電機所控制組 110318051 張皓博

HW2\_handmade講解

我使用Visual Studio來寫這個C++作業，執行方式為把.exe檔跟hand.bmp放在同個資料夾底下後，執行.exe即可產生輸出圖檔及command資訊，最後有bounding\_box的圖為drawingImage.bmp，手中間的紅點為centroid。

這個作業不用修改到header，所以我將header直接以54bytes的vector儲存。

產出兩張圖：binaryImage.bmp、drawingImage.bmp。

產出command資訊：



總共有13個function：

1. **void bmpRead(const char\* pictureName):**

先利用fopen開啟傳入的Image，然後啟用讀取二進位的模式，利用fread函式將header及raw data各存成一維vector。

1. **void bmpWrite(char const\* pictureName, vector<unsigned char>header, vector <unsigned char> pixels\_Output, unsigned int width, unsigned int height):**

可以傳入圖片名稱、圖片的header資訊、圖片的raw data資訊以及圖片的長寬，方式跟bmpRead是一樣的，不同的是從fread變成fwrite，將header資訊跟改造後的raw data依序寫回圖片。

1. **vector<vector<unsigned char>> oneDimension2twoDimension(vector<unsigned char> pixels\_input):**

用以把vector從一維512\*512\*3轉為二維512\*(512\*3)的function。

1. **vector<unsigned char> twoDimension2oneDimension(vector<vector<unsigned char>> pixels\_input2D):**

用以把vector從二維512\*(512\*3)轉為一維512\*512\*3的function。

1. **vector<unsigned char> rgb2gray(vector<unsigned char> pixels\_input):**

gray = 0.2989 \* r + 0.5870 \* g + 0.1140 \* b

用以把RGB圖轉為灰階圖，利用權值計算後，BGR三個值寫入一樣的值使得維持channels為3的灰階圖。

1. **vector<unsigned char> gray2binary(vector<unsigned char> pixels\_input):**

利用threshold比大小的方式把gray scale轉為binary image，一樣將BGR三個值寫入一樣的值使得維持channels為3的灰階圖。

1. **vector<unsigned char> dilation(vector<unsigned char> pixels\_input):**

這裡將filter設為十字，dilation做法是判斷觀察點如果為白點, 那將觀察點設為原點的十字區域(上下左右)都成為白點。

1. **void giveLabel(vector<vector<unsigned char>> &pixels\_label2D, int row, int col, int label):**

由於在connected component時需要大量的寫到給BGR三個一樣的值，故獨立出一個function以節省版面。

1. **int find\_root(int x):**

利用遞迴的方式，尋找label的根點並更新，就可以找完剩下衝突的label。

1. **vector<unsigned char> connectedComponents(vector<unsigned char> pixels\_input):**

這裡的connected component我使用課堂說到classic algorithm的作法搭配equivalence Label的作法

* + 1. 先找到binary image的白點，再做邊框判斷，避免陣列空間超出框架。
    2. 確認無超出框架後，判斷左邊跟上面的label，如果皆為0，給予新的label，如果有衝突，選擇小的label接續，同時記錄衝突label。
    3. 利用find\_root()解決剩餘衝突的label。
    4. 將更新後的equivalence label寫回圖片，為正確的label image。

在function最後面整理equivalenceLabel的vector，取得每種label的值當作後續作圖判斷使用。

1. **void properties\_analysis(int row\_max, int row\_min, int col\_max, int col\_min):**

藉由bounding box的長寬作判斷，可以比較出longest length跟手是上下向還是左右向。

1. **void bounding\_box(vector<unsigned char>& pixels\_binary, vector<unsigned char>& pixels\_label, unsigned char label, int part):**

判斷同一種label的最大最小row、col，可以知道邊框座標，運算bounding box的長寬，再慢慢一個點一個點畫紅色上去，形成紅色的bounding box。

1. **void centroid\_area(vector<unsigned char>& pixels\_binary, vector<unsigned char>& pixels\_label, unsigned char label, int part):**

加總每個區域白點的row和col，再除以白點的數量(area)，就可以得到centroid座標(平均座標)。

上一次的作業我大多使用char []，再這次寫作業的過程中發現char []要return起來非常麻煩，傳資料也需要使用pointer(\*)容易出錯，就大肆的爆改成全部都使用vector儲存，其中遇到最大的麻煩是如何解決label衝突的問題，在第一次衝突解決後，還會存在一些因為解決衝突順序倒反的問題(例如5→3，3→2，5卻沒有變成2)，剩餘的衝突label在我查了許多資料後，想到find\_root()的遞迴才解決，經過兩次的作業，我在coding上處理得更簡潔乾淨，看了很愉悅！