Digital image process HW#1

電機 07 黃國祐 0310781

1.

BMP format:

Bmp 檔案的儲存方法為無失真的儲存，因此檔案相對來說可能較大，而其檔案結構分為。

Bitmap file header 為前 14 bytes, 透過判讀前 2 個 bytes 可以判斷這個黨是否為 bmp 檔，而在這個之後的 4 個 bytes 則記錄圖片大小，最後 4 個 bytes 則記錄像素資料的起始位置。

而其後則為 BITMAPINFOHEADER，其大小依據不同類型的圖片，會有多種可能，但其前 40bytes 依舊保留為固定格式

透過判讀 BITMAPINFOHEADER 中的色深和壓縮方法和圖片寬度和高度和調色盤顏色數，可以進一步對下面的資料作分析。

| | | | | |
|---|---|---|---|---|
| Bitmap Info Header Palette Bitmap Array | 0022h | Bits Per Pixel | 2 | 每個像素的位元數<br>1：單色點陣圖（使用 2 色調色盤）<br>4：4 位元點陣圖（使用 16 色調色盤）<br>8：8 位元點陣圖（使用 256 色調色盤）<br>16：16 位元高彩點陣圖（不一定使用調色盤）<br>24：24 位元全彩點陣圖（不使用調色盤）<br>32：32 位元全彩點陣圖（不一定使用調色盤） |
| | 0026h | Compression | 4 | 壓縮方式：<br>0：未壓縮<br>1：RLE 8-bit/pixel<br>2：RLE 4-bit/pixel<br>3：Bitfields |
| | 002Ah | Bitmap Data Size | 4 | 點陣圖資料的大小（單位：byte）。 |
| | 002Eh | H-Resolution | 4 | 水平解析度（單位：像素/公尺） |
| | 0032h | V-Resolution | 4 | 垂直解析度（單位：像素/公尺） |
| | 0036h | Used Colors | 4 | 點陣圖使用的調色盤顏色數 |
| | - | Important Colors | 4 | 重要的顏色數 |

JPG format:

Bmp 檔案的儲存方法為失真壓縮，因此在相同大小的圖片下，jpg 的檔案相對較小，因此較適合用於網路傳輸，但因為其為失真壓縮，因此圖片再觀看上會出現失真。

其壓縮的流程為

(1)RGB -> YCbCr　(2)down sampling(因為人類對色彩的感應較差，而對亮暗的感應較強，因此會對色彩做取樣)　(3)DCT，透過 DCT 轉換後會將圖片轉換至頻率空間　(4)quantization，透過 quantization 進一步壓縮資料量，並同時減少高頻的成分(因為人類對高頻成分的分辨能力較差)(5)Z 字形編碼　(6) Huffman coding，透過 coding 來進行無失真壓縮

而解壓縮的流程則為相反。

在 jpg 的檔案中，

其內容(標記代碼)從頭開始約為(以一塊而論):

0xFFD8，圖像開頭(用來識別檔案是否為 jpg 檔)

0xFFE0，APP0 塊的起始點

APPn，其他的資訊

DQT，定義 Quantization 的表格

SOF0，圖像的外觀和長度的

DHT，定義 Huffman Table

DRI，定義差分編碼的間格

SOS，像素資料的起點

EOI，結束

2.

在本次作業中，我使用的 resolution 分別為 64、32、16 levels。

實現的方法主要為透過整數的 shift，利用 c++在 shift 的時候會補零的特性，自動扣除掉尾巴的值，藉此達到改變取樣精度的效果。

在輸出的圖形中，我們可以透過觀察明顯發現，在大範圍顏色接近的地方例如第二張圖的背景和第一張圖中色卡中的區塊在 resolution 降低時，較能夠感覺出來明顯的差異，而其他原先顏色變動較大的區塊則較感覺不出差異。 正如我在上面 JPG 圖檔說到的，這個是因為人類對高頻成分的分辨能力較差，因此才會有這樣的結果。

3. scaling

在這個作業中，我們使用到的是 bilinear 的 scaling，相較利用取最近的值來進行縮放，這樣做的優點在於說他在邊緣的地方不會出現鋸齒狀，因為 bilinear 本身也具有一些 smoothing 的效果，若是從頻域上來看 up/down sampling 這個動作，我們可以知道他會在圖形上產生高頻的成分。而 Bilinear Interpolation 的 smoothing 的效果會將這個高頻的成分減少，藉此使圖形看起來比較正常

在 bilinear 中我們是利用線性的方式對兩個像素的中間的值做逼近，我們知道這是一個 curve fitting 的問題，因此在不考慮 overfitting 的狀況之下，透過增加方程式的複雜度，也就是次方，我們可以對新的像素點得到比較準確的數值。在 bicubic interpolation 中，我們利用三次方程式對這個曲線做逼近，藉由假設曲面為連續的，我們可以知道在相鄰的四個點中，分別取前三和後三，在中間的兩個點上的一次微分和二次微分的數值應該是相同的，藉由這個運算，我們可以求出線性的 cubic interpolation，而在另外一個維度再做一次即可得 bicubic interpolation.