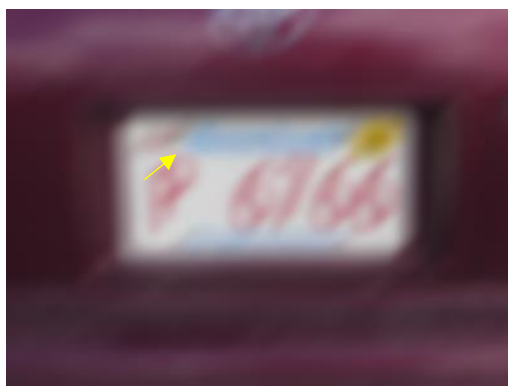


# Digital Image Processing Homework 4 Report

109511207 蔡宗儒




## 1. Image Restoration

本次作業要消除 Gaussian blur 和 motion blur，而我觀察模糊後的 input1.bmp 車牌上的 P 發現 motion blur 大概是往 45 度角方向位移，因此我就假設 point spread function 在此方向上，反覆設計往 45 度角、不同大小的 motion blur filter，再將此 motion blur filter 做 shift 跟 padding，將此結果做 DFT 轉成 frequency domain 後做 wiener filter，再將 wiener filter 與 input1.bmp 的 DFT 的 RGB 三通道相乘，最後轉回 spatial domain，得出 restoration 後的圖。另外我也有再加上 gaussian filter 來盡可能消除 gaussian blur，不過效果並不顯著。



最後做出的成果如下。經過多次嘗試參數後，我將 gaussian filter 的 kernel size 設為 5，motion blur filter 的 kernel size 設為 21，wiener filter 的  $k$  設為 0.1。可以觀察到儘管依然模糊，但已經可以正常辨識車牌號碼。然而雖然用肉眼感覺沒那麼模糊了，但我在計算 psnr 時發現這樣做出來的 psnr 並沒有變小，反而變大了。這是因為還原後會在圖上產生一圈一圈黑白相間的雜訊，算 psnr 就會不增反減。而當  $k$  變小的話這個問題就會更加地嚴重，不過我暫時沒有想到該如何解決此問題。

### Result

input1_ori.bmp	input1.bmp	output1.bmp
		

### Result(output1.bmp with different k)

$k = 0.1$	$k = 0.01$	$k = 0.001$
		

可以觀察到  $k$  越小的話，整個圖片所產生的割裂感越嚴重。

而第二張圖我有嘗試在 frequency domain 計算 input1.bmp 和 input1\_ori.bmp 的 transfer function，再用此來還原 input2.bmp，然而用此方法我最後得出的結果會是一張全黑的圖，但若還原 input1.bmp 成 input1\_ori.bmp 卻非常成功，我猜測問題應該是在不同 size 的圖片中要處理 frequency domain 的計算時有其他需要注意和考量的地方，而我並沒有注意到。因此我最後選擇和用和第一題一樣的方式，將 gaussian filter 的 kernel size 設為 5，motion blur filter 的 kernel size 設為 19，而 wiener filter 的 k 則設為 0.005。因為此題的要求是辨識車牌，所以儘管 k 較小會產生更嚴重的割裂感，psnr 也會越大，但同時也能將車牌還原地更好。結果如下，大部分的车牌號碼都已能還原。

## Result



Corresponding table

WYG573	11FH756	PHP2455	MKA532	405ZMU
MAV794	AFV2018	993KCM	YUT207	7121AN8
YMX644	MMG604	MKM239	378984K	JJS269
VS7SFL	JJS131	552AOY	2AA4510	RCA3412
992KCM	9427A06	HPR476	YUT042	HLFV4
8A231	4144AG?	YSE068	MHF686	342A
YUT002	HHG352	JGN048	SAB3399	11H38

## 2. Reference

[1] [FTW](#)

[2] [FTW3 学习笔记 2：FTW（快速傅里叶变换）中文参考](#)