

# 數位影像處理

## Digital Image Processing

### 作業二 HW2

R11921038 江讀晉

2022/10/26

#### Overview

- Problem 1 編譯：`python p1.py`
- Problem 2 編譯：`python p2.py`
- Problem 3 編譯：`python p3.py`
- Problem 4 編譯：`python p4.py`
- 使用的 FFT (fast Fourier transform) 套件：`numpy.fft`

#### Problem 1. Sobel Kernel

##### a. The Fourier spectrum of the test image

使用 `numpy.fft.fft2` 計算出測試影像「keybord.tif」之離散或快速傅立葉轉換，再用 `numpy.fft.fftshift` 將頻率對齊中心，最後計算 magnitude 大小： $20\log(frequency)$ 。Spectrum 結果如 Fig.1 所示。

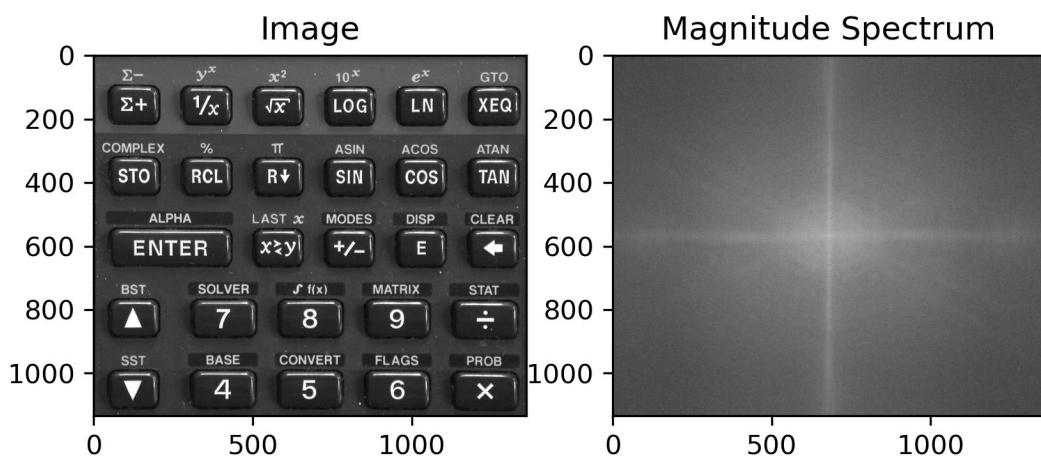


Fig.1 左側為 keybord.tif 原圖，右側為其對應光譜

##### b. Kernel with odd symmetry

根據課本敘述 *The Sobel mask exhibits odd symmetry, provided that it is embedded in an array of zeros of even size. To maintain this symmetry, we place  $h(x, y)$  so that its center is at the center of the  $602 \times 602$  padded array.* (as cited in Digital Image Processing, 4<sup>th</sup> ed., chapter 4)，因此實做方式為計算出 padded image 的中心座標，再將 kernel 依據計算結果放置於中心位置。

至於 transfer function  $H(u, v)$ 如何產生，則依照課本及課堂投影片講述之流程。然而，值得

注意的是，因為有 `np.fft.fftshift` 和 `np.fft.ifftshift`，因此乘以 $(-1)^{(x+y)}$ 之步驟可被取代，而 FFT 計算之結果無法保證原本 odd and real function 轉換至 odd, pure imaginary 的結果，因此需要手動調整數值，才能得到如課本 Fig. 4.38 之結果。如下方 Fig.2 所示。

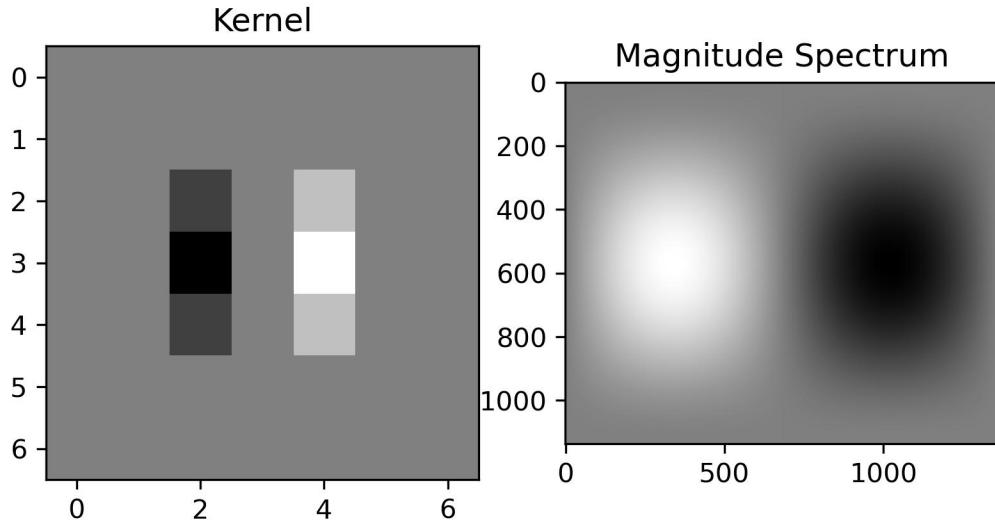


Fig.2 左側擷取 Sobel kernel 在 padded image 中心的鄰近區域，右側為 transfer function

#### c. The result of frequency-domain filtering

將 frequency-domain 的 keyboard 影像和 transfer function 相乘，並執行 `np.fft.ifft2`，得出 filtered 結果。圖像結果如 Fig.3 所示，和 Fig.1 相比，經過 vertical Sobel kernel filtering 的 spectrum 在垂直軸線上無明顯變化，即 spatial-domain 圖像結果水平邊界已被濾掉。

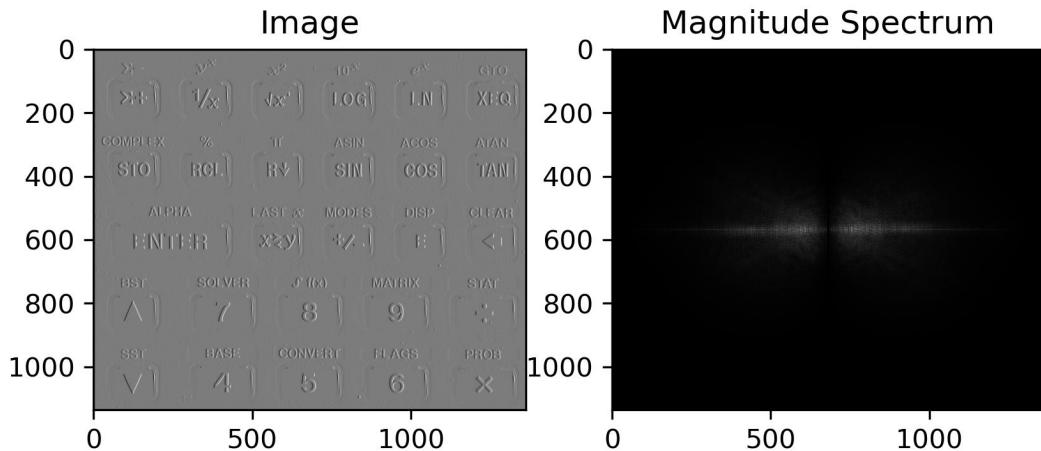


Fig.3 左側為 filtered 結果，右側為其 spectrum

#### d. The result of spatial-domain filtering

使用 opencv 的 Sobel function 實作 spatial-domain filtering。如 Fig.4 所示。

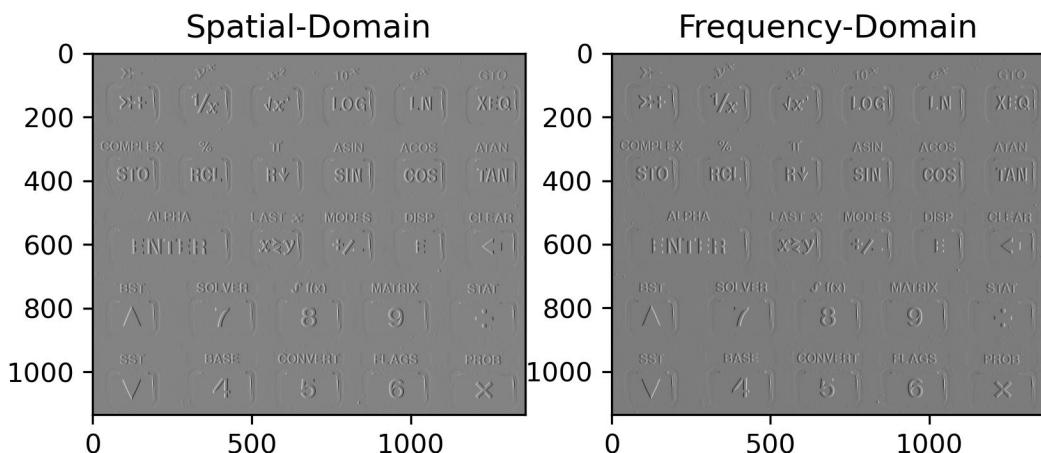


Fig.4 兩張圖為 filtered 結果，左側使用 spatial-domain，右側使用 frequency-domain

#### e. The result of frequency-domain filtering without odd-symmetric kernel

根據 1-c 的引述和課堂投影片內容，我認為沒有實作 odd symmetry 的 kernel 僅會造成 filtered 的結果不會對齊在圖像中心，如 Fig.5 所示。因此解決此問題的方法僅需在最後做一次 shifting 即可達到和 1-c 相同結果。

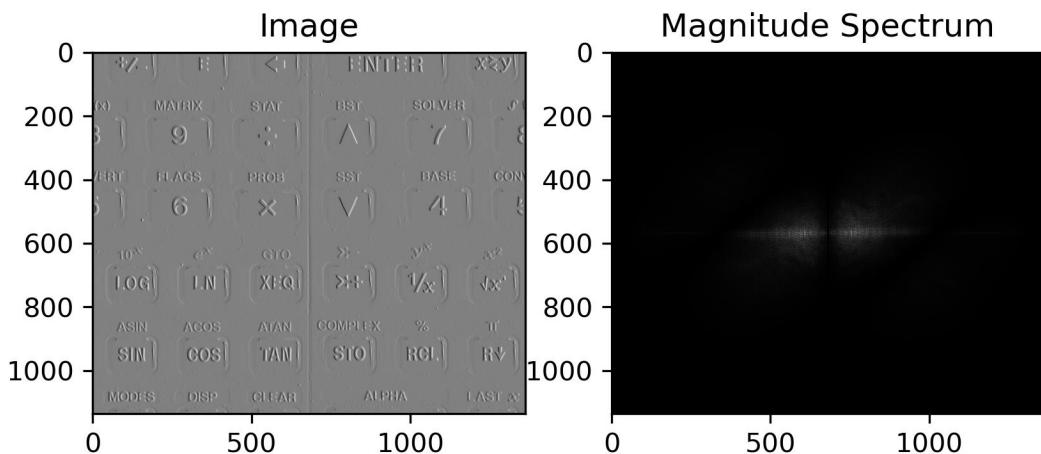


Fig.4 The result of frequency-domain filtering without odd-symmetric kernel

## Problem 2. Image Enhancement with Homomorphic Filtering

### 1. Homomorphic filtering

本題的測試圖片為「Einstein.tif」和「phobos.tif」，如 Fig.5 所示。

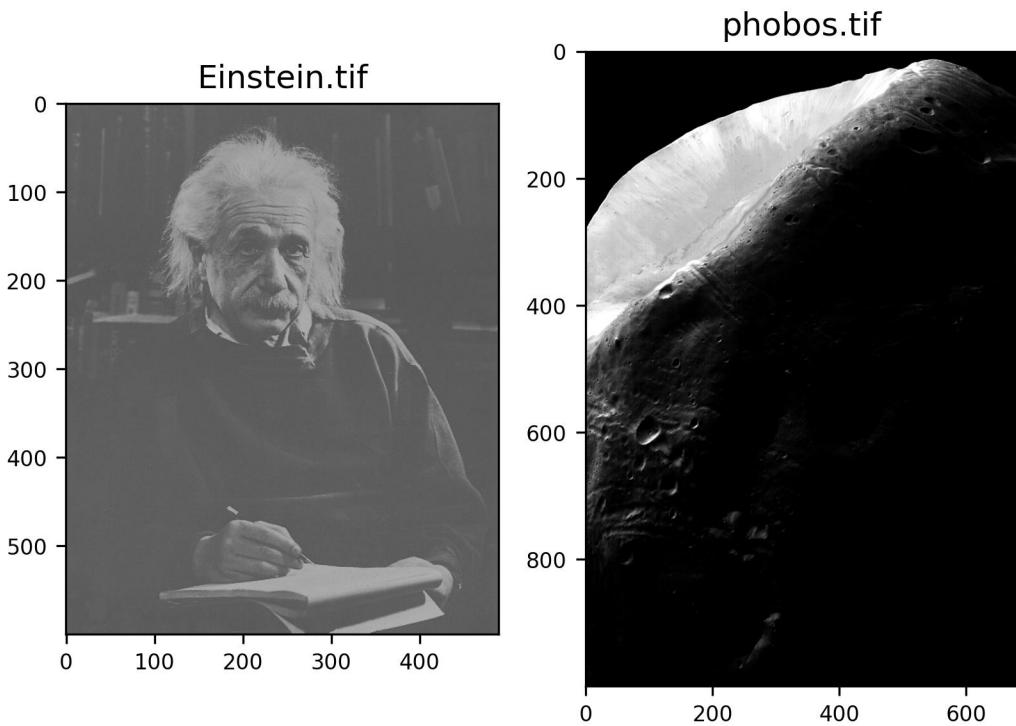


Fig.5 原始圖片，左側為 Einstein，右側為 phobos

由 Fig.5 可以看出，Einstein 整張圖較為灰濛，contrast 不高；phobos 的右下角處幾乎看不見細節，僅看出黑暗一片。根據課堂投影片，homomorphic filter 可以達到 contrast enhancement 並且 sharpening of reflectance components，因此決定以 homomorphic filter 實踐 image enhancement。

homomorphic filter 的原理為，影像 illumination 的部分為 slow spatial variation，因此有 low frequency components；reflectance 則有 high frequency components 因為其變化較劇烈。使用 homomorphic filter 則可將其兩部分分開作用，以達到較好的影像結果。



Fig.6 Homomorphic filtering 流程

程式碼的實踐則基本上按照上圖 Fig.6 流程，公式則參照下方

$$H(u, v) = (\gamma_H - \gamma_L) \left[ 1 - \exp \left\{ -c \left( \frac{D(u, v)}{D_0} \right)^2 \right\} \right] + \gamma_L$$

其中四個參數  $\gamma_H, \gamma_L, c, D_0$ ，則使用 trial and error 的方式調整。

## 2. Image enhancement

實作的結果如 Fig.7、Fig.8 所示。

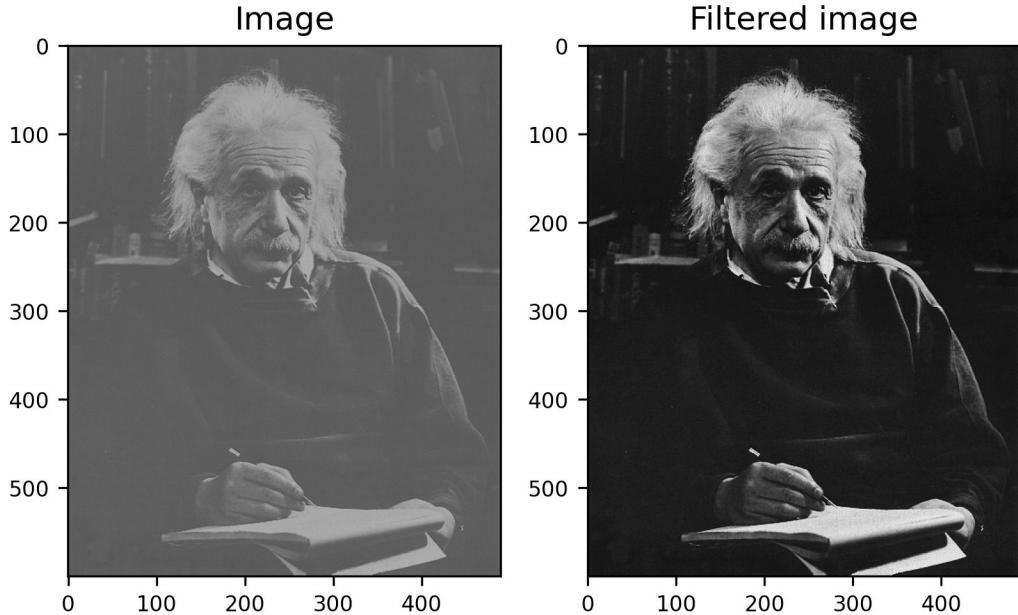


Fig.7 Einstein 前後對比圖

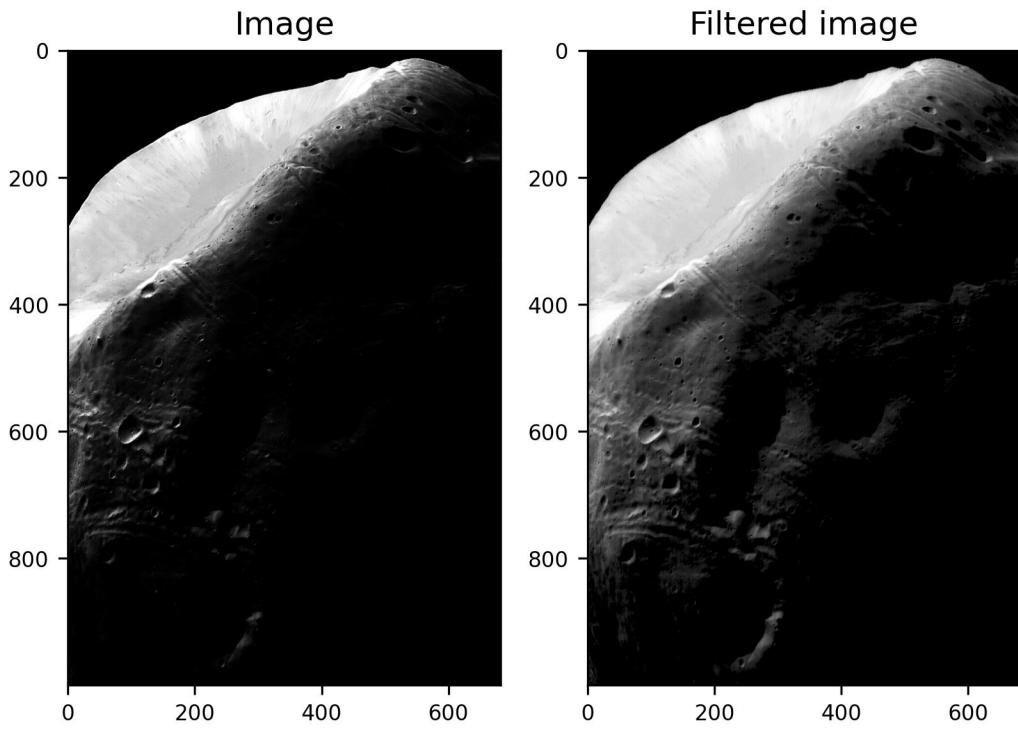


Fig.8 phobos 前後對比圖

在 Einstein 的部分，可以發現對比度提高，且背景和書籍變得較為清晰明顯；在 phobos 的部分，我使用兩個結果做 superposition，其一為凸顯右下角，另一為銳化原本光亮處。雖然其疊加結果仍使得部分細節消失，但整體圖片的左上和右下的內容較原圖平衡。

### Problem 3. Optimum Notch Filtering

本題使用 optimum notch filtering，先計算 notch pass 得到 noise，再以最小化標準差的方式求出 weight，以得到處理過後的影像。

#### 1. Notch filter

Notch reject filter 的目的是將影像上的 noise 移除，求出較為清晰的影像；notch pass filter 則是求出 noise。根據課堂投影片內容，我實作 Butterworth notch reject filter of order n，參考公式如下：

$$H_{NR}(u, v) = \prod_{k=1}^Q H_k(u, v) H_{-k}(u, v) = \prod_{k=1}^Q \frac{1}{1 + \left[ \frac{D_{0k}}{D_k(u, v)} \right]^{2n}} \frac{1}{1 + \left[ \frac{D_{0k}}{D_{-k}(u, v)} \right]^{2n}}$$

where  $D_k(u, v) = \left[ \left( u - \frac{M}{2} - u_k \right)^2 + \left( v - \frac{N}{2} - v_k \right)^2 \right]^{0.5}$

and  $D_{-k}(u, v) = \left[ \left( u - \frac{M}{2} + u_k \right)^2 + \left( v - \frac{N}{2} + v_k \right)^2 \right]^{0.5}$

首先，我先將「Martian terrain.tif」的 spatial image 和 FFT 後的 magnitude spectrum 繪製出來，如 Fig.9 所示。可以發現，其 spectrum 呈現出數個光點，以及一些星雲狀的白光處，因此本張圖片的 noise 非 single-frequency burst。

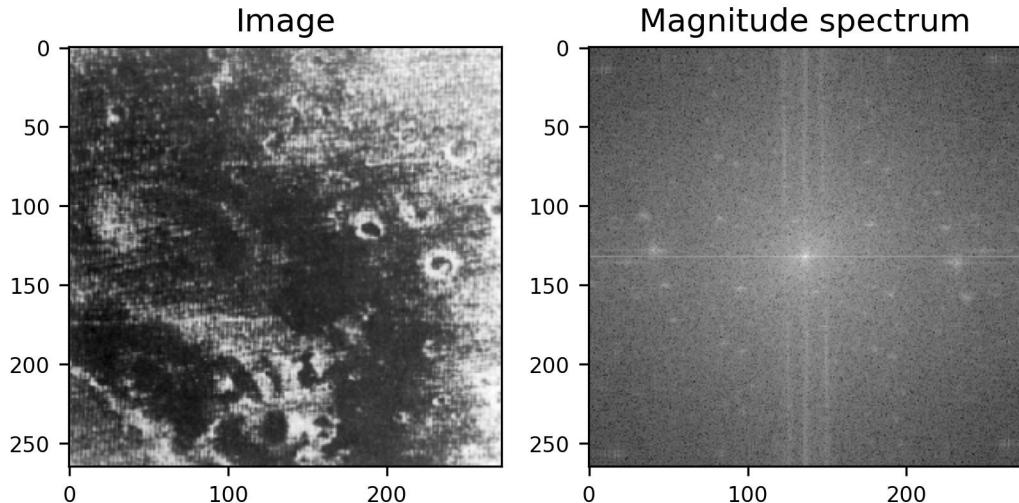


Fig.9 Martian terrain 的 image 和 spectrum

為了求出 noise，我將我主觀認為相對佔比重的 7 個光點座標  $(u, v)$  標示出來，得到一 notch reject filter，如 Fig.10 所示。之後再計算 notch pass filter，即  $H_{NP} = 1 - H_{NR}$ ，計算出 noise，如 Fig.11 所示。

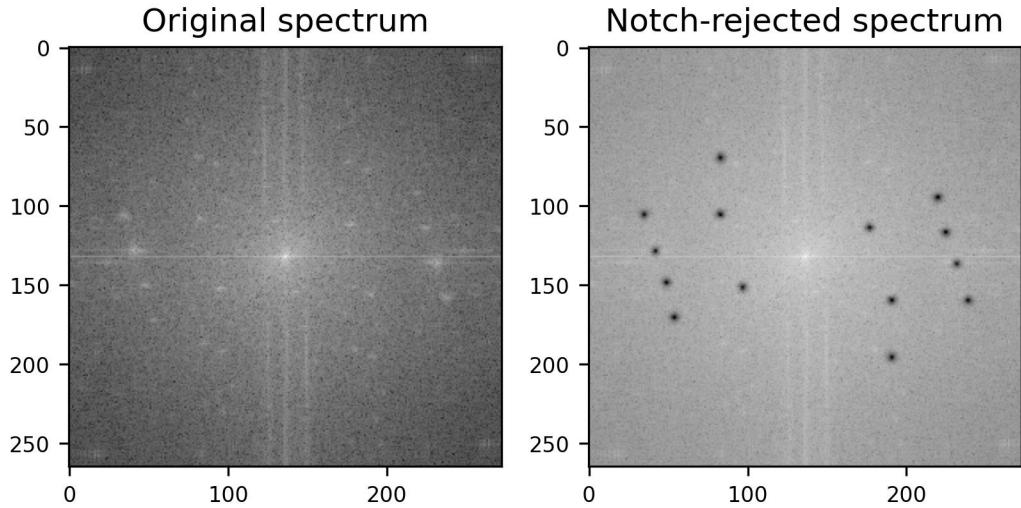


Fig.10 左側為原本 spectrum，右側為 Fourier transform multiplied by a notch reject filter

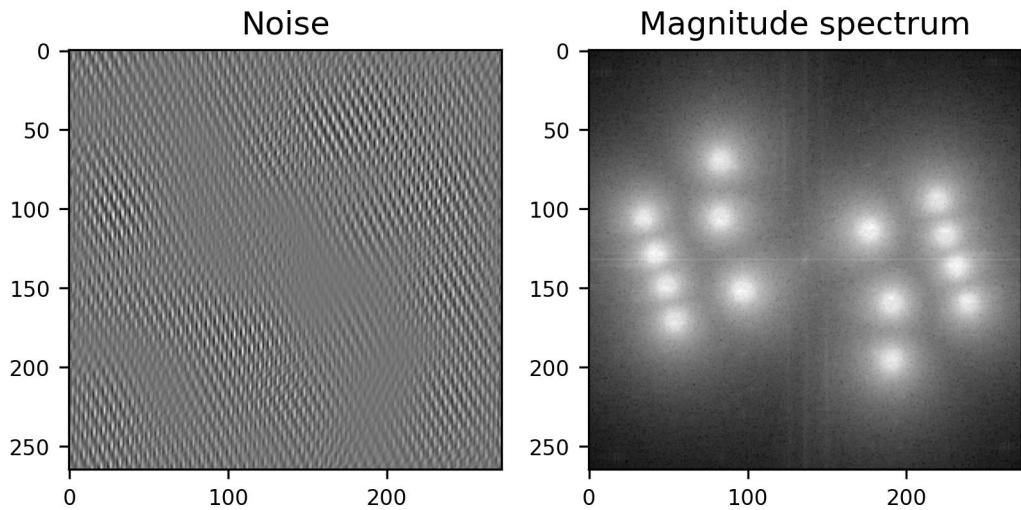


Fig.11 左側為 noise，右側為 Fourier transform multiplied by a notch pass filter

到此步驟，interference patterns 已經提取出來了，然而，如果直接實作 notch reject，可能會將過多影像資訊移除，因此接著實作 optimum notch filtering

## 2. Optimum notch filtering

Optimum notch filtering 的作法為將原本毀損的影像減去加權過的 interference pattern，以還原出原本的影像資訊。其實作方法為，選定一鄰近區域  $S_{xy}$ ，此區域之 local variance 越小，其影像資訊表現越不 noisy，因此還原之影像品質越高。參考課本公式算出 weight， $g$  為鄰近區域  $S_{xy}$  中的毀損影像， $\eta$  為鄰近區域  $S_{xy}$  中的 noise：

$$w(x, y) = \frac{\bar{g}(x, y)\eta(x, y) - \bar{g}(x, y)\bar{\eta}(x, y)}{\bar{\eta}^2(x, y) - \bar{\eta}^2(x, y)}$$

求出 weight 後，再算出還原的影像：

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y)$$

### 3. 結果和比較

Optimum notch filtering 的超參數為鄰近區域  $S_{xy}$  size 的選擇。我將 size 設為 13、43 和 73，比較其結果，可以發現 size 相對小的話，其計算結果會出現雜訊，如 Fig.12 所示。因此就我的寫法，選擇 size 設置較大較為合適。

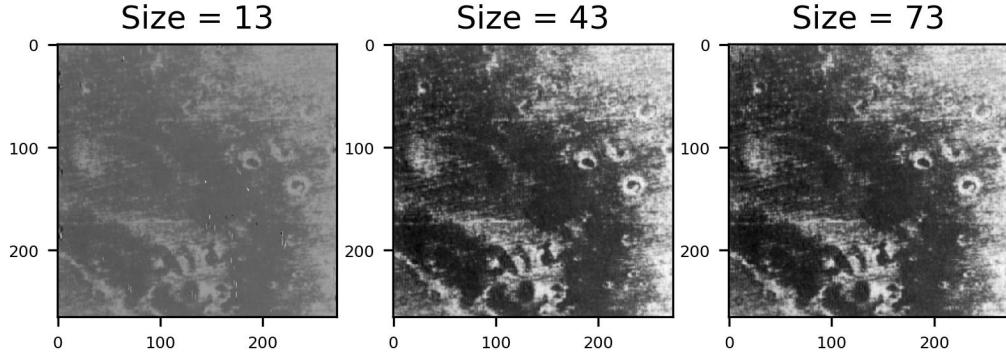


Fig.12  $S_{xy}$  size 大小比較

此外，我也比較 optimum notch filtering 和純用 notch rejected filter 的結果。事實上，單純肉眼看，兩者表現相近，將原影像的顆粒感部分移除，而在影像較白處明顯地 noise 已被消除，如 Fig.13 所示。推測可能原因為，我在 frequency domain 上將大致的 interference pattern 標記出，因此 notch rejected filter 可以接近 optimum notch filtering。另外，本題的 hyperparameter 總數較多，因此如果調整至適當參數，或許可以得到更為理想的 filtering 結果。

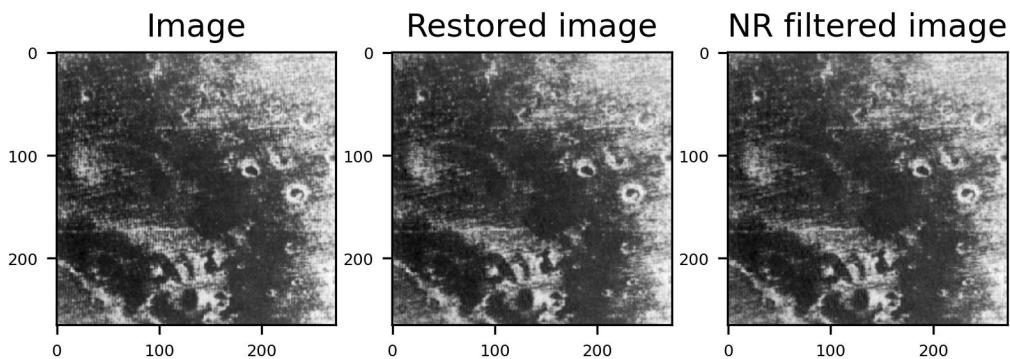


Fig.13 左側為原圖，中間執行 optimum notch filtering，右側則是經過 notch rejected filter

### Problem 4. Wiener Filtering

本題使用 Wiener filtering，計算方法參考投影片。frequency domain 計算公式如下， $\hat{F}$  為勁過濾波的影像， $H$  為 degradation function， $G$  為毀損影像， $K$  為超參數，代表 SNR(signal noise ratio) 之倒數：

$$\hat{F}(u, v) = \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + K} \right] G(u, v)$$

### 1. Determine the best Wiener filter

Wiener filter 計算方式如上，因此需決定的為 degradation function。在「degraded photographer」中，看起來 motion blur 較為不嚴重，gaussian noise 則為主要雜訊。在實際畫出 spectrum，如 Fig.14 所示，也可以發現並沒有明顯貫穿全圖的 interference pattern，因此 degradation function 採用 gaussian kernel，其公式如下：

$$H(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

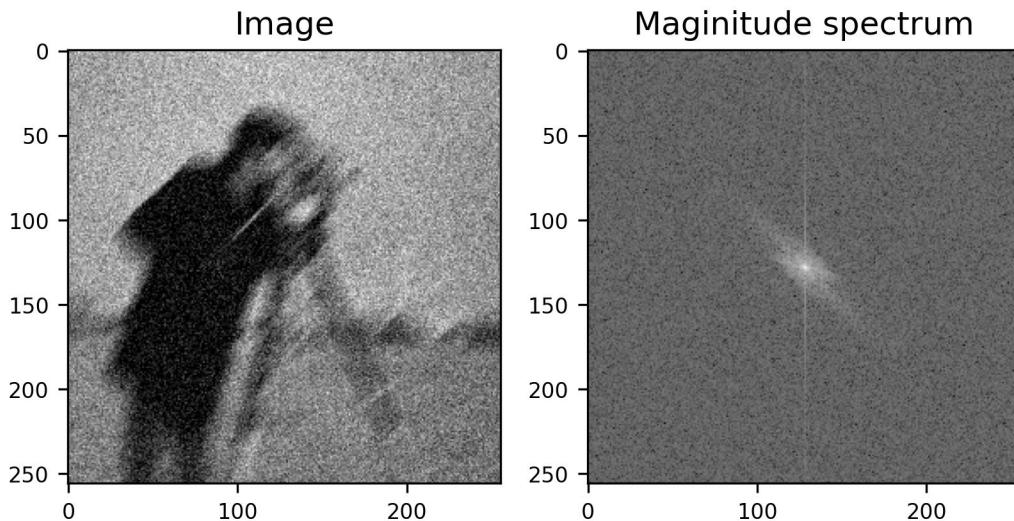


Fig.14 Degraded photographer 的 image 和 spectrum

在「degraded football player」中，看起來 motion blur 較為嚴重， noise 不大明顯。在實際畫出 spectrum，如 Fig.15 所示。degradation function 如下，其中 T、a、b 為超參數：

$$H(x, y) = Tsinc[\pi(ua + vb)]e^{-j\pi(ua+vb)}$$

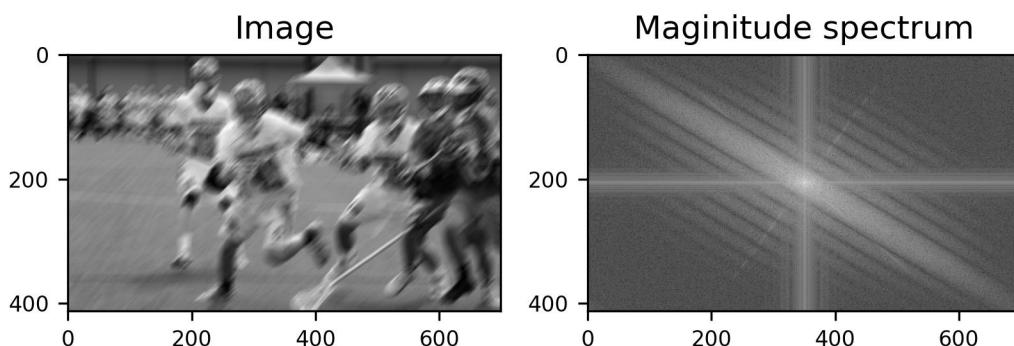


Fig.15 Degraded football player 的 image 和 spectrum

## 2. Show the restored image of each Wiener filter

在「degraded photographer」的實作中，我使用 `scipy.signal.gaussian` 產生 gaussian kernel，並選擇適當 kernel size。結果如 Fig.16 所示，可以發現雖然畫面仍然有些模糊，但將原本密集的黑點雜訊去除許多。

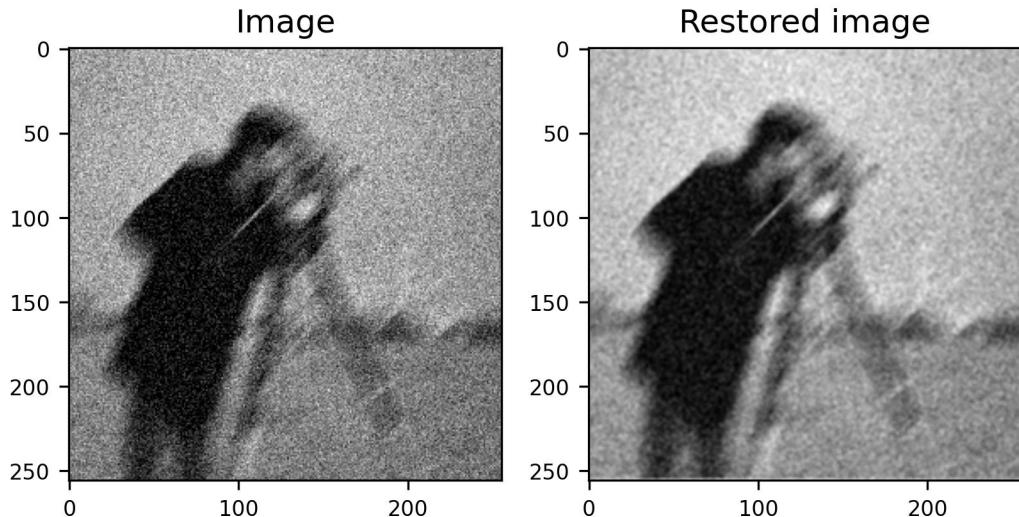


Fig.16 Degraded photographer 的原圖和重建的影像

在「degraded football player」實作中，我嘗試不同的  $T$ 、 $a$ 、 $b$ ，最後得出的影像如 Fig.17 所示。雖然仍然 blurred，但比起原圖，物體移動的軌跡較為削減。

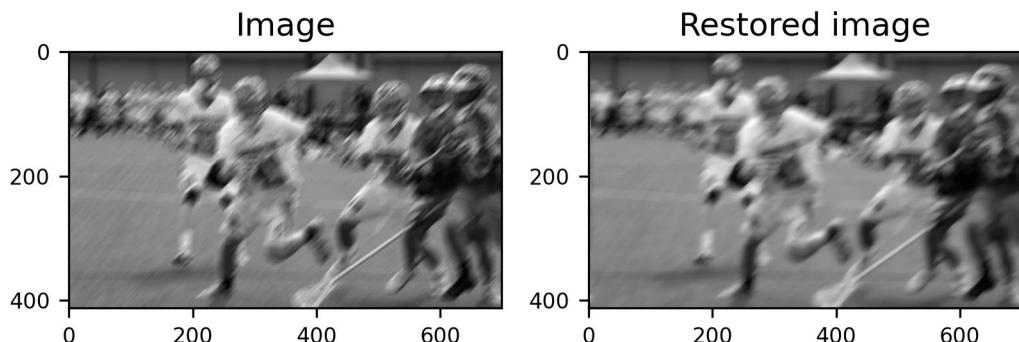


Fig.17 Degraded football player 的原圖和重建的影像