

## I. Image input/flip/output(BMP Format)

BMP 取自點陣圖 Bitmap 的縮寫，也稱為 DIB（與裝置無關的點陣圖），是一種獨立於監視器的點陣圖數位影像檔案格式。常見於微軟視窗和 OS/2 作業系統，Windows GDI API 內部使用的 DIB 資料結構與 BMP 檔案格式幾乎相同。

圖像通常儲存的顏色深度有 2（1 位）、16（4 位元）、256（8 位元）、65536（16 位元）和約 1677 萬（24 位元）種顏色（其中位是表示每點所用的資料位）。8 位元圖像可以是索引彩色圖像外，也可以是灰階圖像。表示透明的 alpha 通道也可以儲存在一個類似於灰階圖像的獨立檔案中。帶有整合的 alpha 通道的 32 位元版本已經隨著 Windows XP 出現，它在 Windows 系統的登入介面和系統主題中都有使用。

n 位  $2^n$  種顏色的包含調色盤的點陣圖近似位元組數可以近似為下列公式

$$\text{BMP 檔案大小} \approx 54 + 4 \cdot 2^n + \frac{\text{width} \cdot \text{height} \cdot n}{8}$$

(取自維基百科)

**BMP的標頭格式**

1. 格式: BM (0x 42 4D)
2. 檔案大小: 0x 0436 = 1078 bytes
- 3&4. 保留 全部為零
5. 資料偏移: 0x 36 = 54 bytes

資料來源: <http://www.jinnistag.com/2010/08/bmp-format-graphic-illustration.html>

朝陽科技大學 資訊與通訊系 廖俊龍

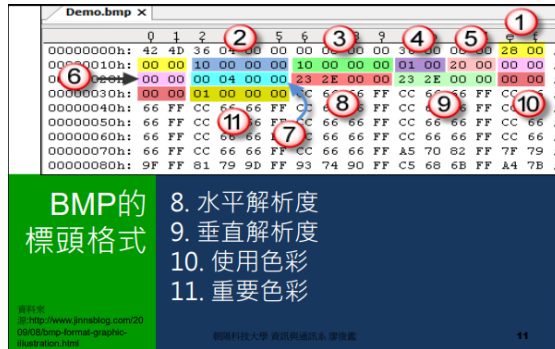
1. 格式，2 bytes，在 BMP 檔為 (0x) 42 4D
2. 檔案大小，4 個 bytes
3. 保留字元，2 個 bytes
4. 保留字元，2 個 bytes
5. Offset，4 個 bytes，從頭到位圖數據的偏移

**BMP的標頭格式**

1. 標頭大小: 0x 28 = 40 bytes
2. 影像寬: 0x 10 = 16 pixels
3. 影像高: 0x 10 = 16 pixels
4. 影像平面數: 0x 01 = 1
5. 位元深度: 0x 20 = 32 bits
6. 壓縮方式: 0x 00 = 未壓縮
7. 影像大小: 0x 0400 = 1024 bytes

資料來源: <http://www.jinnistag.com/2010/08/bmp-format-graphic-illustration.html>

朝陽科技大學 資訊與通訊系 廖俊龍



1. 標頭大小，4 個 bytes
2. 影像寬度，4 個 bytes
3. 影像高度，4 個 bytes
4. 影像平面數，2 個 bytes
5. 像素數，2 個 bytes，值有 1、2、4、8、16、24、32
6. 壓縮類型，4 個 bytes，最常用的是 0，表示不壓縮
7. 位圖數據大小，4 個 bytes
8. 水平解析度，4 個 bytes
9. 垂直解析度，4 個 bytes
10. 調色盤的顏色索引數，4 個 bytes，0 代表所有
11. 重要色彩，4 個 bytes，0 說明都重要

調色盤是可選的，不過這裡的 8 位色圖有調色盤。那麼接下來的數據就是調色盤了。調色盤就是一個顏色的索引，這裡是 8 位色圖，一共有 256 中顏色，由於每個顏色都有 RGB 三原色，也就是要 3 個字節表示，這樣的話 256 個顏色就不能表示所有的顏色，所以需要一個索引，用一個字節的索引指向 4 個字節表示的顏色（RGB 加上 Alpha 值）。如果把這 4 個字節表示為一個 Color 類型，那麼調色盤就是 Color 的陣列。由於 Color 類型也是一個陣列，調色盤就像一個二維陣列 palette[N][4]，其中 N 是顏色的數量，這裡就是 256。因此，這個例子中的調色盤的大小就是 256x4=1024 字節，在調色盤之前，有 14 字節的 bmp 文件頭，40 字節的位圖信息頭，加上 1024 字節的調色盤，一共 1078 字節，也就是說真正的圖像數據前面有 1078 字節。

最後就是位圖數據。由於是 8 位色圖，所以每個像素用 1 個 bytes 表示。當 biHeight 為正數，說明圖像倒立，從左下到右上，以行為主序排列。  
(取自朝陽科技大學 影像處理實務)

在 flip 的操作中，我們把數據左右對調即可。

## II. Resolution

在此次調整 resolution 的過程中，我們維持 8 位色圖的 RGB 為一組 1 個 bytes(8 bits)的長度，區別就是我們使用 Quantization 的方法，而有效的數據量減少為  $2^6$ 、 $2^4$ 、 $2^2$ 。

## III. Scaling

雙線性插值 (Bilinear interpolation) 是在二維空間內進行插值的一種方法。其主要思想是對圖像的兩個方向 (通常為 x 和 y 方向) 分別進行一維的線性插值，再結合這兩個方向的結果得到最終的插值結果。

具體的操作步驟如下：

1. 水平插值：首先在水平方向上，對目標點的左右兩個已知點進行線性插值。
2. 垂直插值：然後在垂直方向上，對目標點的上下兩個已知點進行線性插值。
3. 組合：最後，基於上述兩次插值的結果，再次進行線性插值，得到最終的插值結果。

以圖像放大為例，當我們想要放大一幅圖像時，原有的像素位置是已知的，但放大後的圖像中間會出現很多新的、未知的像素位置。雙線性插值就是通過周圍四個最近的已知像素點的值，來估計這些未知像素點的值。

數學上，假設我們有四個點： $f(x_1, y_1)$ 、 $f(x_1, y_2)$ 、 $f(x_2, y_1)$ 、 $f(x_2, y_2)$  的插值結果，那麼雙線性插值可以表示為：

$$f(x, y) = f(x_1, y_1)(1 - a)(1 - b) + f(x_2, y_1)a(1 - b) + f(x_1, y_2)(1 - a)b + f(x_2, y_2)ab, \quad \text{其中 } a = x - x_1, b = y - y_1$$

這個公式的意思是，新點的值是周圍四個點值的加權平均，而這些權重取決於新點與這四個點的相對距離。