

Digital Image Processing HW3

M093781 趙宇涵

I. 問題簡介：

本次作業需使用到 image restoration 的方式，把 degrading 後模糊、受到破壞的照片盡量還原，並使用 PSNR 作為判斷還原優劣的依據。

II. Restoration 方法介紹：

本次使用到三種還原的方法，分別為：Wiener filter, Constrained Least Squares restoration, 以及 Blind deconvolution. 以下將分別說明。

1. Wiener filter:

由於在 image restoration 的過程中，如何處理雜訊會是很重要的課題，若直接還原圖片而不處理雜訊，會受到高頻雜訊的影響，產生都是雜訊的影像，因此 Wiener filter 考慮了雜訊的因素，透過設定適當的 K 值，可以自動壓抑高頻，成功抑制雜訊帶來的干擾。

Wiener filter 的計算公式如下：

$$\begin{aligned}\hat{F}(u, v) &= \left[\frac{H^*(u, v)S_f(u, v)}{|S_f(u, v)|^2 + S_\eta(u, v)} \right] G(u, v) \\ &= \left[\frac{H^*(u, v)}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)} \right] G(u, v) \\ &= \left[\frac{1}{|H(u, v)|} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)} \right] G(u, v) \\ \hat{F}(u, v) &= \left[\frac{1}{|H(u, v)|} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)\end{aligned}$$

2. Constrained Least Squares restoration (CLS)

這個方法接續 Wiener filter，但更進一步對重建後的影像限制，把重建後的範圍控制在 $\|g - H\hat{f}\|^2 = \|\eta\|^2$ ，並且在算式中加入 Laplacian kernel，使重建後的影像更平滑，避免高頻的部分。計算公式如下：

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

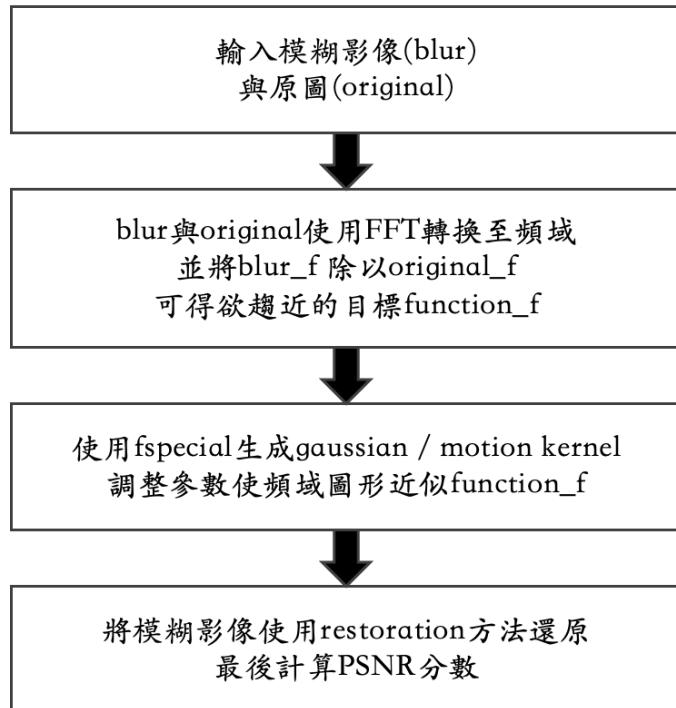
而此處 $P(u, v)$ 為 Laplacian kernel 的傅立葉轉換。 γ 則根據上述的限制調整至適當的數值。

3. Blind deconvolution

此方法則是利用回授的方式，自動找到最適合的 restoration function，此處使用的是 MATLAB 的套件來計算。

III. 影像重建流程

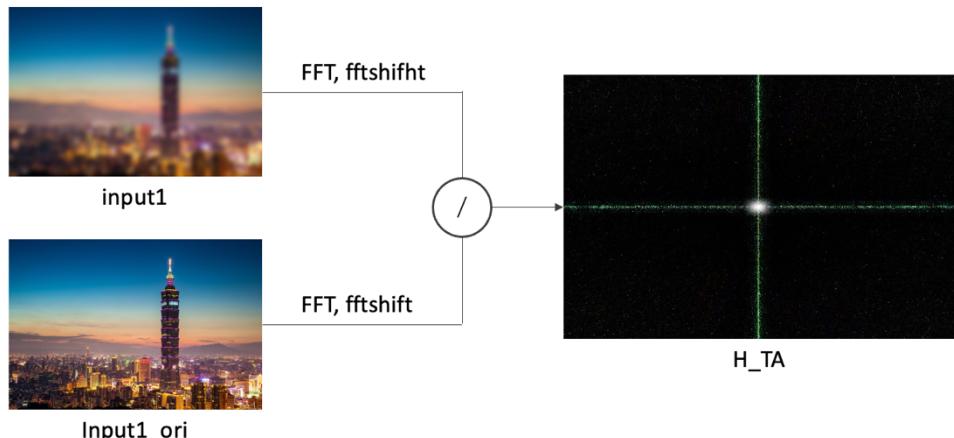
在影像重建的過程中，最重要的是估計出把原圖模糊的 function，在與助教討論時，助教說明可以使用原圖與模糊的圖片進行比較，較能夠猜出此 function，而若要使用兩張圖進行比較，較方便的方法是把影像都轉換到頻域，再將模糊後的圖片除去原圖，即可得到頻域中的 function，接下來就能調整 kernel 使其接近題目上的 function。本次作業使用這樣的架構進行，流程圖如下所示：

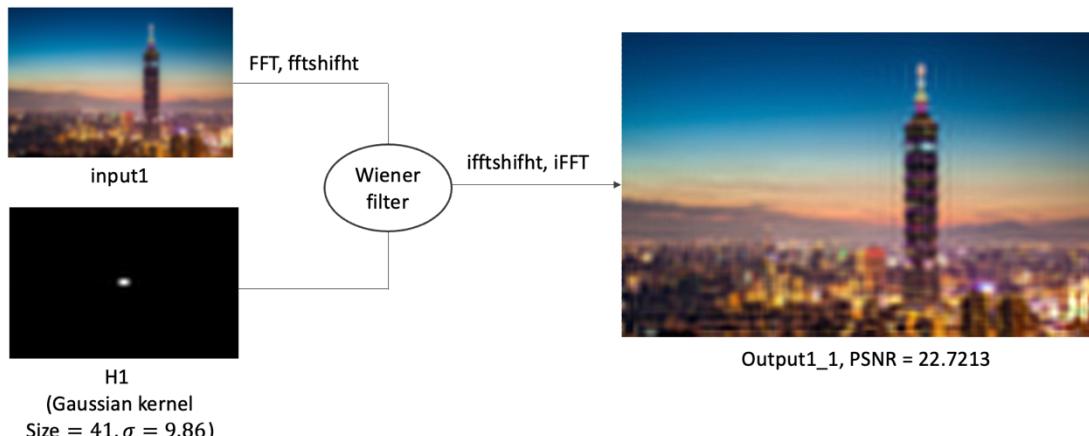


IV. 各影像分析與討論

此段落將針對各個影像所需要的處理方式進行分析，並展示處理後的結果。

1. Input 1



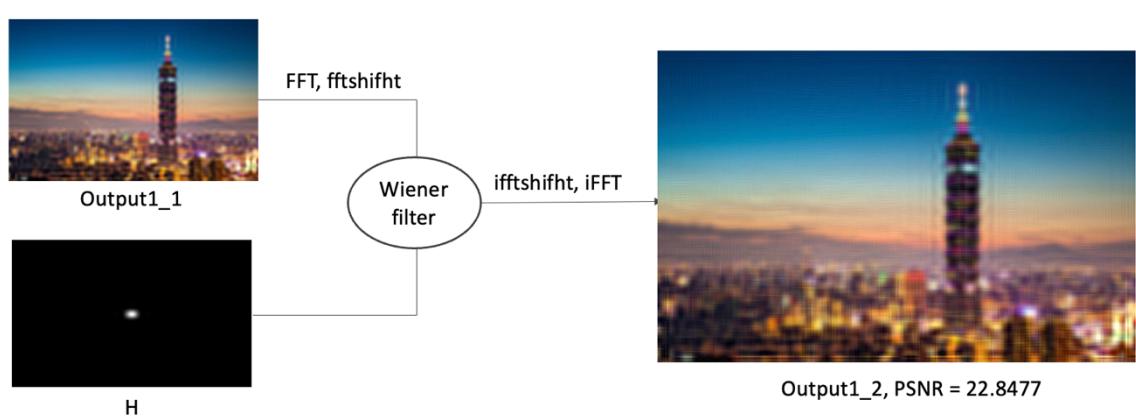


首先說明 input1，根據助教的提示以及兩張圖在頻域相除後的結果，知道這個 kernel 是高斯 kernel，並在多次嘗試後，找到能夠提高 PSNR 分數最高的是 kernel size 41, σ 9.86，可以將原本的 21.6306 提升至 22.7213。

但是可以看出輸出的影像輪廓較為清楚，但是細節依舊模糊，因此推測可能還有另外一個高斯 kernel，經過相同的方法，找到第二個高斯 kernel 為 kernel size 25, σ 2.17，可以將 PSNR 些微提升至 22.8477。

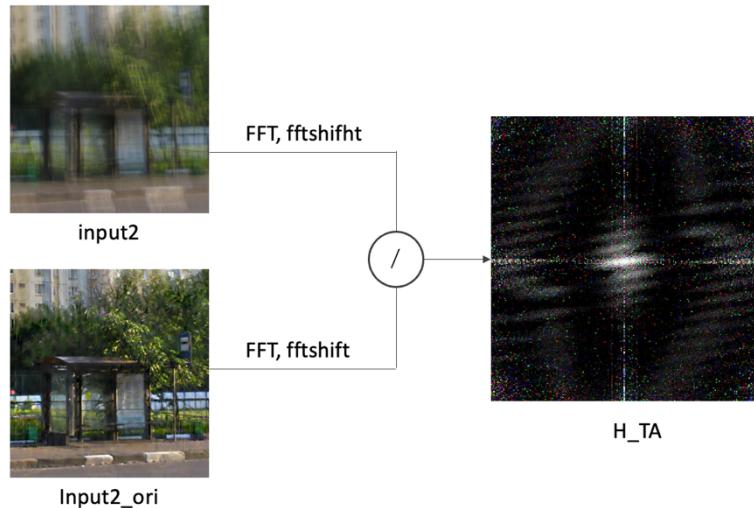
$$H = H1 \times H2$$

H1
 (Gaussian kernel
 Size = 41, σ = 9.86)
 X
 H2
 (Gaussian kernel
 Size = 25, σ = 2.17)
 =
 H

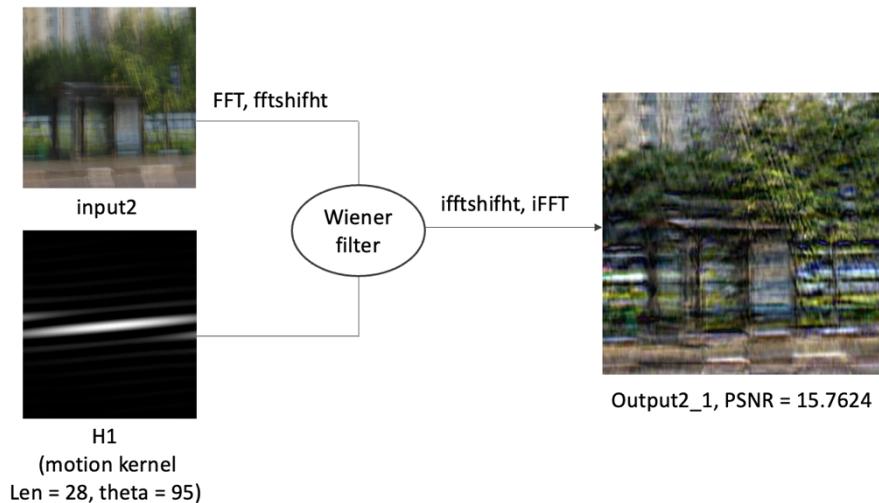
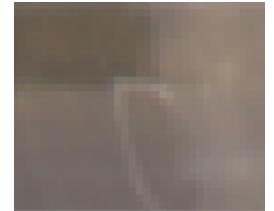


雖然最後 PSNR 分數有些微提升，但依舊沒有達到復原細節的部分，反而還多出許多細部線條。

2. Input 2



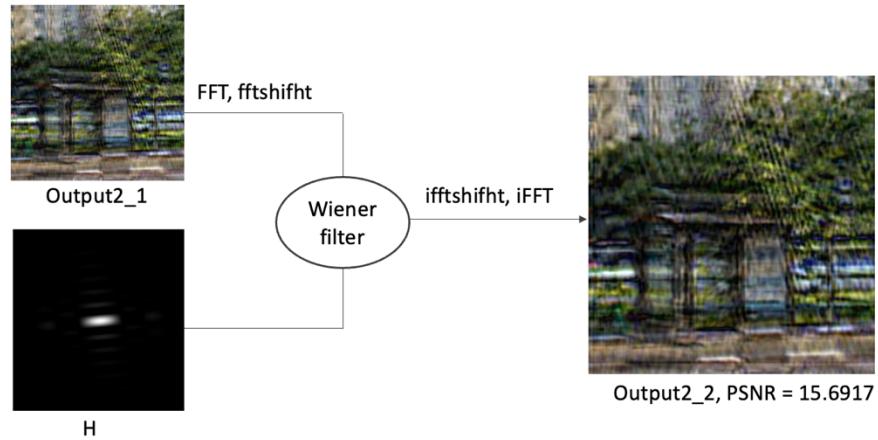
接下來第二張影像，根據助教的提示以及對圖片的觀察，可以得知此為 motion kernel，然而從一些小細節（例如左下角的白點，如右圖所示）可以得知此圖並非單一的 motion，而是先向左再向下的運動軌跡，因此我的想法是依序處理兩個 kernel，首先是處理向下的軌跡。



接下來堆疊向左的軌跡，length = 8, theta = 10。

$$H1 \text{ (motion kernel Length = 28, theta = 95)} \times X \text{ (motion kernel Length = 8, theta = 10)} = H$$

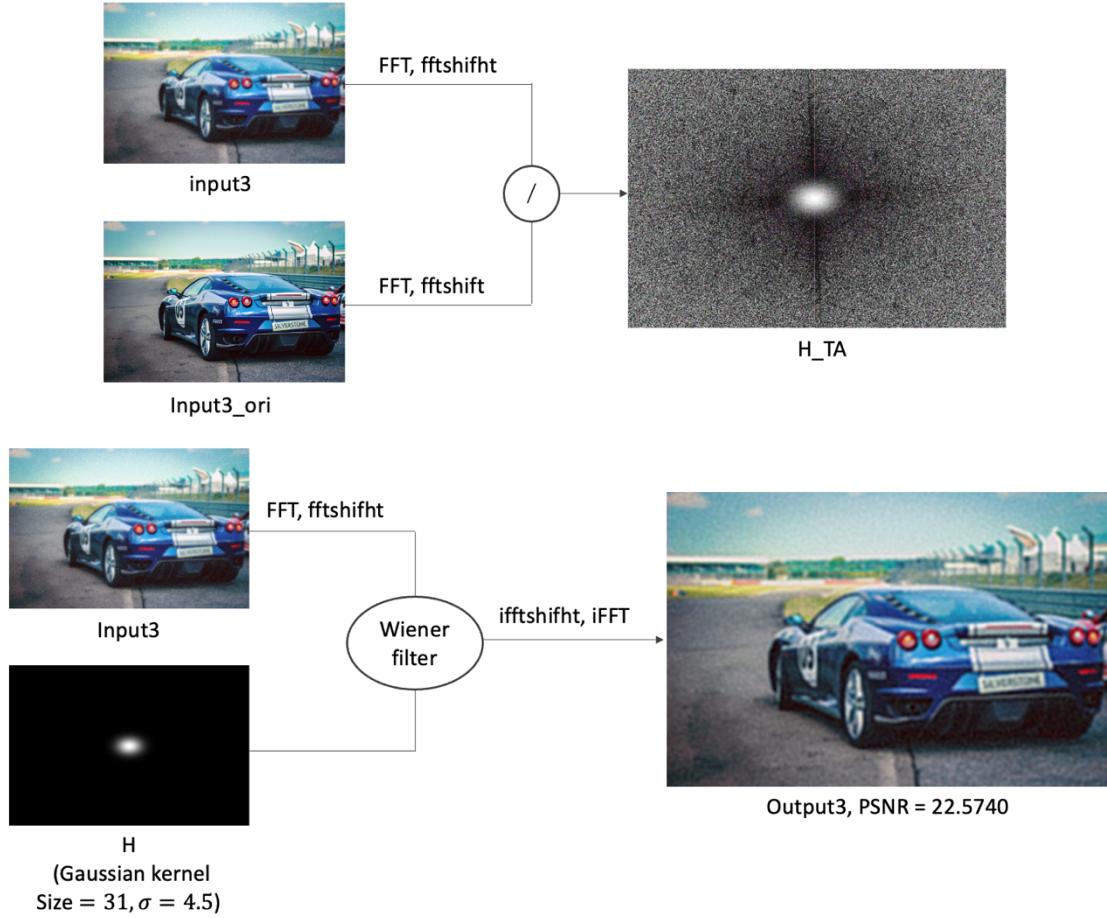
The diagram illustrates the convolution operation. It shows three images: H1 (motion kernel, length = 28, theta = 95), X (motion kernel, length = 8, theta = 10), and H (the resulting image). The multiplication symbol between H1 and X is preceded by an equals sign, indicating the result of the convolution.



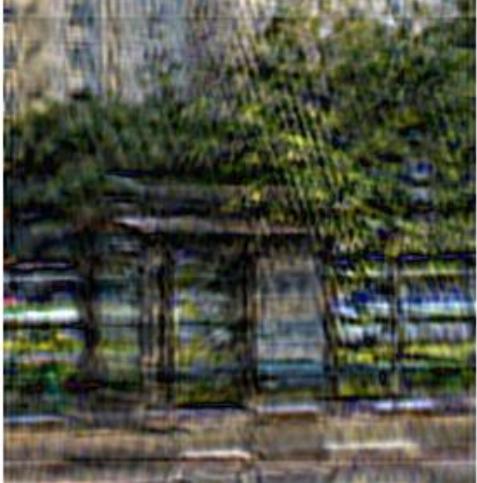
最後，我試了幾種方法處理 input2 都沒辦法提升他的 PSNR，最後輸出的圖片看起來已經沒有移動的軌跡，但多了很多不要的紋路，推測可能是因為這種圖片不是用兩個 kernel 堆疊起來，而是一個非線性的 kernel，有了這種想法後，我也試過 CLS 以及 blind deconvolution，但這兩種都沒辦法很有效的改善。

3. Input 3

Input 3 則是高斯模糊後加上雜訊的影像，我使用相同的方法分析，得到的 kernel function 為 size 31 $\sigma = 4.5$ 的高斯 kernel。



V. 實驗結果

Input	Parameters	Output	PSNR
1	<p>H1: Gaussian Size 41 Sigma 9.86</p> <p>H2: Gaussian Size 25 Sigma 2.17</p> <p>Restoration method: Wiener filter K = 0.001</p>		22.85
2	<p>H1: Motion Len 28 Theta 95</p> <p>H2: Motion Len 8 Theta 10</p> <p>Restoration method: Wiener filter K = 0.01</p>		15.69
3	<p>H1: Gaussian Size 31 Sigma 4.5</p> <p>Restoration method: Wiener filter K = 0.1</p>		22.57