

Image Processing HW8+

0316206 葉家瑋 資工系三年級

Programing language: C++

OpenCV version (if needed): 3.2

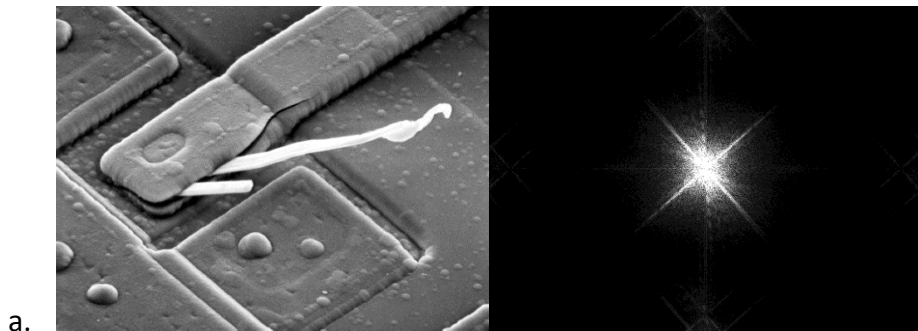
Develop environment: Visual Studio 2012

A. Program flow

- I. 首先先讀黨，再來創造一個儲存結果的矩陣，再來先對 **row** 做一次傅立葉轉換，這樣的時間複雜度只要 $O(n^3)$ ，就不會跑很久而當掉了，而我裡面的作法是用一個 **template** 的型態叫做 **complex** 來做運算，而因為是只對 **row** 來做傅立葉轉換，所以只變 **row**，**column** 不動，所以才會有 `newtemp/complex<double>(cos(pi*2*i*u/SrcImg.rows), sin(pi*2*i*u/SrcImg.rows))` 這樣的算式產生。
- II. 再來就是對 **column** 做傅立葉轉換，但是這次的藥用上面已經做過一次傅立葉轉換的數值來繼續做運算，而當最內圈的 **for** 迴圈做完運算，就要對此點做除上一個常數的動作，不然會是全白的，再來在做取絕對值的動作，也就是 `sqrt(pow(tempflash[u][v].real(),2)+pow(tempflash[u][v].imag(),2))`，而如果超過 255 的質我們要把它回歸 255 不然會有雜訊，最後再將結果的值存入結果的矩陣。
- III. 最後就是將圖片重組，因為做完傅立葉轉或左上方會變成原點，要把原點放回中心，最後就是把圖 **output** 出來。

B. Result and Analysis

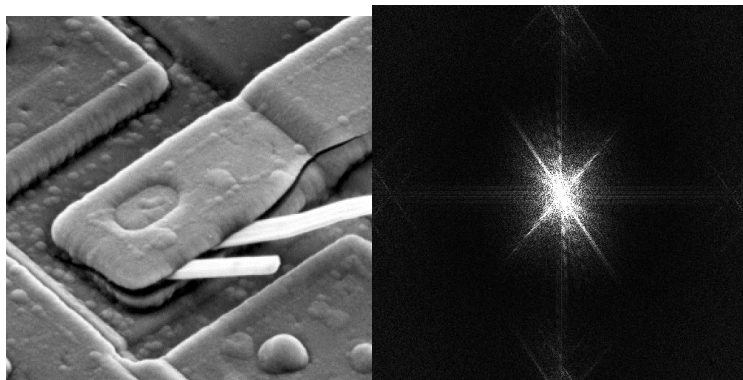
I. Fig. 4.29



a.

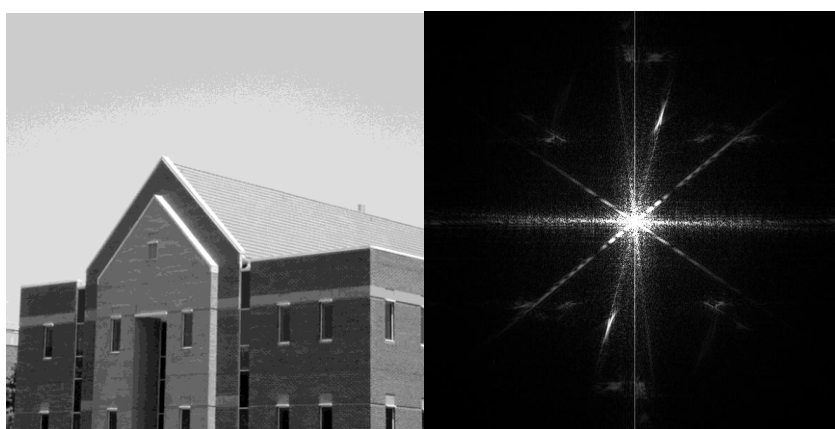
- b. 跟課本上的比，我的中間白色的部分比較多，比較少雜訊，但是因為這樣，中間那條線的中心附近的類似虛線的部分就比較不明顯。

II. Fig. 4.36



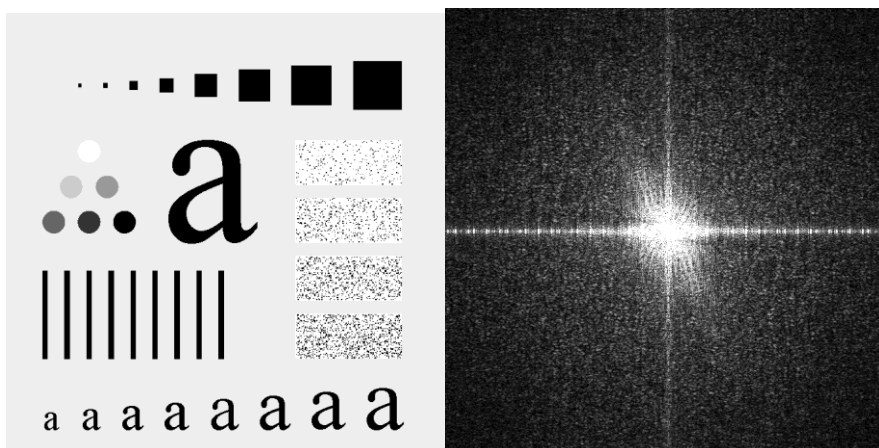
- a.
- b. 其實這張圖得比較結果跟第一張很像，中間白色的部分太多以致於最中心的部分無法看清楚。

III. Fig. 4.38



- a.
- b. 也是中間太白了以致至於中心點無法看清楚，而這張圖更不一樣的的部分就是可本上的比較灰，而我的圖比較暗。

IV. Fig. 4.41



- a.
- b. 也是中間太白了以致至於中心點無法看清楚，而這張圖是比較接近課本的了。