

影像處理 LAB2

103062135 施乃仁

3_1 Image Enhancement Using Intensity Transformations

做法說明

兩種作法都是直接代公式進去就好。而邊界的部分，log transform 因為需要取 $\log(1)=0$ ，可以先把 (0~255) 加 1 變成 (1~256) 再取 log，會是 (0~ $\log(256)$)，最後乘上 $255/\log(256)$ ，就能把 (0~255) 對應到 (0~255)。至於 power law transform，就乘上 $255/(255^r)$ ，便能從 (0~255) 對應到 (0~255)。稍微需要注意的只有取 log 和 power 前要轉 type。

結果圖片

原圖

log transform



Power transform $r = 0.2$

Power transform $r = 0.4$



Power transform $r = 0.67$



Power transform $r = 1.5$



Power transform $r = 2.5$



分析以及討論

這題要求使用 log transform 和 power transform 來做 enhancement。Log transform 的特色是會將接近 0 的區域變明顯(也就是偏黑的區域)，整體就能讓原本較黑的區域變白變明顯。至於 power transform 則是依係數而定，如果 $r > 1$ 就是強化較黑區域的辨識度(整張圖會變白)，如果 $r < 1$ 就是強化較白區域的辨識度(整張圖會變黑)。

3_2 Histogram Equalization

做法說明

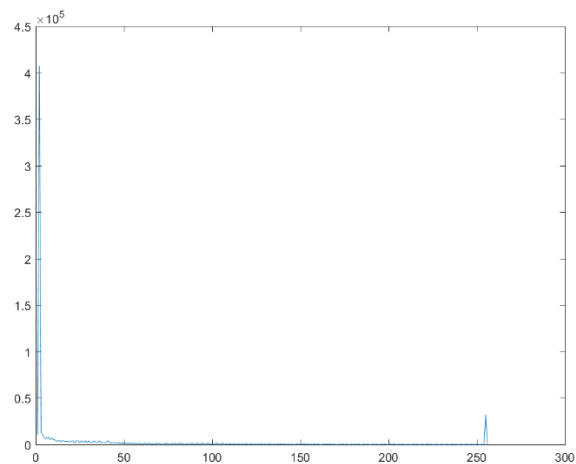
imageHist 是要算不同值在 matrix 裡出現的次數，因此就把整個 matrix 遍歷一遍，將對應 index 的值加 1。另外因為 matlab 裡的 index 是從 1 開始，而值的範圍是 (0~255)，因此在轉換時要記得 $\text{index} = \text{值} + 1$ 。而 histEqualization 就先開個變數存 sigma，然後就照公式以及 imageHist() 算出 sigma 的結果，而 sigma 的值也可以沿用下去。照這樣得到轉換 table 後，就能將原圖轉換，將 $\text{input}(i, j) \rightarrow T(\text{input}(i, j) + 1) \rightarrow \text{output}(i, j)$ ，而 input 和 output 都是 (0~255)，但 table 是將 (1~256) 對應到 (1~256)，因此在箭頭處分別要加一和減一。

結果圖片

原圖



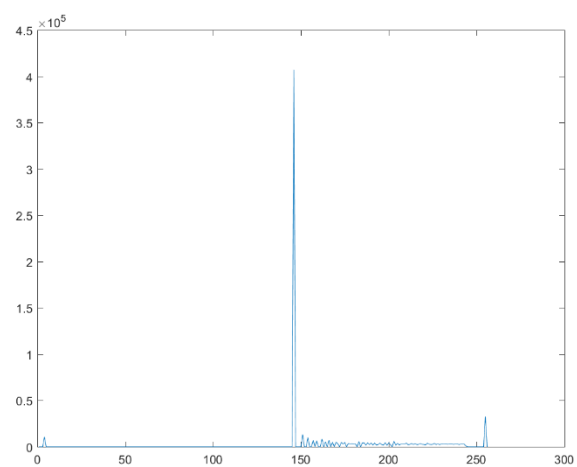
原圖 histogram



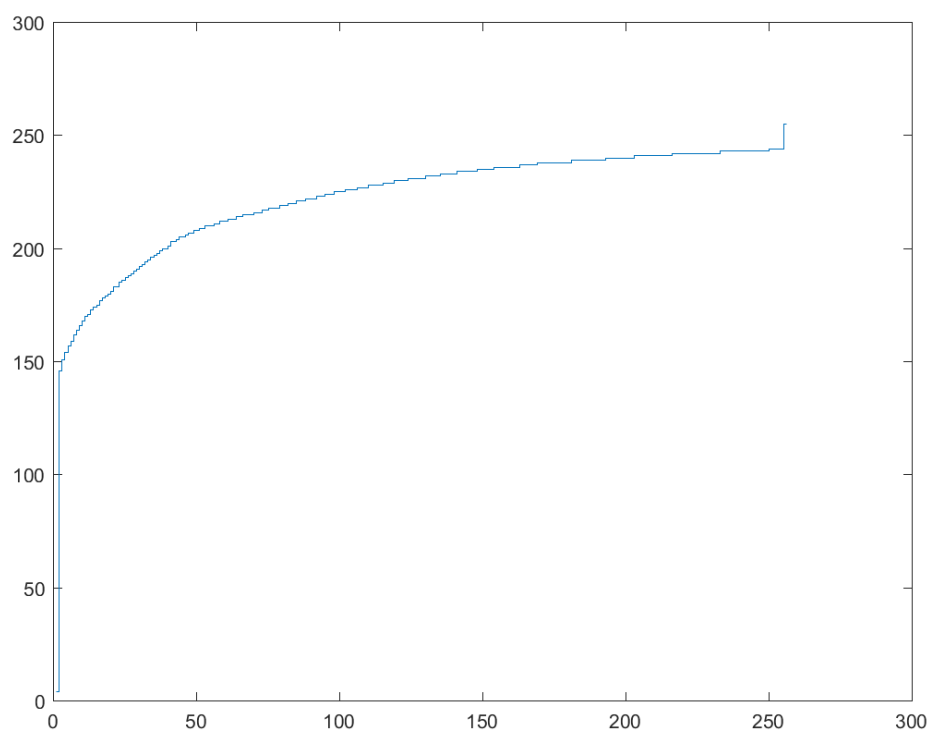
Enhance 後



Enhance 後 histogram



Transform table



分析以及討論

觀察 histogram 可以發現原圖的 pixel 大多是分布在偏黑的區域，因此 Transform table 就將這些偏黑的點的亮度拉高，看一下 Enhance 後的 histogram，發現會變成

集中在中間的區域，可能是因為某一值相同的 pixel 實在太多，無法將其分散，但至少轉換後的圖 histogram 有比較平均。

3_3 Spatial Filtering & 3_4 & Enhancement Using the Laplacian

做法說明

spatialFiltering 需要依照 mask 的大小，對每個點都必須去遍歷一次這個 mask，並且乘上在原本 matrix 裡對應的點。這裡要先算出 mask 的 size，再轉換成如何對應原圖，例如 mask size = 3，我會先把 $\text{int}(3/2) = 2$ ，這樣就知道 (k, l) 需要從 -1(1-2)~1(3-2)，然後再用 (i, j) 去跑一次加上這些的組合作為原圖的 index，也就是 (i-1, j-1), (i-1, j), (i-1, j+1), ……，而對應到 mask 的 index 則是 (k+mask size, l+mask size)。在這裡我使用的是 pad with zeros，也就是超出邊界的點直接忽略。其他用法我也有寫在程式的註解裡，ignore the boundary 只需照 mask 的大小縮小(i,j)的範圍，應該是最好寫的。mirroring 則是要加判斷式，雖然麻煩，但整體而言 mirroring 應是優於其他兩個，會比較相近於正確的結果。

而 laplacianFiltering 則是使用固定的 mask (二階導數的相似值)去帶入 spatialFiltering，得到 scaledLaplacian 這個 matrix，再將其乘上 scale 後加上原本的 matrix。

結果圖片

Mask1 = [0 1 0; 1 -4 1; 0 1 0]

Mask2 = [1 1 1; 1 -8 1; 1 1 1]

原圖

Mask1, scale=1



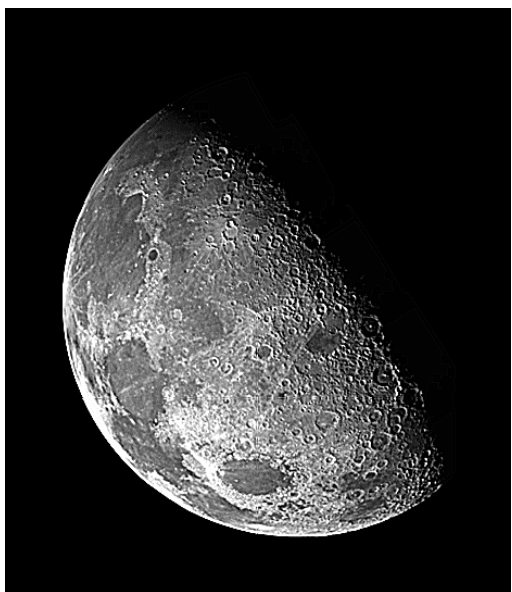
Mask1, scale=-1



Mask1, scale=-3



Mask1, scale=-5

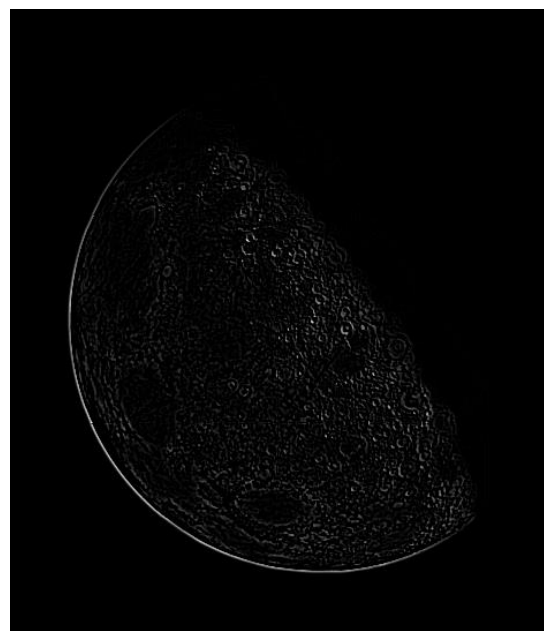


Mask2, scale=-5



Mask1 scaledLaplacian

Mask2 scaledLaplacian



分析以及討論

先比較 Mask1 和 Mask2，發現 Mask2 做出的差距會較明顯(值較大)，因此銳化的幅度會大於 Mask1。再來比較不同 scale 的差異， $scale > 0$ ，相當於把那些二階導數較大的區域再加亮，整張圖看起來就會比較模糊。而若是 $scale < 0$ ，則會加強對比，也就是銳化整張圖，扣掉的 scaledLaplacian 越多，銳化程度越高。

3_5 Unsharp Masking

做法說明

Unsharp Masking 的概念就是先把原圖模糊化，再用原圖減掉模糊化的圖得到 mask。因此這裡就用 average mask 將 input 模糊化，再依公式 $output = input - scale * mask$ 算出 output。需要注意的是 average mask 的形式會是 $1/n[1\ 1\ 1; \dots; 1\ 1\ 1; \dots; \dots]$ ，也就是先全部填 1，外面再除以總數，讓整個矩陣內部的值加起來是 1。

結果圖片

$$Mask1 = 1/9[1\ 1\ 1; 1\ 1\ 1; 1\ 1\ 1]$$

$$Mask2 = 1/25[1\ 1\ 1\ 1\ 1; 1\ 1\ 1\ 1\ 1; 1\ 1\ 1\ 1\ 1; 1\ 1\ 1\ 1\ 1; 1\ 1\ 1\ 1\ 1]$$

原圖



Mask1 blurred



Mask1 scale=1



unsharp mask1



Mask1 scale=-1



Mask1 scale=4.5



Mask2 scale=1



Mask2 scale=4.5



Mask2 blurred



unsharp mask2



[分析以及討論](#)

觀察圖可以先發現，使用 average mask 的 size 越大，越能將整張圖模糊化，因此得出的 gmask 效果也比較好。至於 scale 的效果則和 laplacian 相反，因為是加上能使圖銳化的 gmask， $scale > 0$ 會銳化，而 $scale < 0$ 則是模糊。