

HW1 Report

P76114082 歐禮寬

1. RGB Extraction & transformation

(1) method

R,G,B 的轉換透過分別取出對應的channel並將pixel 的 R,G,B 都設為該channel 的值，gray scale 則透過公式 $R * 0.299 + .G * 0.587 + B * 0.114$ 得到並將 R,G,B 都設為該值。

(2) result => R,G,B,Gray



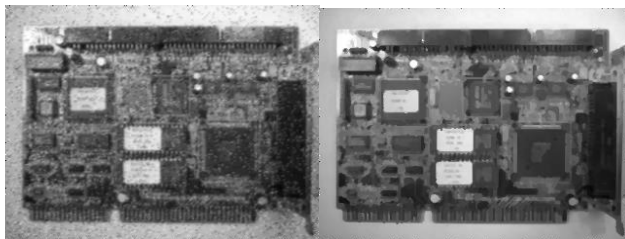
2. Smooth filter (mean and median)

(1) method

Mean filter 將每個 pixel 跟周圍8個 pixel 相加並平均，得到的值為 pixel 新的值

Median Filter 將每個 pixel 與周圍8個 pixel 排序在取中間的那個值，該值為 pixel 新的值

(2) result => mean, median



(3) discussion

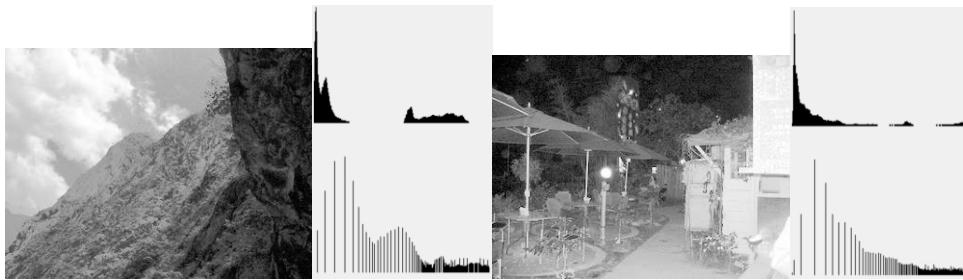
直接看的話 median filter 能有效的去除 noise，但圖片較模糊感覺像細節感覺都不見了

3. Histogram Equalization

(1) method

先算出每個灰階值出現的次數及機率，再根據此機率把每個灰階值對應到新的灰階值，最後更新圖片的每個 pixel，直方圖則是透過各 pixel 與出現最多次的 pixel 比例來繪製在 Bitmap 上面

(2) result



(3) discussion

透過直方圖可以直接觀察到原本許多 pixel 都集中在較暗的地方，而透過此方法則將其分散到其他地方

4. A user-defined thresholding

(1) method

根據 trackbar 的值，將大於等於的 pixel 都變為白色，其他則變為黑色

(2) result



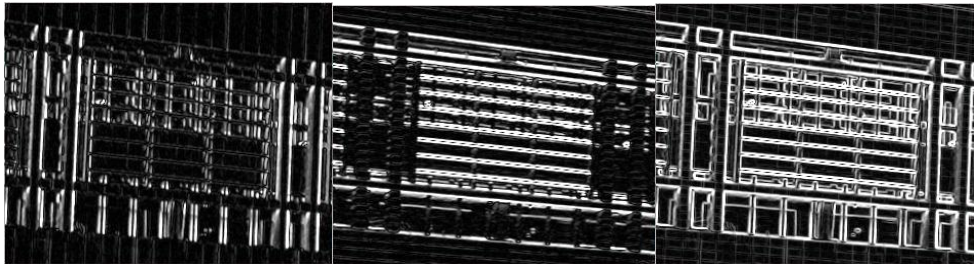
5. Sobel edge detection

(1) method

透過 sobelx operator 得到垂直的邊，sobely operator

的到水平的邊，兩者合併可得到垂直及水平的邊

(2) result => vertical,horizontal,combined



(3) discussion

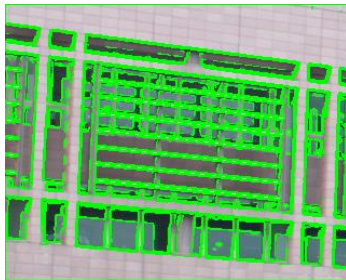
sobelx 及 sobely 運算完要注意將負值取正，而大於 255 的值也要記得設為 255 避免非預期的錯誤

6. Edge overlapping

(1) method

先根據 trackbar 的值，將大於等於的 pixel 都變為白色，其他則變為黑色，再將白色邊的部分用綠色畫到原圖上

(2) result

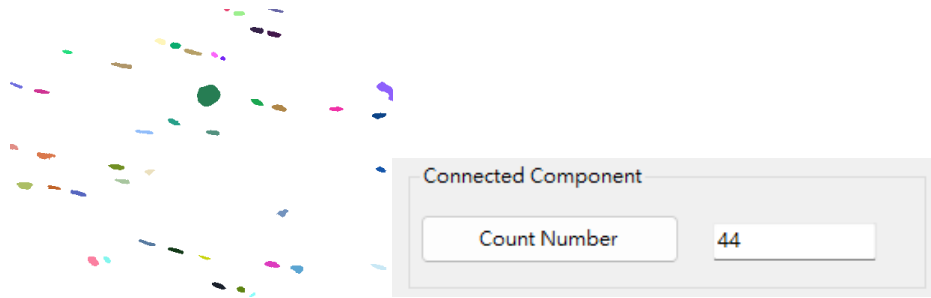


7. Connected Component

(1) method

先將白色的部分設為拜訪過，在對每個未拜訪過的 pixel 進行 dfs 並將 connected component 數量加一，中間拜訪到的 pixel 也都會設為拜訪過，顏色則透過random的方式決定

(2) result



(3) discussion

若單一個 connected component 太大，會因為 dfs 遞迴呼叫太多次導致 stack overflow，因為想不到合適的方法找出下一個不一樣顏色的 RGB value，所以透過 random 的方式取值，不過這樣顯然有機會出現重複的顏色。

8. Image registration

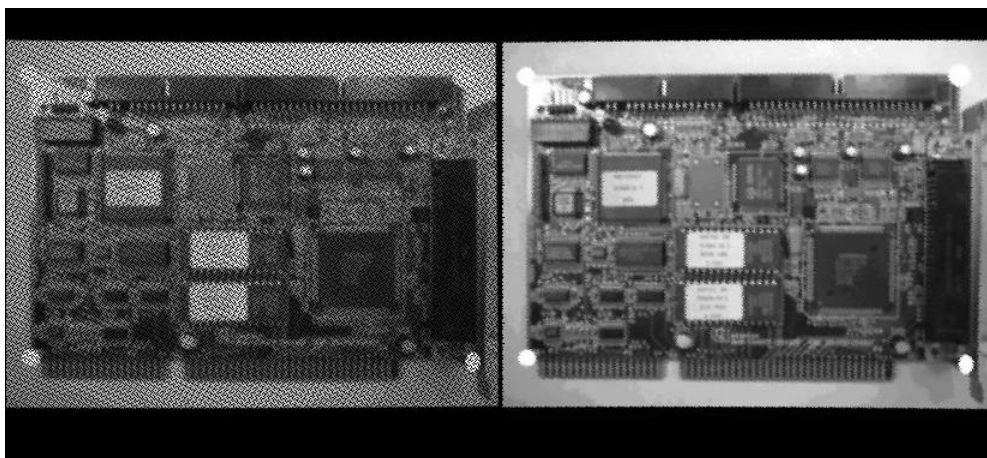
(1) method

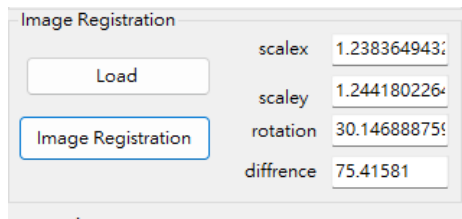
透過提供的4個 tie point 利用高斯消去法解 $x = c1*v + c2*w + c3*v*w + c4$ 及 $y = c1*v + c2*w + c3*v*w + c4$ ，再用解出的式子將 imageB 的 pixel(v,w) 對應到 imageA 的 pixel(x,y)

scalex 透過兩圖左上和右上的點之間的距離比例求出，scaley 則透過兩圖左上和左下的點之距離的比例求出

rotation 則透過兩向量的內積 dot 及行列式 det 再透過 $\arctan(\text{dot}/\text{det})$ 求得，兩向量分別為 imageA 的左上點到右上點及 imageB 的左上點到右上點乘以 scalex

(2) result





(3)discussion

可能是因為 `scale` 及 `rotation` 的關係，`imageB` 每個 `pixel` 並沒有一一對應到 `imageA`，所以產生的圖會有洞，我這裡只是簡單用最近的 `pixel` 來把它補起來

我覺得可能是因為滑鼠在 `imageA` 及 `imageB` 上點的點不一定是正確對應的點所 `registered` 後的圖會歪歪的。

Conclusion

這次作業讓我學會了許多影像處理的方法，更重要的是可以不透過現有的 `library` 來達成，讓我覺得蠻有成就感的！