Report

0310515 葉尚昀

Gaussian blur:

方法一：

此照片為 gaussian blur 的照片，於是我們就利用 gaussian 的 fft 去與 G(u,v) 相除，期待找到最 fit 的 variance 跟 kernel 大小來還原圖片。

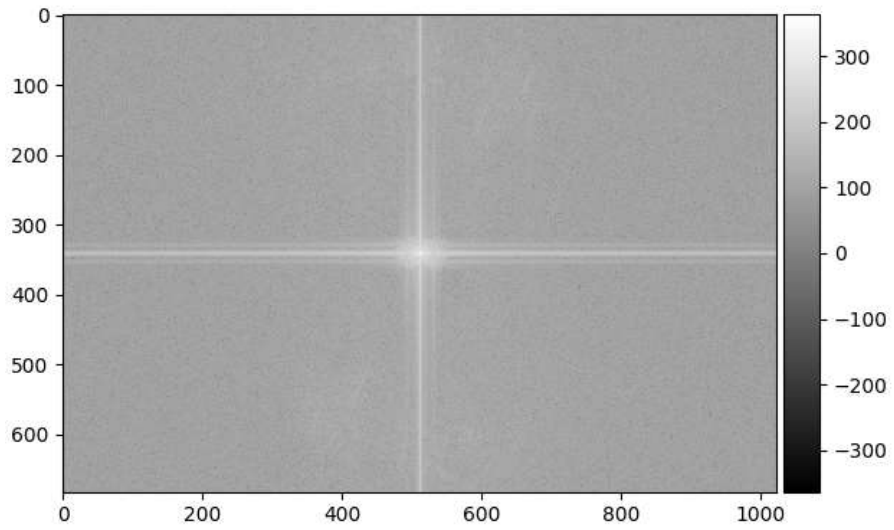

Fig 1. G(u,v) 可以看見四腳正方形的區域幾乎都被 mute 掉了(高頻的區域)

於是我們期待找到一個最 fit 的 Gaussian kernel 來相除，但在實作中發現，若 variance 太大在 fft 後就會變得比較尖銳，也就是高頻區域接近 0，inverse 後就會接近∞，讓整張圖片直接變成白色，於是只能找到一個極小的 variance 讓 fft 後的圖形較為平滑。

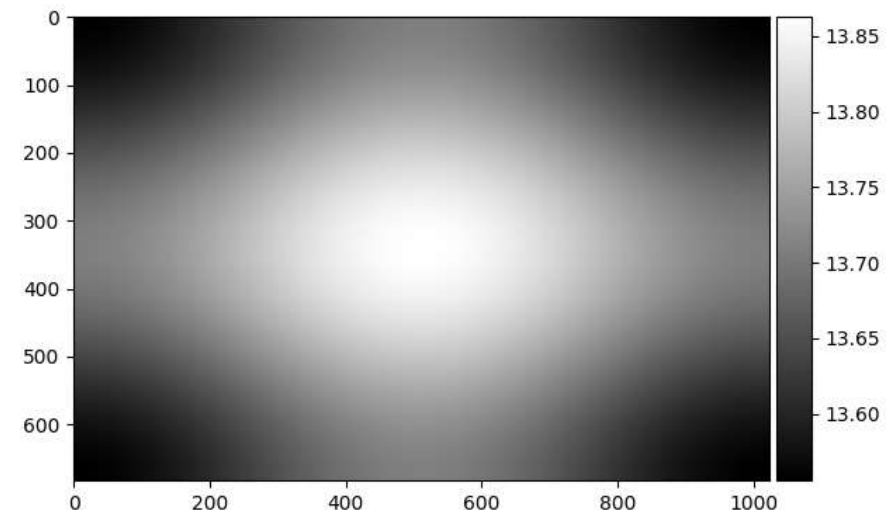

Fig 2. H(u,v), 因 variance 約為 0.3 kernel 為 17*17，但因 vairance 很小於是 FFT 後的 variance 就很大。

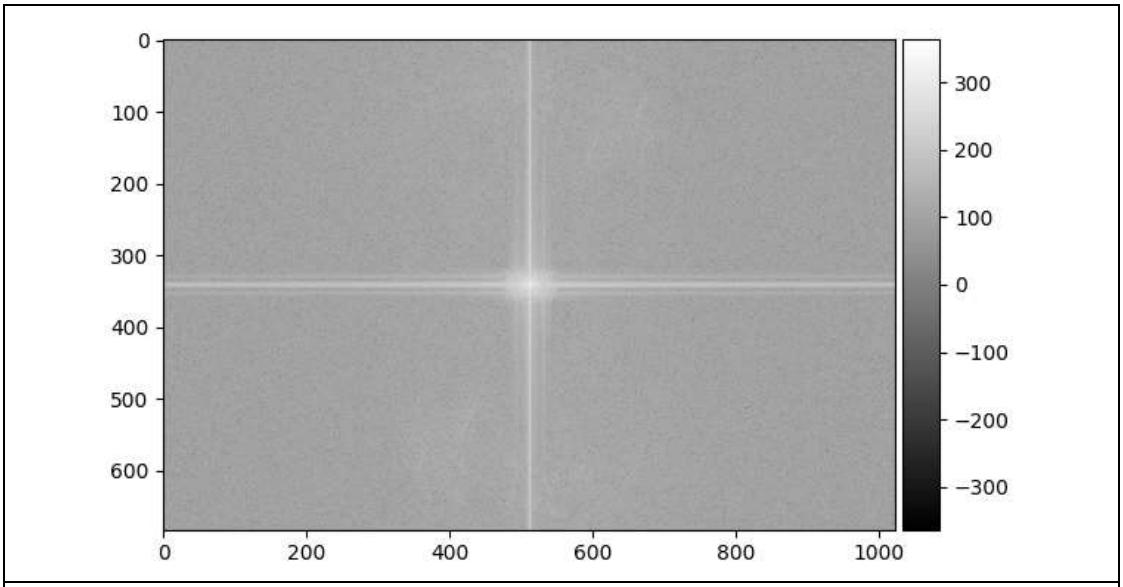

Fig 3.做出來的還原圖，基本上與 input 沒什麼差別

此方法做出來原本的 PSNR 為 150.342369789

還原後的 PSNR 為 150.342678586

處理過的圖片進步了 0.000205396%

方法二：

看 input 頻譜圖的感覺便是，若使用的的確是 gaussian filter，那在高頻被濾的那麼乾淨的情況看來，此 filter 的 variance 應該是滿大的才對，但如何解決 inverse 後值接近無限的問題呢？照常理來說，因為在 blur 的時候是除上了接近 0 的值，之後再除上接近 0 的數應該要相消才對，但因數位的儲存精準度有限，才會使還原結果不如預期，於是我設計了一個 clip，inverse filter 的直限制在 1-2 之間，超越 2 的就設為 2，其結果為：

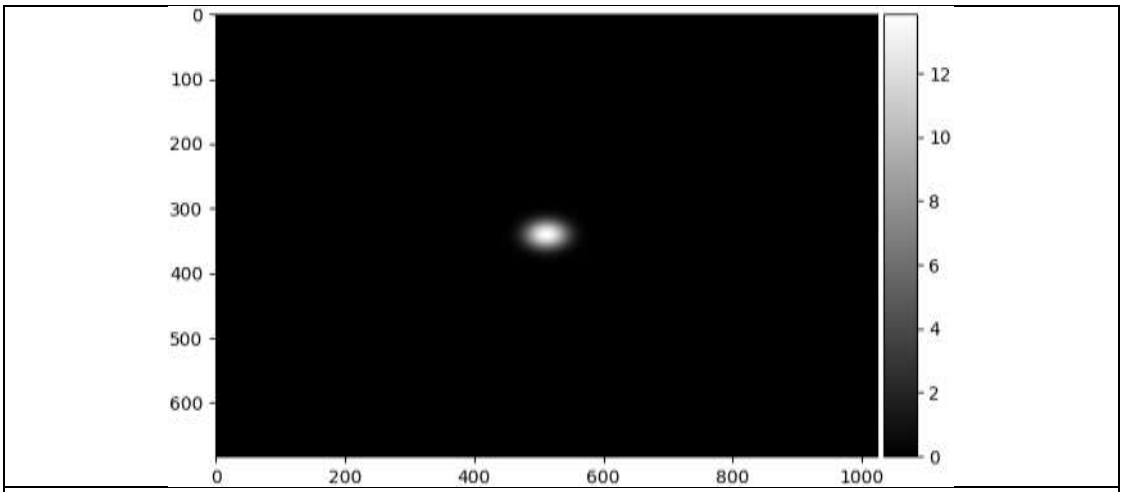

Fig 4. Fft of Gaussian filter with variance: 8, kernel size: 101

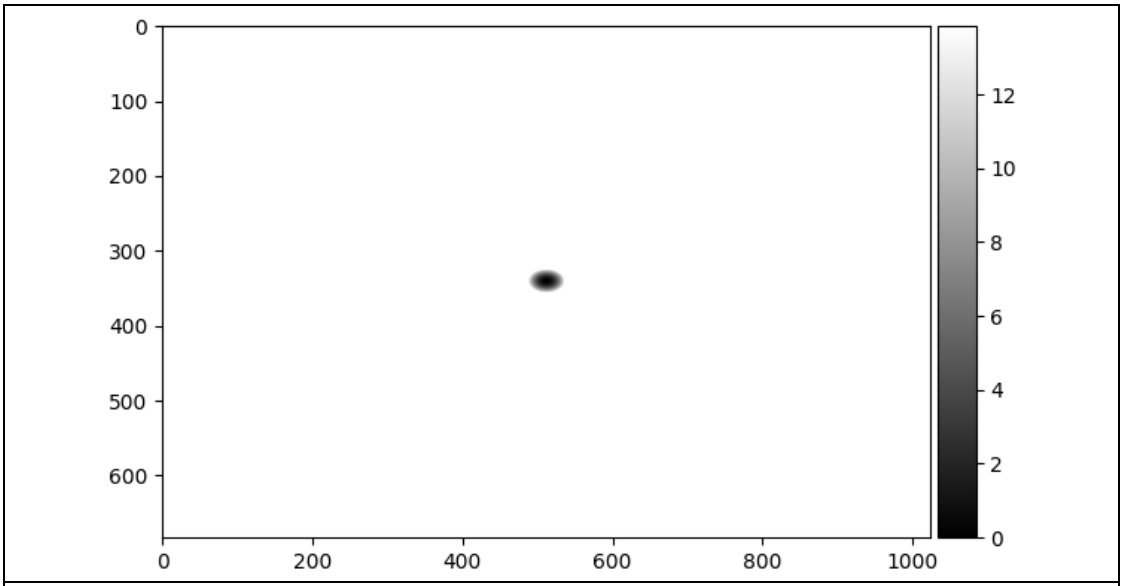

Fig 5. Inverse filter of Fig 4， min=1 max=2

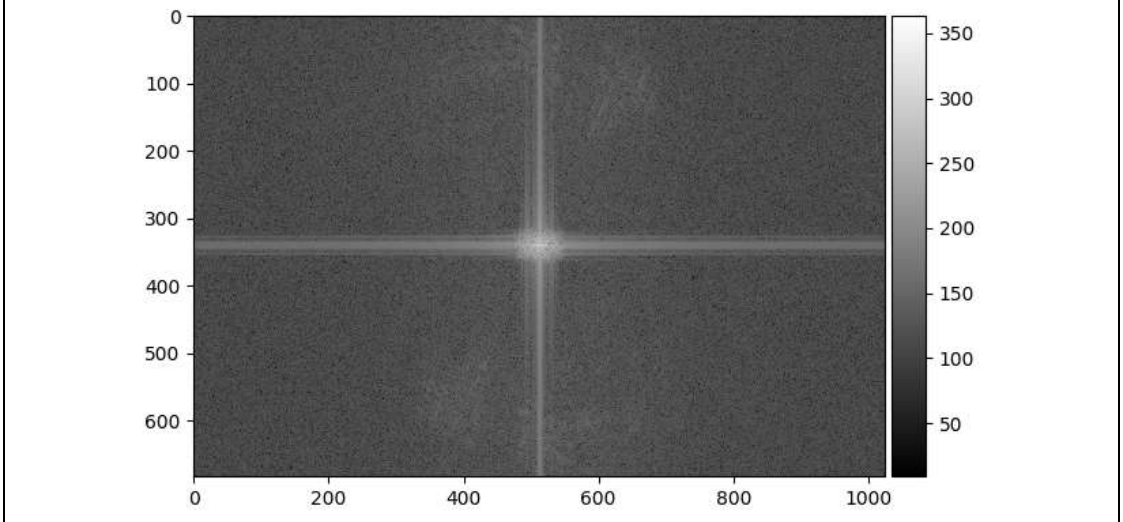

Fig 6. 還原後的圖，明顯看出來比第一種方式還原高頻的效果好很多

PSNR：

原圖: 150.342369789　還原後的圖: 152.097440778 進步率:1.167382815%

Motion blur:

Motion blur 為 圖形被一個 2D uniform distribution 所模糊化，其 kernel 的 sum 為 1。於是嘗試找到對應的 kernel，試著利用 wiener 的方式找回原本的圖片。

方法一: 使用 gaussian kernel

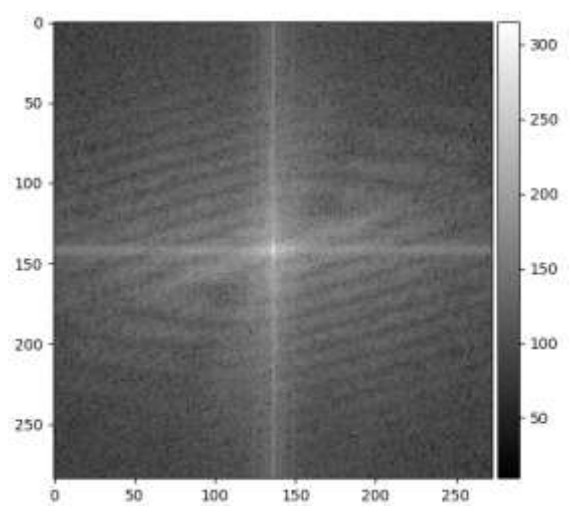

Fig 7.輸入圖的 FFT

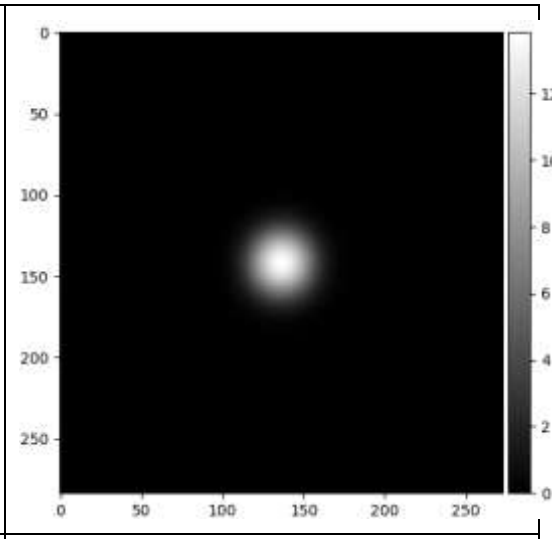

Fig 8. Gaussian filter with variance: 3.5, kernel size: 31

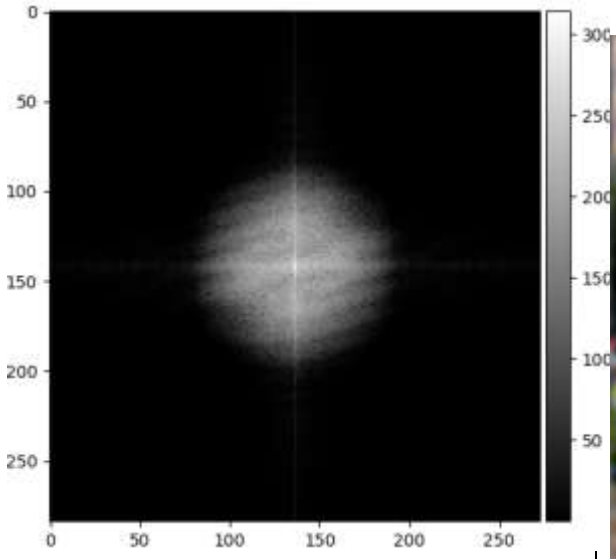

Fig 9. Result FFT

Fig 10. Result，雖然 motion 的部分被修正了，但是整張圖卻有許多 wiener 所產生的雜訊。

PSNR:

原圖: 135.170787807 還原後的圖: 139.209317573 進步率: 2.987723777%

方法二：使用 Motion Filter，使用 clip 不只用 wiener

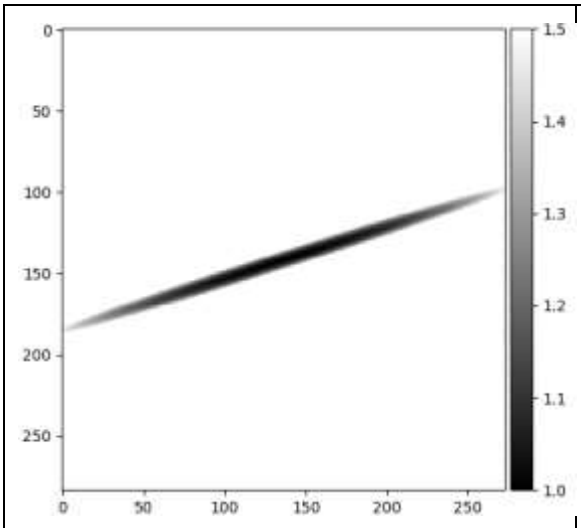

Fig 11. Motion filter, size: 25

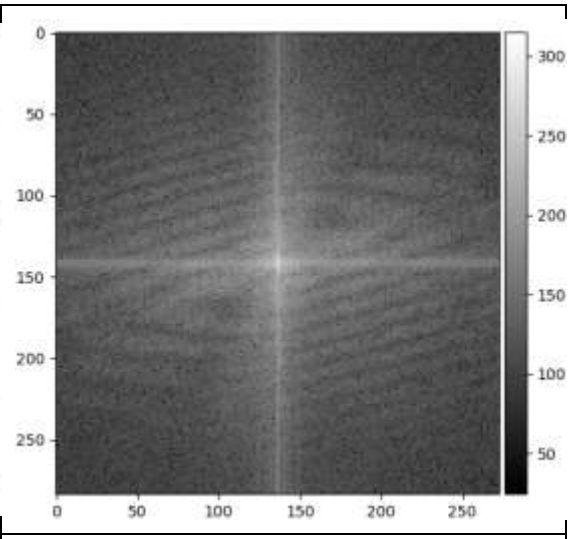

Fig 12. Result fft

Fig 13. result

PSNR:

原圖: 135.170787807 還原過的圖:136.307766733 進步 0.841142487%

Gaussian Blur + Noise:

使用 gaussian filter 和 wiener 來優化這張圖片。

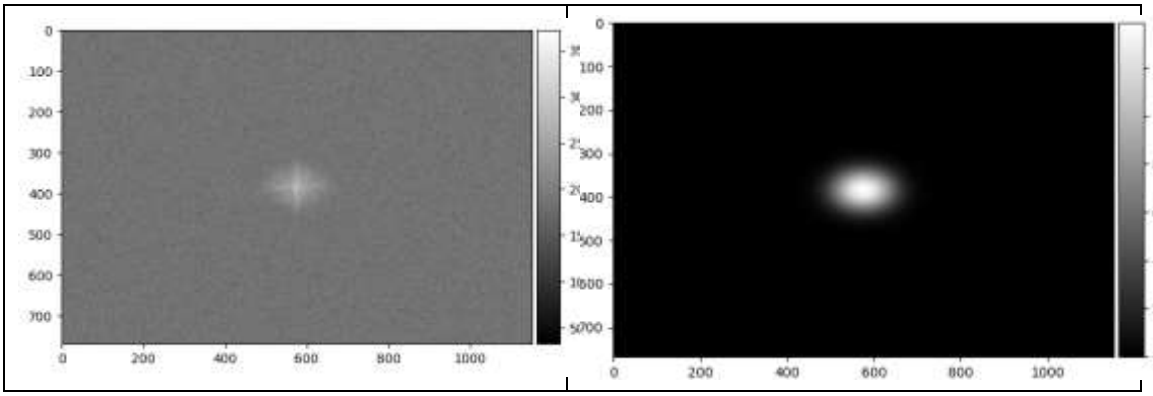

Fig 14. 原圖 FFT	Fig 15. Gaussian filter, variance: 4 size: 31
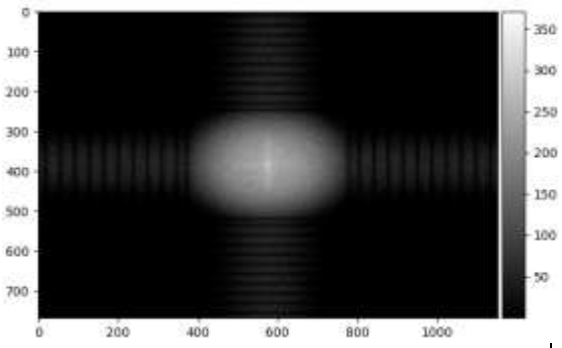	
Fig 16. fft of result	Fig 17. Result

PSNR:

原圖: 135.45608397 還原的圖：157.35884241 進步率：16.169638009%