Advanced Computer Vision

HW. 2

姓名: 李訓廷

學號: 108318120

**Describe the employed source code editor and how to execute your program:**

*使用 VScode 作為開發程式碼之 IDE，因相較於 Visual Studio 較為輕量，並使用mingw64 - g++ 做程式的編譯。*

*C:\LLVM\bin\g++.exe -g -std=c++11 e:\C-project-cv\HW1\hw2\_cv.cpp -o hw1\_cv.exe -I …*

*後面則須使用 -I 帶入 opencv 所安裝之路徑。*

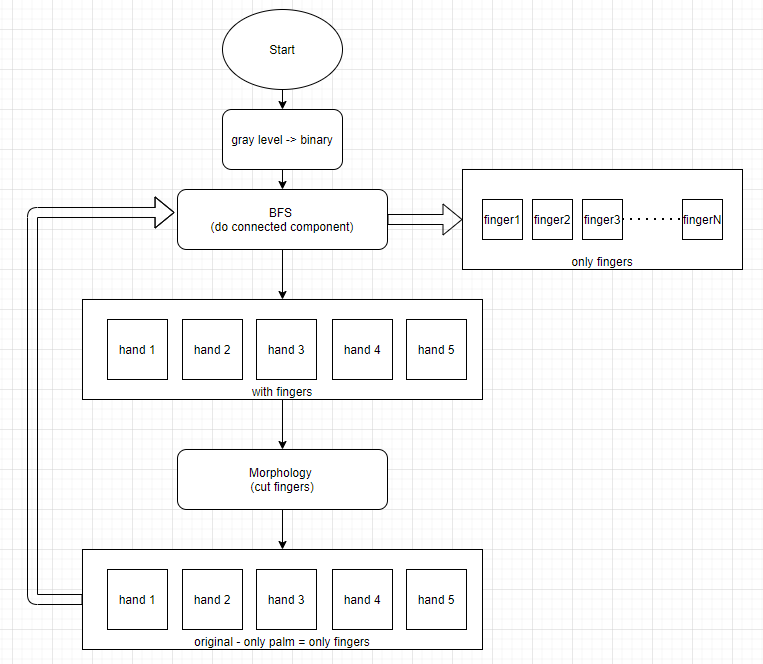
Hw2.exe輸入圖片路徑 (XXX/hand.bmp)

Output: out.bmp

Hw2\_cv.exe輸入圖片路徑 (XXX/hand.bmp)

output out\_cv.bmp

**Introduce your work, method, and discussions:**



**Not using opencv(hw2.cpp):**

首先使用 BFS 遞迴方式找出二值化後的 connected components

void BFS(RGB\* *pixel*, int *idx*, int *row\_idx*, int *col\_idx*, ccp\* *comp*)

並使用 struct 來存儲每一個compnents 的內容

typedef struct components

{

    int total\_pixel = 0;

    int total\_x = 0, total\_y = 0;

    int min\_x = 0, min\_y = 0, max\_x = 0, max\_y = 0;

    bool handVertical = true;*// mode handVertical the hand is vertical: true  or  horizontal: false*

    int finger\_cnt = 0;

}ccp;

根據得到的每一隻手(共五隻)

去做 morphology 找出去掉手掌的圖

find\_fingers(img\_binary, &comp, img\_finger, components\_counts);

結果如下圖



然後再次執行 BFS 找出手指頭的 connected components

找手指頭 components 過程中若找到手指頭則根據 整隻手的 min max x y 以及手指頭的 centroid 去判斷該手指屬於哪一隻手並執行 struct components 裡的手指頭計數++

(hand\_list + h)->finger\_cnt++;

用紫色的線畫在結果圖中，幾條就代表幾根手指頭

最後透過

void plot\_bbox(RGB\* *img*, ccp\* *comp*, int *rgb*[])

根據每隻手、跟手指頭的components strcut 裡獲得的值、計算得到的值畫在二值化的圖上。

**Discussion:**

1. Morphology 中根據手掌是

垂直: 使用水平細長的 rectangle mask (3 \* 27)

水平: 使用垂直細長的 rectangle mask (27 \* 3)

來執行 opening 先erode 後 dilate

1. 手指的 components 有找出很多較小且不是手指頭的區域，透過 area 大小把較小的區塊篩掉，最後才得到是真正手指的區塊

**Using opencv(hw2\_cv.cpp):**

流程基本跟 C 版本一樣，只是都使用 cv 的套件來完成。

**Discussions:**

1. 不同於C版本的是這裡做了兩次 opening ，rectangle mask (25 \* 25)做第一次，ellipse mask (25\* 25) 做第二次，使得在找出手指頭的部分得到了比C版本更細緻一點的結果

k = getStructuringElement(MORPH\_RECT, Size(25, 25), Point(-1, -1));

morphologyEx(dst, dst, MORPH\_OPEN, k);

k = getStructuringElement(MORPH\_ELLIPSE, Size(25, 25), Point(-1, -1));

morphologyEx(dst, dst, MORPH\_OPEN, k);

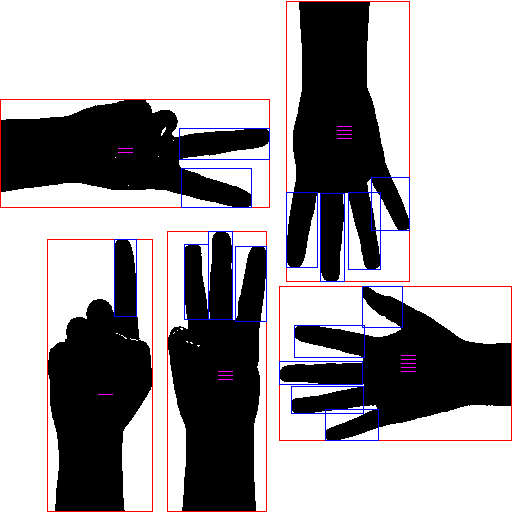
1. 因為可以使用 cv 故有些結果直接使用 putText 畫在結果圖上，方便許多

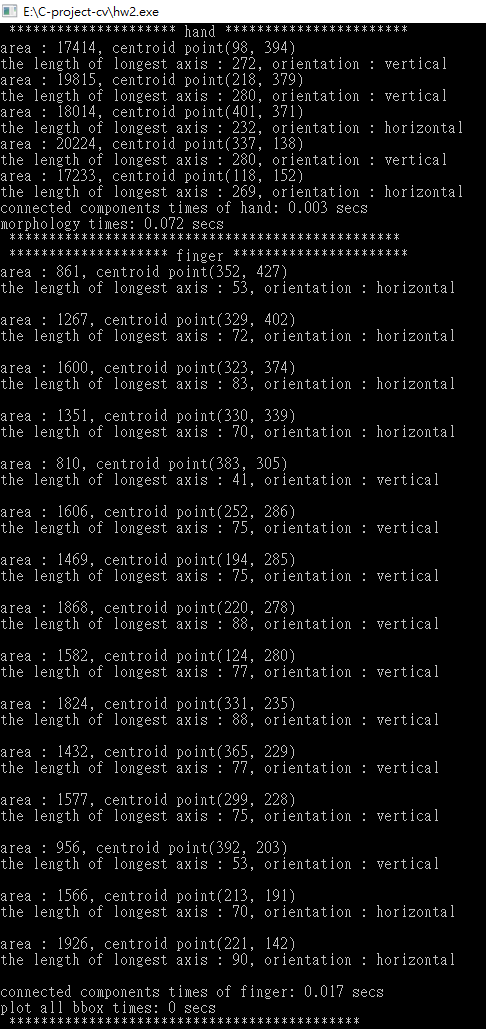
**總結:**

以上兩支程式都使用 timer做計時，可以發現在 connected components、morphology 速度都是自己刻的好幾倍，但plot 在兩個版本得到的結果都為0，判斷是畫圖自已實現的複雜度的確不高或者使用的計時 library 精度不夠高之原因。

**Results:**

Without opencv:





With opencv:

