一個是只考慮到第10類，沒有第11類。這是由於AC值為零使用run-length的概念來取代，而且AC值的範圍也沒有DC值這麼廣。

另一個索引值就是零的個數，以Run來代表。在8X8區塊的灰度影像中，JPG標準最多只針對15個連續的零來編碼，因此Size的範圍就是0至15。如果有連續16個零就以ZRL來編碼。

因此(Run/Size)的範圍就是從(0/1)至(15/10) = (F/A)，對應表項目很多，只列出一小段。

↓AC值(侷部)的Huffman編碼(取自[1]第150頁)，共四頁

Table K.5 – Table for luminance AC coefficients (sheet 1 of 4)

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	111111110000010
0/A	16	111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	111111110000100
1/7	16	111111110000101
1/8	16	111111110000110
1/9	16	111111110000111
1/A	16	111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111

在這個很長很長的表格中，有兩個特殊的編碼，一個是0/0，另一個是F/0。0/0代表區塊結束(End Of Block EOB)，F/0代表前面有15個零，而且自己也是零(ZRL, 共有16個零)。其它的編碼都沒有Size=0的索引值。

現在開始來寫AC值的編碼流程：

輸入值：AC(1:63) 是一個具有63個AC值的一維陣列，而且已按Zigzag順序排列。

首先找出AC陣列中最後一個不為零的序號值K，即AC(K)不為零，AC(K+1)至AC(63)都是零。

接著依序從i=1至K處理AC陣列的值

B=AC(i)

如果B大於零

依上表求出其size值，設為size(B)；

求出前面有幾個連續的零，設為zero(B)；

由 zero(B)/size(B)=run/size 查表找出對應的Huffman Code, 設為HC

輸出 HC及B的二進位碼, 並將零的個數歸零；

如果B小於零

依上表求出其size值，設為size(B)；

求出前面有幾個連續的零，設為zero(B)；

由 zero(B)/size(B)=run/size 查表找出對應的Huffman Code, 設為HC

計算絕對值B的二進位碼並計算此二進位的補數，輸出 HC及補數，並將零的個數歸零；

如果B為零

如果前面零的個數小於15，將零的個數加1

如果前面零的個數等於15，輸出ZRL的代碼，並將零的個數歸零。

依序完成以上K個AC值，最後輸出EOB的代碼

matlab的程式如下：

But 程式碼太長了，分段顯示

↓程式碼的上半段

```

1  function ACCODE = getAC(AC)
2  EOB = '1010';
3  ZRL = '1111111001';
4  MYCODE=[];
5  NOZERO=0;
6  Hcode={
7      '00' '01' '100' '1011' '11010' '1111000' '11111000' '1111110110' '1111111
8      '1100' '11011' '1111001' '111110110' '1111110110' '11111110000100' '11
9      '11100' '11111001' '1111110111' '11111110100' '111111110001001' '111111
10     '111010' '111110111' '11111110101' '111111110001111' '111111
11     '111011' '1111111000' '111111110010110' '111111110010111' '111111
12     '1111010' '11111110111' '111111110011110' '111111110011111' '111111
13     '1111011' '111111110110' '111111110100110' '111111110100111' '111111
14     '11111010' '111111110111' '111111110101110' '111111110101111' '1111
15     '111111000' '11111111000000' '111111110110110' '111111110110111' '111
16     '111111001' '11111111011110' '111111110111111' '111111111000000' '111
17     '111111010' '111111111000111' '111111111001000' '111111111001001' '111
18     '1111111001' '111111111010000' '111111111010001' '111111111010010' '11
19     '1111111010' '111111111011001' '111111111011010' '111111111011011' '11
20     '11111111000' '111111111100010' '111111111100011' '111111111100100' '1
21     '111111111101011' '111111111101100' '111111111101101' '11111111110111
22     '111111111110101' '111111111110110' '111111111110111' '11111111111100
23 };
24 K = find(AC,1,'last');
25 if isempty(K)
26     ACCODE = EOB;
27     return;
28 end
29

```

上圖中的程式中，Hcode是一個二維的cell，儲存(run/size)對應的Huffman 的編號。利用find 尋找最後一個非零的AC值，並將其索引值儲存至K。如果K是空矩陣，代表63個值都是零，就輸出EOB，並結束程式。

↓程式碼的下半段

```

29
30 for i=1:K
31     B = AC(i);
32     if B>0
33         index = floor(log2(B));
34         MYCODE = strcat(MYCODE,Hcode{NOZERO+1,index+1},dec2bin(B));
35         NOZERO=0;
36     elseif B<0
37         index = floor(log2(abs(B)));
38         bdc = dec2bin(abs(B));
39         for x=1:length(bdc)
40             if(bdc(x)=='0') bdc(x)='1';
41             else bdc(x)='0';
42         end
43     end
44     MYCODE = strcat(MYCODE,Hcode{NOZERO+1,index+1},bdc);
45     NOZERO=0;
46 else
47     if NOZERO==15
48         MYCODE = strcat(MYCODE,ZRL);
49         NOZERO=0;
50     else
51         NOZERO = NOZERO+1;
52     end
53 end
54
55 end
56
57 ACCODE = strcat(MYCODE,EOB);
58
59 end

```

現在來試試看吧！

↓先SHOW一下AC值的內容

```
>> reshape(ZZ, 7, 9)

ans =

     7     -2     0     0     0     0     0     0     0
     2      1     0     0     0     0     0     0     0
     1      0     0     -1     0     0     0     0     0
    -3      2      1     -1     0     0     0     0     0
    -4      2      1     0     0     0     0     0     0
    -1      1     -1     0     0     0     0     0     0
     1     -1     0     0     0     0     0     0     0
```

63個數字若排成一行，不易閱讀，利用reshape，用一個9x7的陣列方式顯示。由於matlab儲存二維陣列的順序是先Y軸再X軸，所以在AC一維陣列的順序是7 2 1 -3 -4 -1 1 -2 1 0....最後一個非零的值是-1，位置是25 (AC(25)=-1)。

↓執行結果

```
>> getAC(ZZ)

ans =

10011101100010100100011000001010100111011100110001000111010100100011101000001010
```

解釋一下：

- AC(1)=7 → index=2, NOZERO=0 → Hcode{0+1,2+1}=100, dec2bin(7)=111 → 輸出100111
- AC(2)=2 → index=1, NOZERO=0 → Hcode{1,2}=01, dec2bin(2)=10 → 輸出0110
- AC(4)=-3 → index=1, NOZERO=0 → Hcode{1,2}=01, dec2bin(abs(-3))=11 補數00 → 輸出0100
- AC(11)=2 → index=1, NOZERO=1 → Hcode{2,2}=11011,
- dec2bin(2)=10 → 輸出1101110
- AC(18)=1 → index=0, NOZERO=3 → Hcode{4,1}=111010,
- dec2bin(1)=1 → 輸出1110101
- AC(24)=-1 → index=0, NOZERO=3 → Hcode{4,1}=111010, dec2bin(abs(-1))=1 補數0 → 輸出1110100

↓再雞婆一點，每一個AC(K)及其編碼如下表

K	編碼
1	100111
2	0110
3	001
4	0100
5	100011
6	000
7	001
8	0101
9	001
10	
11	1101110
12	0110
13	001
14	000
15	
16	
17	
18	1110101
19	001
20	000
21	
22	
23	
24	1110100
25	000
26	1010

上表中，空白部份 (K=10, 15-17, 21-23)代表AC值為零。

大功告成, 累~~~

對了！JPG有幾種變形，以上是簡單的一種，稱之為BASELINE。還有彩色的...漸進壓縮...無損壓縮....更累~~~

--- END ---

張貼者： C.-H. Chu 於 下午4:10

標籤： matlab

4 則留言:



Unknown 2016年7月19日 下午5:05

作者已經移除這則留言。

[回覆](#)



匿名 2016年7月19日 下午5:19

大大您好:

請問在DC值編碼的時候

為什麼不能對負數做一個編碼

跑出的error是

Improper index matrix reference.

Error in getdc (line 19)

mycode=streat(Hcode(index+2).bdc);

不知道是什麼問題

能請大大做一個解釋嗎!??? 謝謝你

[回覆](#)



黃柏瑋 2016年10月13日 下午10:30

哦哦哦哦有這邊文章真是太感謝了~

[回覆](#)



louisL 2018年12月2日 上午11:06

非常感謝，目前正在大學修相關課程，這篇非常有幫助!!

[回覆](#)

輸入您的留言...



發表留言的身分：

Te-Han Kung (Goc

[登出](#)

[發佈](#)

[預覽](#)

☐ [通知我](#)

[較新的文章](#)

[首頁](#)

[較舊的文章](#)

訂閱：[張貼留言 \(Atom\)](#)

技術提供：[Blogger](#).