

HOMEWORK 4 - REPORT

Method

這次作業應用了 Notch Reject Filter，具體是在 Image1 使用 Ideal，在 Image2 使用 Butterworth。

With:

Radius D_0 , center frequency (u_0, v_0)

$$D_1(u, v) = \sqrt{\left[\left(u - \frac{M}{2} - u_0\right)^2 + \left(v - \frac{N}{2} - v_0\right)^2\right]}$$

$$D_2(u, v) = \sqrt{\left[\left(u - \frac{M}{2} + u_0\right)^2 + \left(v - \frac{N}{2} + v_0\right)^2\right]}$$

Ideal notch reject filter transfer function:

$$H(u, v) = \begin{cases} 0, & \text{if } D_1(u, v) \leq D_0 \text{ or } D_2(u, v) \leq D_0 \\ 1, & \text{otherwise} \end{cases}$$

Butterworth notch reject filter transfer function of order n:

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u, v)D_2(u, v)}\right]^n}$$

Step 1: Get the Spectrum

對 image 使用 `np.fft.fft2()`，後使用 `np.fft.fftshift()` 去轉換得到 `fshift`，使用 `spectrum = c*log(1+r)`，`c = (L-1)/log(1+max_value)`，`r` 是 `fshift` 的 `abs` (取實部來計算)，去找 `spectrum`。

Step 2: Apply filter on Spectrum

在 Image 尋找亮點發現，在 Image1 是中間的線，在 Image2 是分散。

Image1 直接對中間的線使用 Ideal notch reject filter，選擇 radius 為 10 去找 notch filter。

Image2 使用 for-loop 去尋找亮的點，從中點到兩點計算距離得到 (u_0, v_0) ，使用 Butterworth notch reject filter，選擇 radius 為 30，order 為 2 去找 notch filter。拿 spectrum 乘給 notch filter 得到 filtered spectrum。

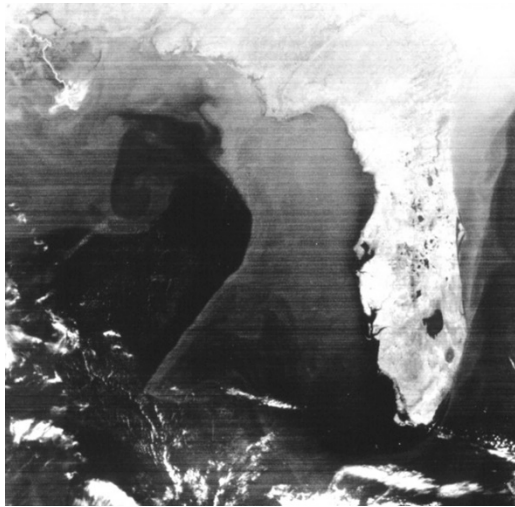
Step 3: Convert the new Spectrum to spatial domain

對 `fshift` 配上找到的 notch filter 使用 `np.fft.ifftshift()`，後使用 `np.fft.ifft2()`，最後使用 `abs` 取實部得到 final image (已經去除 noise)

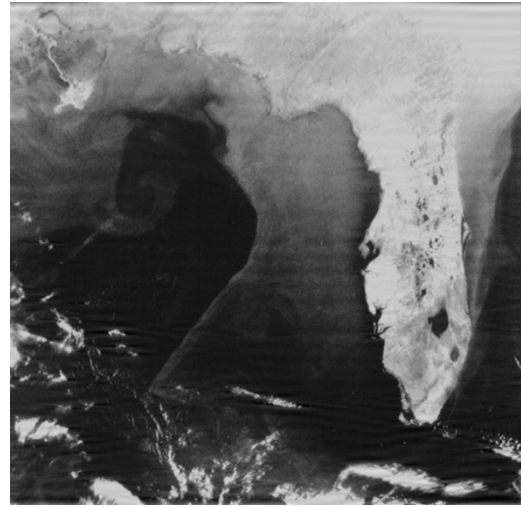
這次使用 `plt` 來顯示 step1 到 step3 的圖。

Result

Image 1

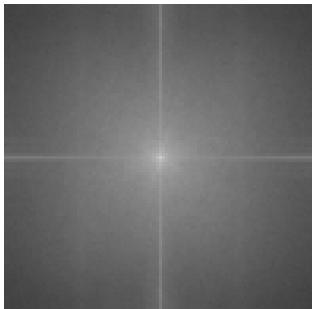


Original

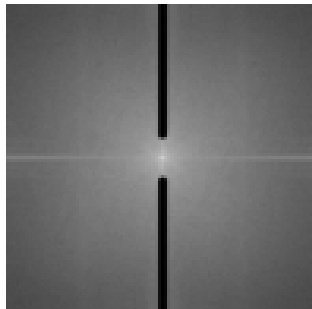


Final

Step 1



Step 2



Step 3

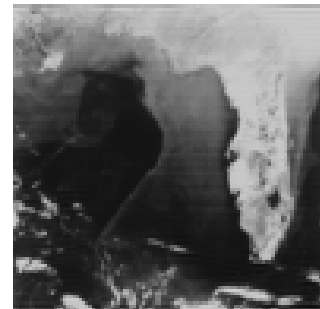


Image 2

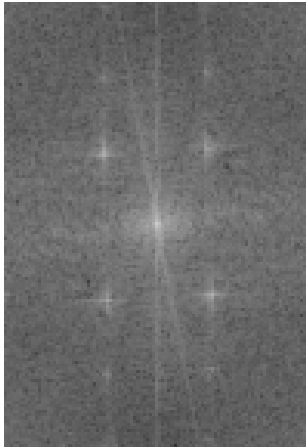


Original

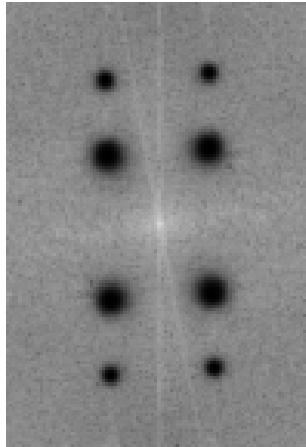


Final

Step 1



Step 2



Step 3



Feedback

實作作業過程對兩張圖使用 3 種 notch reject filter 方法 (Ideal, Butterworth, Gaussian)，通過多次測試決定對 Image 1 選擇 Ideal notch reject filter，Image 2 選擇 Butterworth notch reject filter。

Radius 的大小會影響 final image 的亮度和準確度，模糊感。

在 Butterworth notch reject filter 可以使用 n 來控制點的大小，可以劃出比較適合兩點的黑點。

計算過程使用到 Fourier Transform 倒出 complex 值，所以需要使用 abs 來取得實部進行計算。

這次作業應用很多公式來處理。