

Hw2_Image Sharpening

資工二 409410024 陳品希

Data due:2022/4/30

Data handed in:2022/4/29

- Technical description

我參考了上課的 ppt，使用 c++ 搭配 opencv 來實作 Image Sharpening

Laplacian operator :

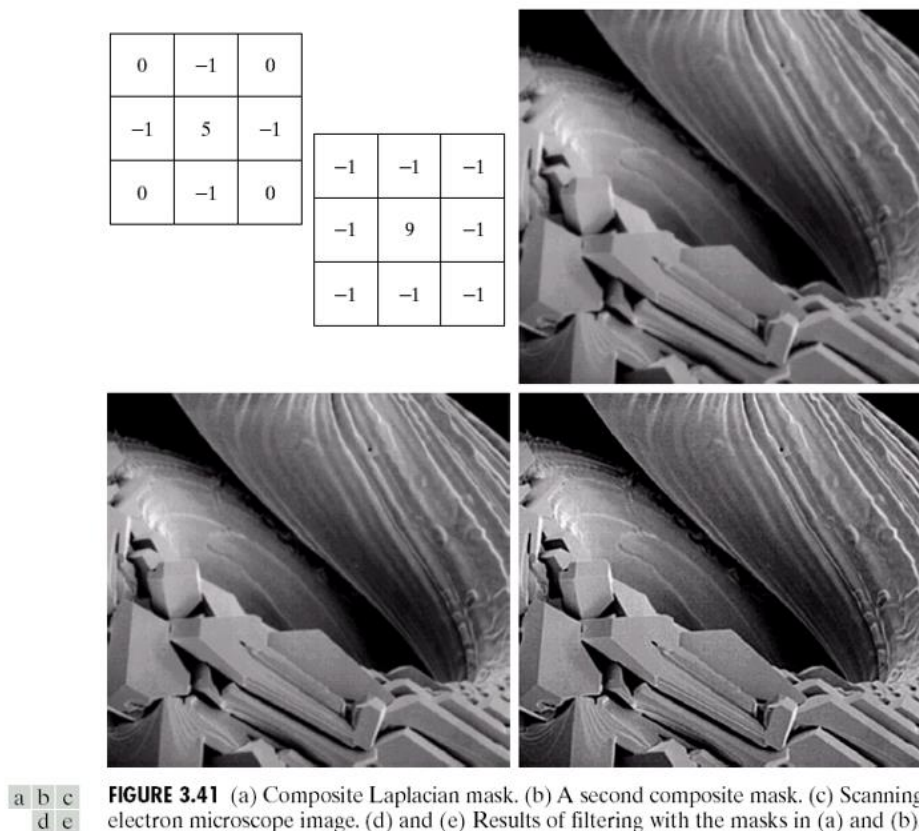
基本上就是對每個點做如下公式

$$g(x,y) = f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] + 4f(x,y)$$

$$= 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)], \quad (\text{if } g(x,y) \text{ is outside } 0 \sim 255)$$

若計算完值不在 0~255 之間(溢位)，則若值小於 0 就改成 0，值大於 255 就改成 255。然後產生完的值就能形成目標圖片。

下面兩種其實都行，我用的是左邊那種。



high-boost filtering :

基本上方法都跟 Laplacian operator 差不多，不同的是中間的值

- High-boost filtering can be implemented with one pass using either of the two masks shown in Fig. 3.42. Note that, when $A=1$, high-boost filtering becomes “standard” Laplacian sharpening.

0	-1	0	-1	-1	-1
-1	$A + 4$	-1	-1	$A + 8$	-1
0	-1	0	-1	-1	-1

a b

FIGURE 3.42 The high-boost filtering technique can be implemented with either one of these masks, with $A \geq 1$.

如上圖，high-boost filtering 的 A 要大於等於 1，而若 A 等於 1 則就是 Laplacian operator。我這裡是把 A 設成 1.5，並把 9 個格子都除以 A ，避免圖片整體變太亮。

如何執行:

我的程式檔案為 `hw2.cpp`，執行檔為 `hw2.exe`

程式會透過 `argv[1]` 讀入圖片的檔名，故移動到適當位置，輸入指

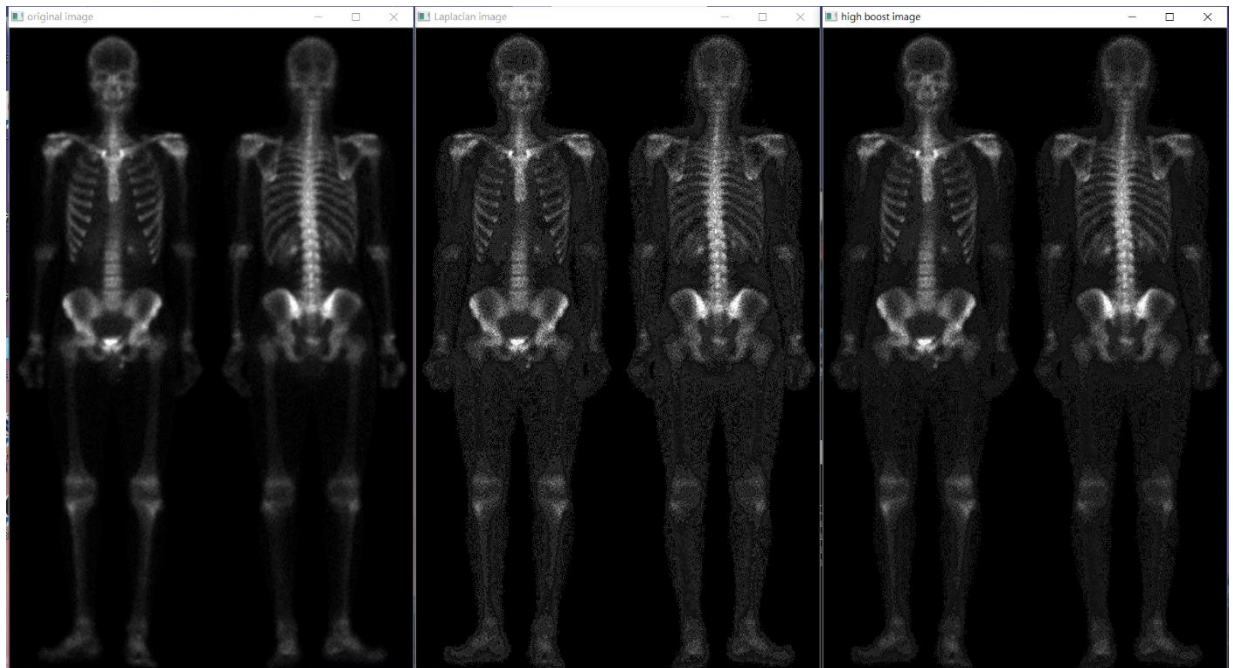
令：`./hw2.exe [檔名]` 即可執行程式。

※注意：程式碼必須跟檔案內附的所有.dll 檔放在同一個資料夾才能執行

- Experimental results

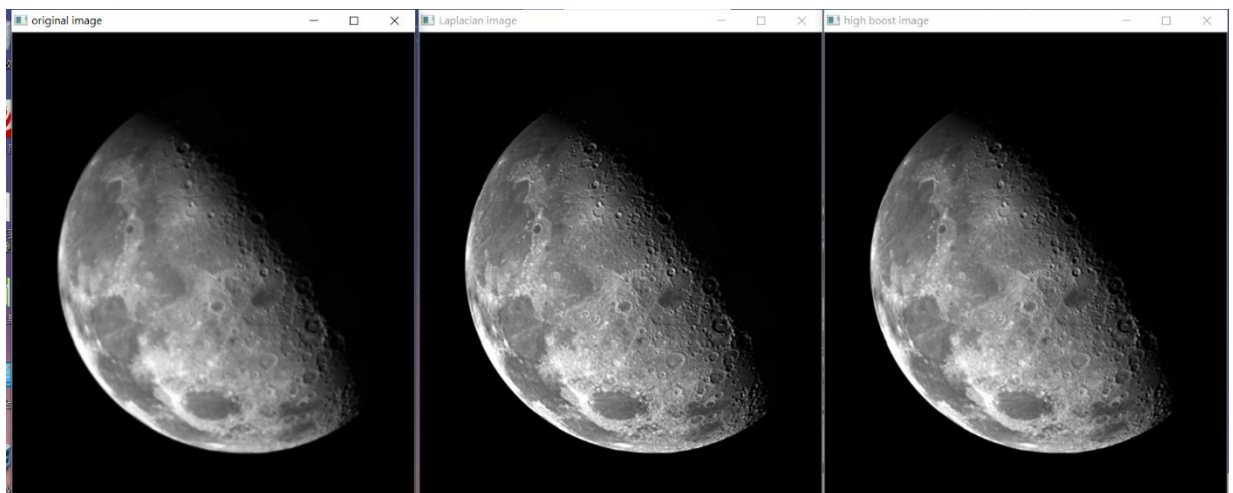
輸入指令 => ./hw2.exe [檔名]

1. ./hw2.exe skeleton_orig.bmp



由左而右分別是 original image, Image Sharpening(Laplacian operator),
Image Sharpening(high-boost filtering)

2. ./hw2.exe blurry_moon.tif



由左而右分別是 original image, Image Sharpening(Laplacian operator),
Image Sharpening(high-boost filtering)

※若截圖看不清楚的話，我另外附了.png 的截圖檔，放在名為測試截圖的資料夾內。

- **Discussions**

這次作業意外的沒有那麼難，但其實我還是寫了很久，主要原因在於我數學不好、英文也不好，整個無法理解要怎麼做。雖然期中考其實有考過這部分了，但果然理論跟實作還是有一定的差距呢！理解了該如何實作後，剩下的部分就變得簡單了許多。經過這兩次作業，我其實覺得寫這種作業還蠻有成就感的，因為我可以很直觀的感受到我到底做了什麼。但我不敢要求多出幾次作業，因為如果多出幾次，我可能會在獲得成就感之前先累癱。希望之後的作業也跟這次一樣有趣，也希望我都能順利寫出來。

- **References and Appendix**

<http://163.13.127.10/pages/cht/courses/yen/proj105/pdf/day4.pdf>