

Home Work4

組員：HW4_黃子青_E24076938_陳俊瑋 E24076603

程式撰寫邏輯：

因為這次作業的 inputmap 的 pixel 的 grayscale 只有 0(黑色)或 255(白色)兩種可能，而且黑色部分的 pixel 通常比白色部分的少，所以我選擇紀錄黑色的部分，這樣可以讓 output 的 txt 檔檔案變得比紀錄白色的部分小。

而讀取的方法是用一個二維的 vector 先把它設成圖片的 height 大小的一維向量，在讀一整個列(大小為圖片的 width)的時候，讓它每次遇到黑色 pixel 的時候就把那個 column 的起始位置 push_back 到那個 vector[row] 上，並設一個 booling 變數紀錄已經開始讀了，再設一個 sum 來記錄中間遇到幾個 column，接下來在遇到白色前會讓 sum++，一直到遇到白色為止，就把起始位置的 column 加上 sum 這樣就等於是終點位置，但要注意的是也有一種狀況是有可能圖片有一整段是沒有經過白色部分就直接到 column 的終點，所以我有用 if else 來判斷，假如這種情形發生，就直接在黑色部分且目前位置為 column 的終點的時候就把起始位置的 column 加上 sum 這樣就等於是終點位置。

跑完一次 row 後，我們用 int segment_num 來存取 vector[row].size()，這樣就可以存取上課簡報所說的 Number of segments in each row

之後把所有 row 都跑完後，就有了壓縮完的資料以及 Number of segments in each row，並把這兩筆壓縮資料 Output 到 txt 檔



compressed_version.txt



Number_of_segments_in_each_row.txt

之後再讀取這兩個檔案來解壓縮回去，先 fill(255)把圖片的底印成白色，之後再根據壓縮的資料，先讀取共有幾個 segment，再把起始位置到終點位置經過的 column 都印成黑色，重複 segment 次，這樣就印完一整列的圖了，也就是解壓縮原本圖片的一整列。

這樣子重複所有的列過後，就可以把原本的圖整個解壓縮回來！

程式執行結果：

等程式執行完後在資料夾底下點開 output_image，即可查看

壓縮過後再解壓縮回去的圖

Debug	2020/4/25 上午 11:14	檔案資料夾	
plugins	2017/2/28 上午 01:21	檔案資料夾	
resources	2017/2/28 上午 01:21	檔案資料夾	
CImg.h	2017/2/20 下午 06:31	C/C++ 標頭	2,826 KB
compressed_version	2020/4/25 上午 11:14	文字文件	15 KB
HW4_黃子青_E24076938	2020/4/25 上午 11:15	C++ Source File	7 KB
HW4_黃子青_E24076938.vcxproj	2020/4/22 上午 12:03	VC++ 專案	9 KB
HW4_黃子青_E24076938.vcxproj.filters	2020/4/22 上午 12:03	VC++ 專案篩選檔...	2 KB
HW4_黃子青_E24076938.vcxproj.user	2020/4/22 上午 12:03	Per-User Project ...	1 KB
input_binary_image	2017/4/9 下午 12:41	BMP 檔案	1,055 KB
Number_of_segments_in_each_row	2020/4/25 上午 11:14	文字文件	2 KB
output_image	2020/4/25 上午 11:14	BMP 檔案	1,055 KB
stdafx	2020/4/22 上午 12:03	C++ Source File	1 KB
stdafx.h	2020/4/22 上午 12:03	C/C++ 標頭	1 KB
targetver.h	2020/4/22 上午 12:03	C/C++ 標頭	1 KB
ultraman_8bit	2020/4/25 上午 12:28	BMP 檔案	89 KB
範例程式	2020/4/22 下午 04:21	C++ Source File	1 KB

原圖



壓縮過後再解壓縮回去的圖

檔名取為 output_image

