**Run-Length Based Image Compression 練習**

附件中為三張利用將晶片高度以色彩視覺化後的圖片。 請設計一個基於 Run-Length 的壓縮法方，對圖檔作無失真壓縮後儲存成新檔案。 部落格上應敘述你的壓縮方法，提供壓縮檔之格式，並計算三張圖的平均壓縮率 (compression ratio)。

語言:c++

開發環境: Windows 10 + Visual Studio 2019 + OpenCV 4.5.4

**壓縮檔之格式**

儲存成dat檔，內有原圖長寬、通道順序、像素值與連續次數，儲存像素值的通道順序為R>G>B。

內容:

寬\*高

R G B

像素值 連續次數

像素值 連續次數

像素值 連續次數

...

...

解壓縮時，先讀出原圖片長寬，就能照通道順序與連續次數填入像素值，還原出原圖。

**程式解說**

**輸入檔案方式:在main裡填入檔案路徑**

string inputfilename[3] = { "img1.bmp" ,"img2.bmp" ,"img3.bmp" };

**輸出為compressed\_+原檔名稱 的dat檔**

**a.壓縮方式**

將圖片分成3個通道(RGB)，再把通道拉平成一維比較好做。

對每一個通道做壓縮，用flattenedRedChannel.at<uchar>看每pixel的值，如果一樣cnt就+1，直到不一樣，再把資料存進vector裡。

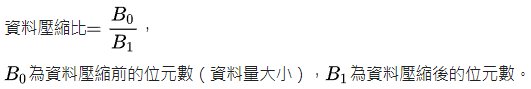
把壓縮後的資料序列放進vector容器，容器中分別放3個通道(R、G、B)的壓縮資料，每個壓縮資料以pair<uchar, int>存。

|  |
| --- |
|  |
|  |

**b.儲存資料**

要存進dat檔裡，可以先把完整格式存成string，再直接輸出成dat檔案(二元檔)。

**c.算壓縮率**



在c++中可以用Seek+tellg來算檔案byte數。

平均壓縮率 = 全部檔案的壓縮率加總 / 檔案次數

**成果**

檔案: img1.bmp 壓縮率: 2.46887

檔案: img2.bmp 壓縮率: 1.43345

檔案: img3.bmp 壓縮率: 2.69067

平均壓縮率: 2.19766