

Advanced Computer Vision

HOMEWORK 2

Lab : VPILab

Advisor : Cheng-Ming Huang(黃正民)

Student : Yu Cho(卓諭)

Student ID : 106318025

Data : 2017.09.26

I. Problem

A. Connected components

- Generate a binarized image (threshold 128) of lena512.bmp
- Write a C or C++ program to analyze the connected components with 4-connected neighbor.

B. Bonus

II. Method

A. Connected components

- C++ Program

1. 先將 lena512.bmp RGB 色彩空間像素資訊經由

$\text{Gray} = 0.299 * R + 0.587 * G + 0.114 * B$ 這個公式將三原色轉為灰階值，再將這個灰階值丟入三個通道中。完成 RGB 轉灰階。

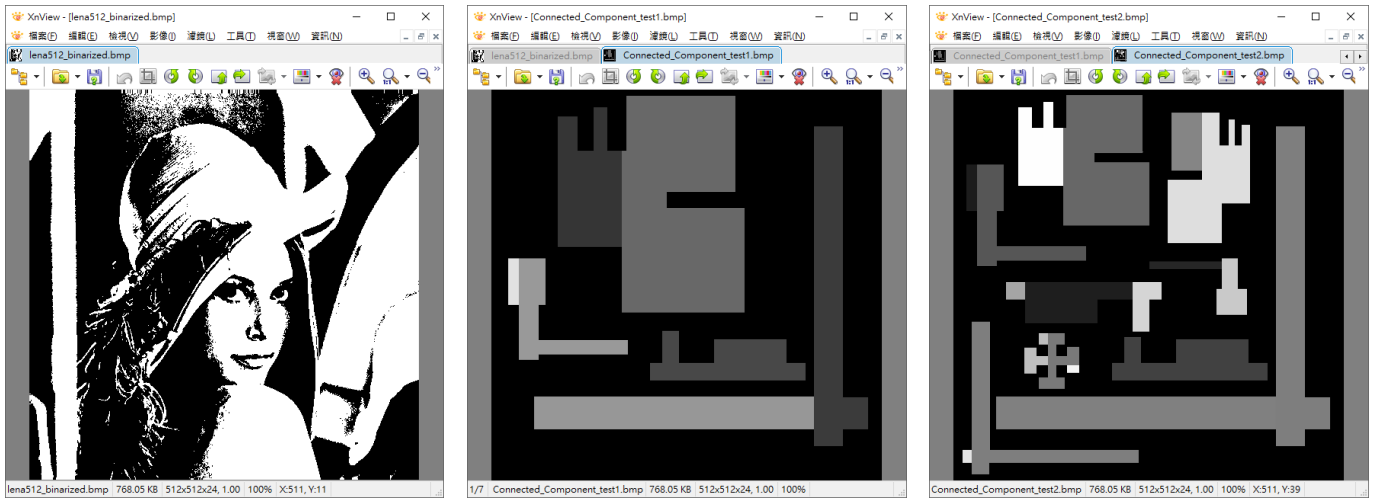
2. 設定閾值為 128 將像素點大於 128 的值設為 255，小於等於 128 的像素值設為 0。完成二值化。
3. 用 A space-efficient two-pass algorithm 方法取得 Connected components，此方法一開始都與 The Classical Algorithm 相同，差在多了兩的矩陣，分別為 equivalence pair、equivalence table。equivalence pair 存放當走訪的像素點為 255 且上面及左邊有標籤過，發生衝突兩像素的 Label。equivalence table 則為將這些 pair 的像素所有相關連性重新整理，得到一個列為幾個抓到的區域，行是有幾個第一次標 Label 時不同 Label 但卻是同一個區域的資訊。

- C++ Program OpenCV

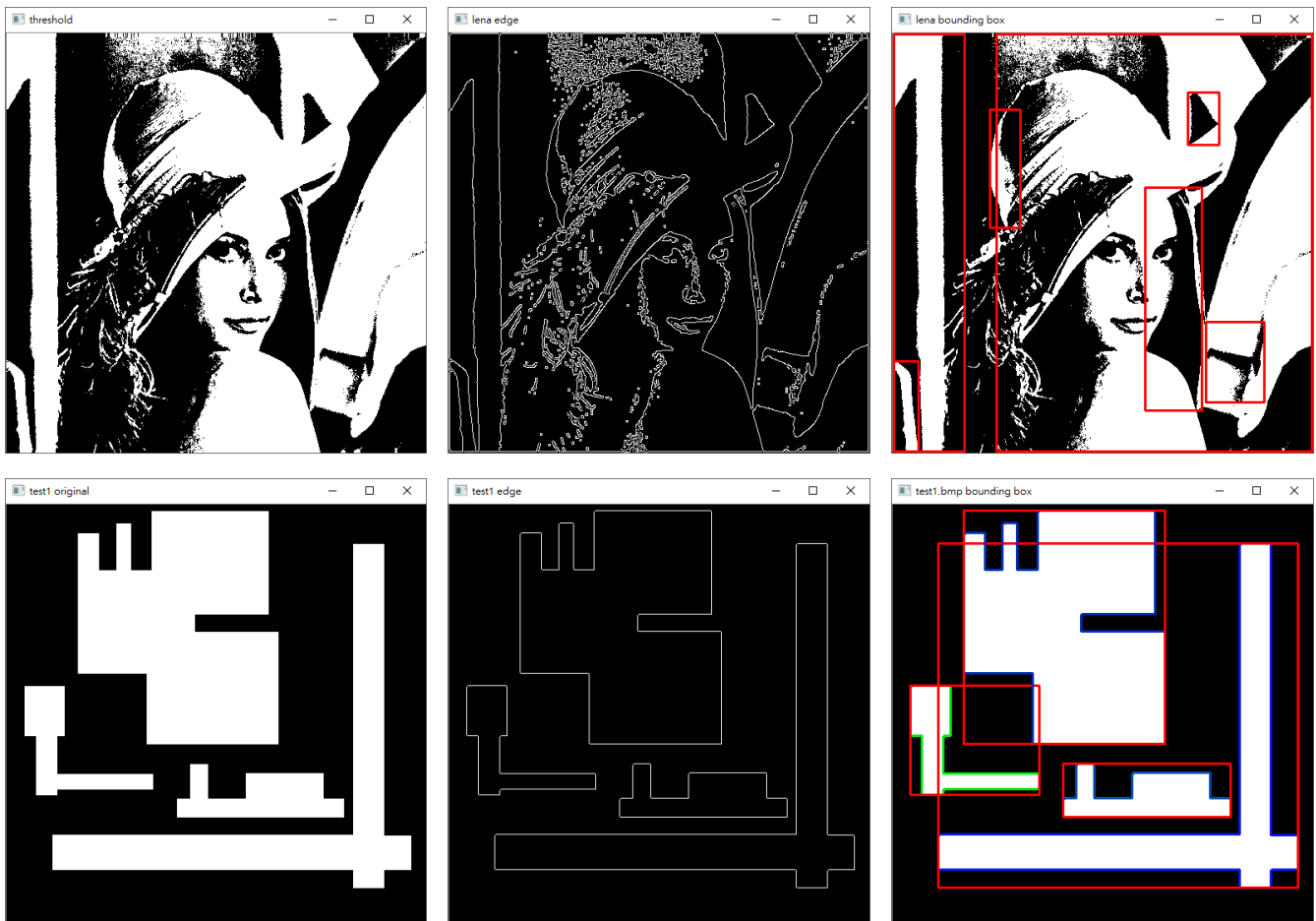
1. 利用 `threshold()` 將輸入的 Mat 矩陣中的影像透過一個 128 閾值做二值化並存入另一個 Mat 中。
2. 利用 `findContours()` 找出輸入的灰階二值化影像的 contours 並存入由 `vector<vector<Point>>` 宣告的變數中。
3. 寫一個 for 迴圈將極大值設為 `contours.size()` (指的是有幾個 contours)，利用 `contourArea()` 得到該 contour 的面積，並下判斷式，若大於面積 1000 則將此 contour 用 `boundingRect()` 找出範圍搭配 `rectangle()` 函數將 bounding box 繪出。

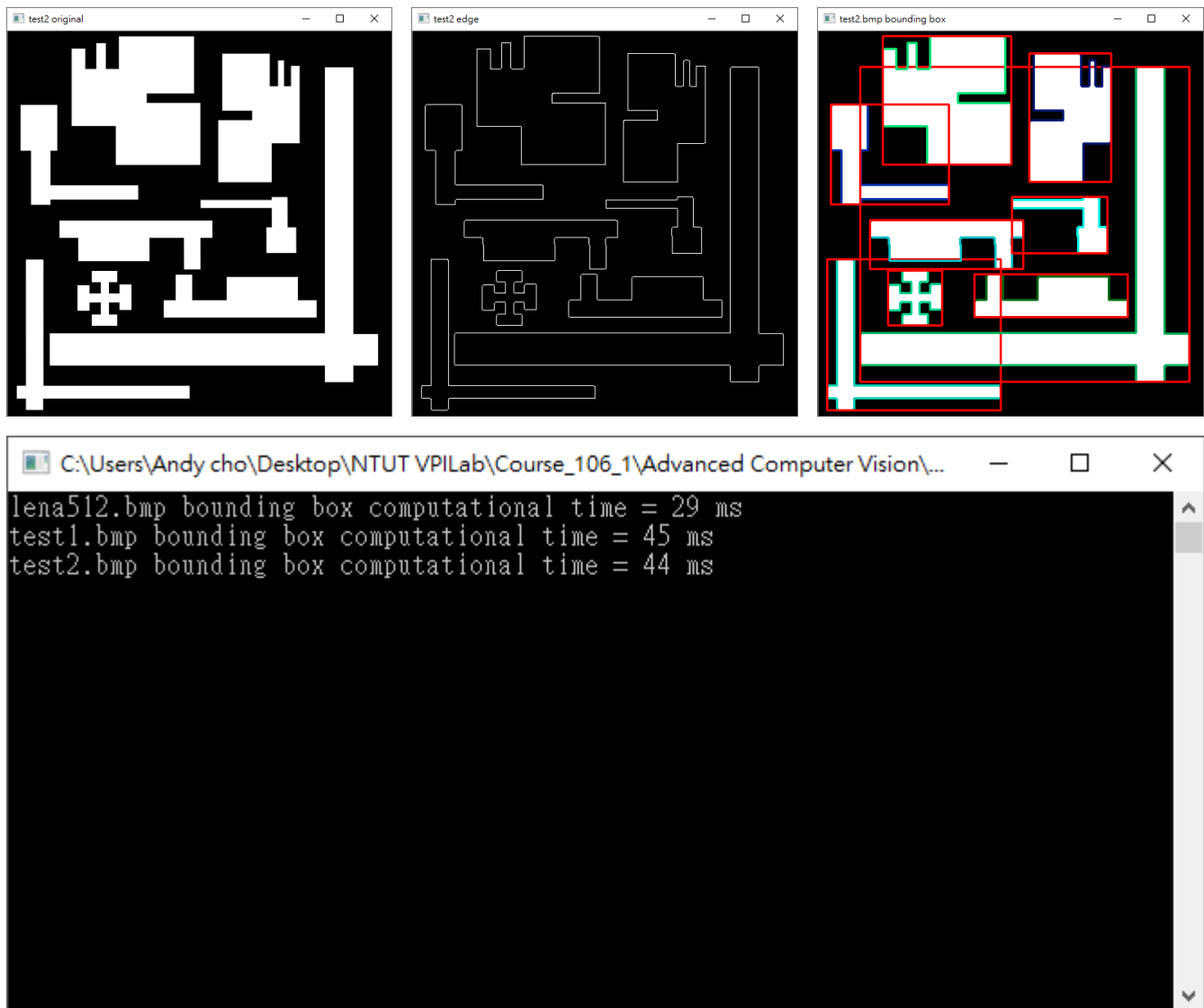
III Results

- C++ Program



- C++ Program using OpenCV





IV. Discussion

在 C 語言的演算法中，

1. 會看到 Connected components 得區域取得沒有正確顯示，我認為是在第二次要使用 equivalence table 更新 Label 時輸出矩陣並沒有正確更新到，因此出來的結果與 The Classical Algorithm 只掃一次的結果並沒有差別，原因是此方法就是為了省去來回掃描的方式浪費記憶體，所以只掃一次並將 Label 資訊存出，但我城市中的 BUG 為沒有正確將相連接的區域更新為同一個 Label。
2. 在 equivalence table 時的二為矩陣宣告會出現崩潰，問題是在不確定矩陣大小下要宣告矩陣必須使用動態記憶體，而我只使用此方法在宣告一為陣列過，同樣的宣告在二為陣列會出錯，我想是因為指標的問題，大致的解決方法是將原本宣告的二維振烈改成之前熟悉的一維陣列，是否能夠正確執行還沒有測試故不知道是否能夠完美解決問題。
3. 最後的區域著色也尚未解決，只暫時先以隨機的灰階值顯示區域。