影像處理 LAB4

學號 x1052078
姓名 陸周濤

5-3 Periodic noise reduction using a notch filter

做法說明

function [output_s] = addSinNoise(input_s, A, u0, v0)

Use 2 for loops to assign pixel value of ouput_s in spatial domain by the following MATLAB code:

```
output_s(i,j) = input_s(i, j) + A*sin(2*pi*(u0*i/M + v0*j/N));
```

which is derived from the formula given in specification:

$$\eta(x,y) = A \cdot \sin\left[2\pi\left(\frac{u_0 x}{M} + \frac{v_0 y}{N}\right)\right]$$

function [output_f, Notch] = notchFiltering(input_f, D0, u0, v0)

Given the fact that the spectrum of image is centralized, calculate targets of the two symmetric notches (u0, v0) and (u1, v1) by

```
u1 = M/2 - u0;
v1 = N/2 - v0;
u0 = M/2 + u0;
v0 = N/2 + v0;
```

where the (u0, v0) is initialized by the actual value of noise frequency.

Then, assign each values on each position with the threshold of D0.

Finally, use array multiplication to get the filtered frequency domain matrix.

function [psnr] = computePSNR(input1_s, input2_s)

By the definition of PSNR

$$PSNR(dB) = 10log_{10}\left(\frac{MN \cdot f_{peak}}{\sum_{x-0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2}\right)$$

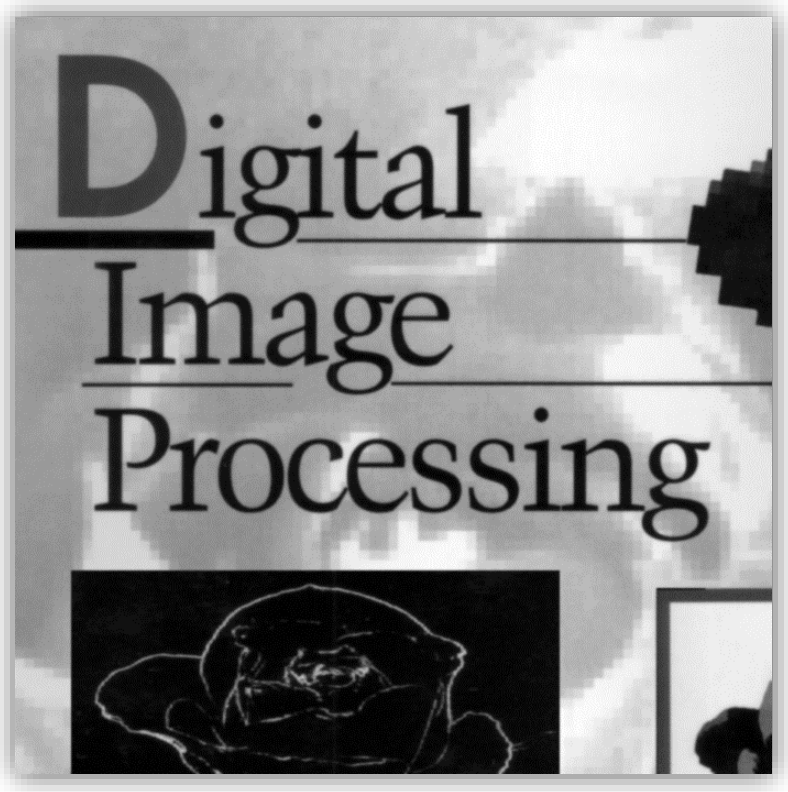
First, find the peak pixel value by

```
peak = max(max(input1_s));
```

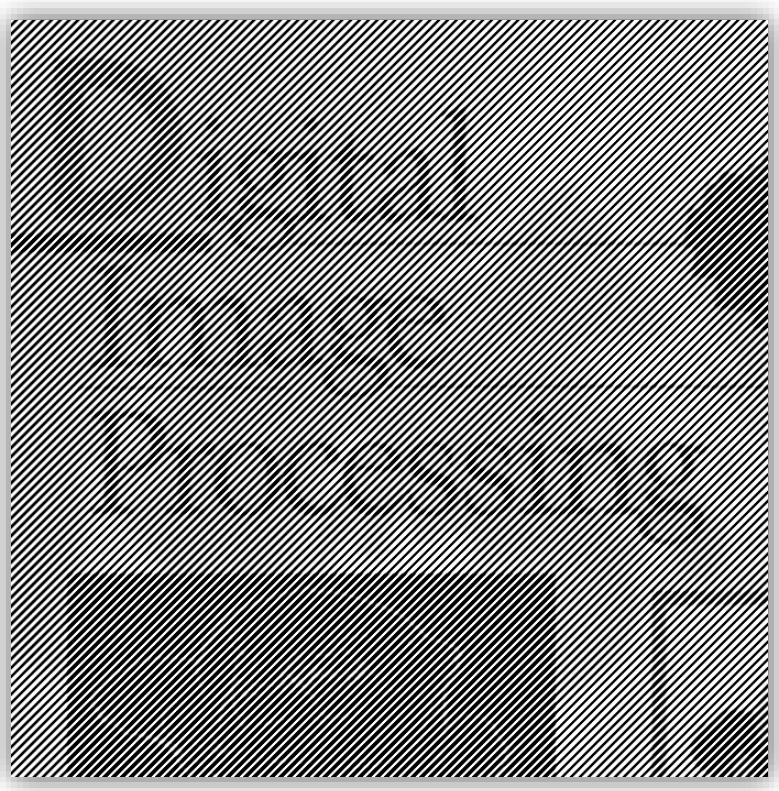
then calculate the PSNR as the definition stated.

```
psnr = 10*log(M*N*peak/sum(sum((input1_s-input2_s).^2)))/log(10);
```

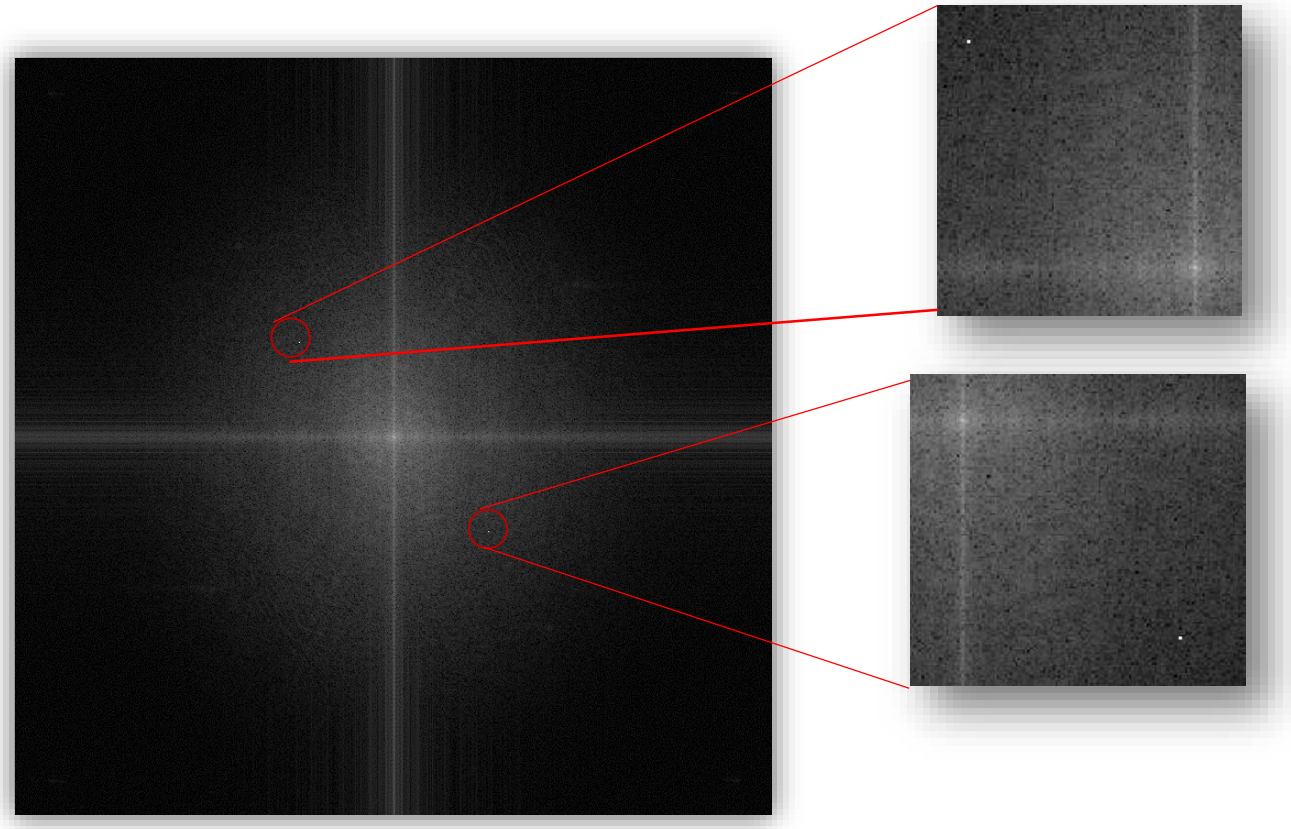
結果圖片



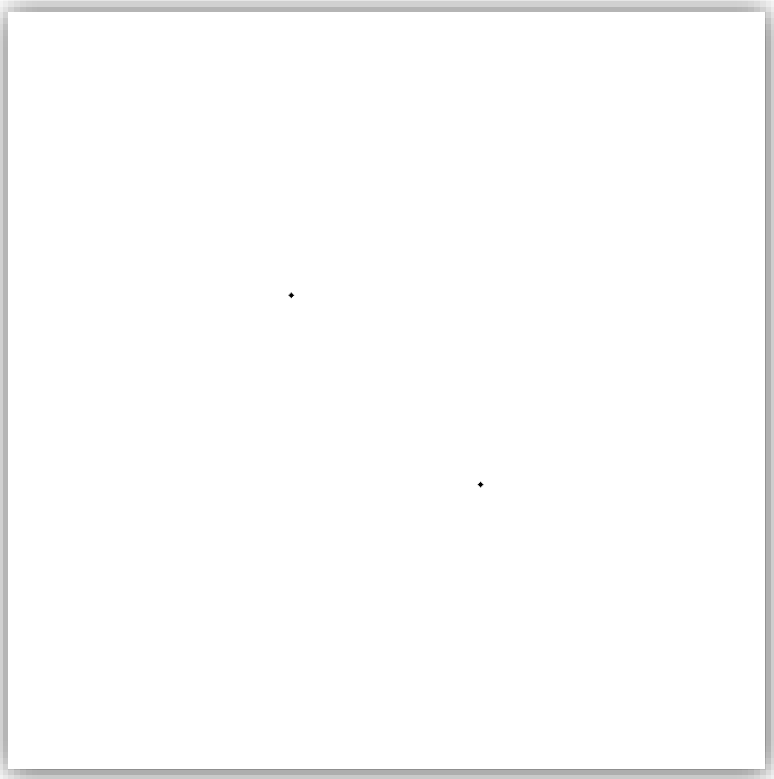
The origin image



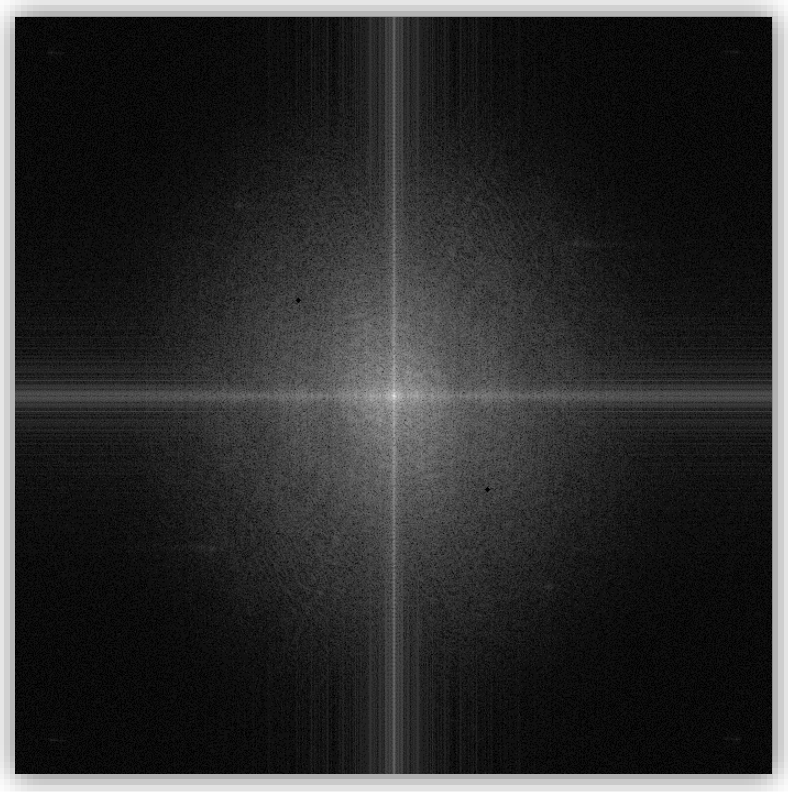
The noised image (A=100, u_0=M/8, v_0=N/8)



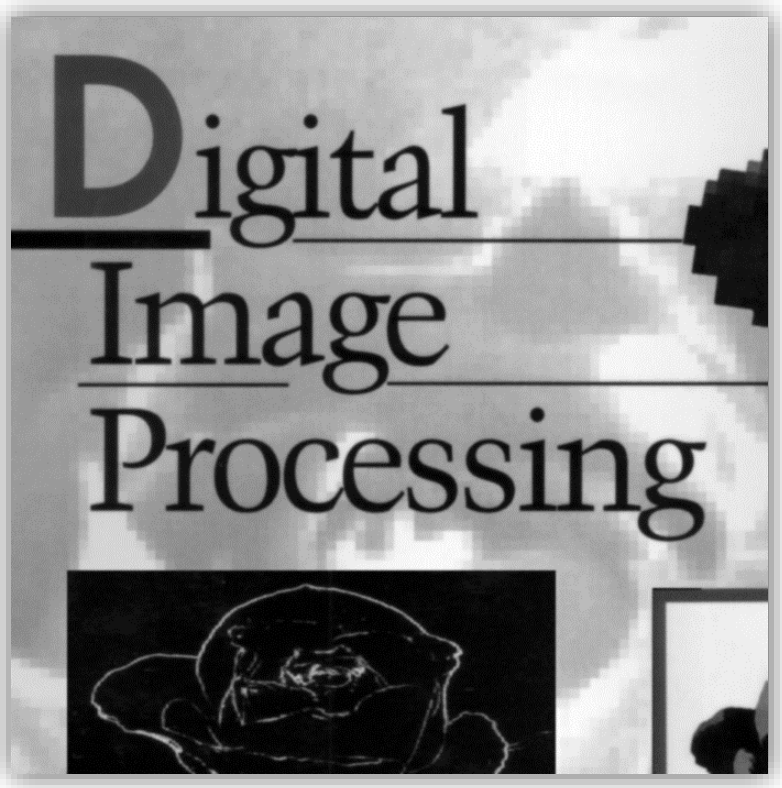
Frequency spectrum



Notch filter



Filtered frequency spectrum



Resulted spatial image

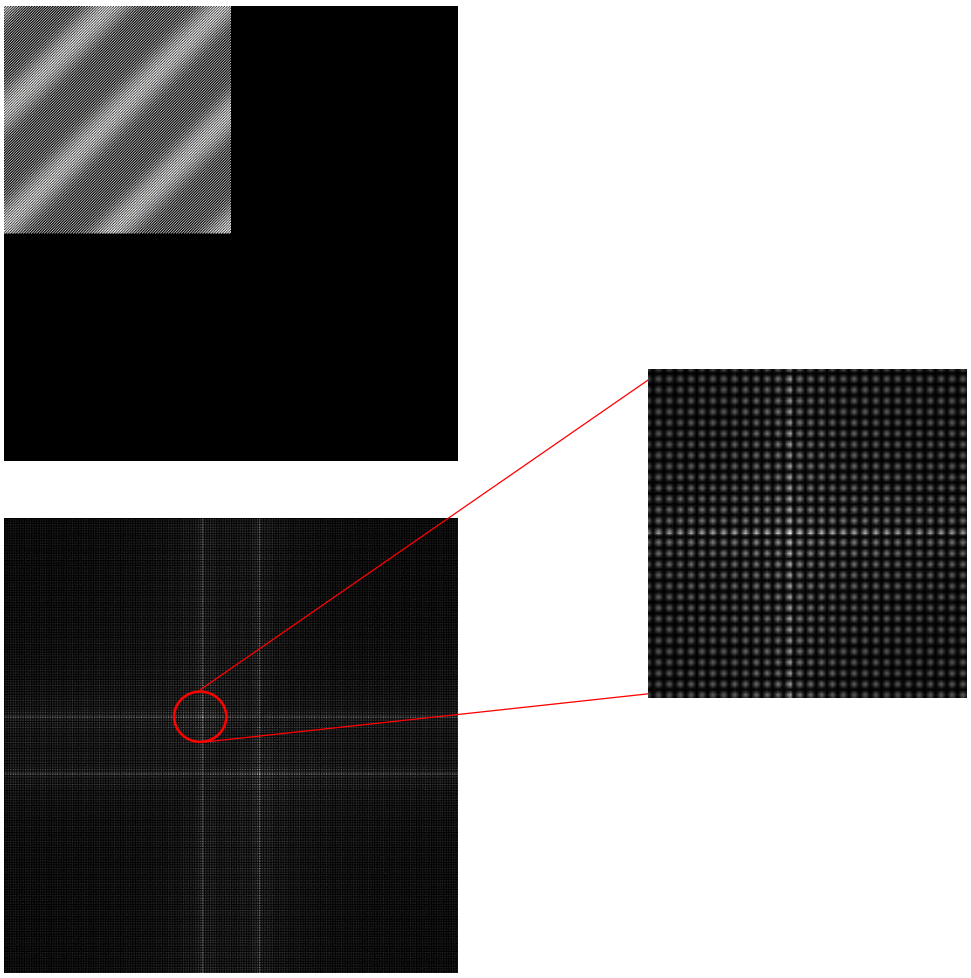
```
>> problem1

psnr =

   74.4384
```

PSNR of the estimated image

分析以及討論

1. The effect of padding zero in the ordinary process of frequency domain filtering.

   Given the importance of padding zero to correct the border error while filtering the image in frequency domain, the result images showed above, contrarily, do not pad zero throughout the whole processing, for the reason that padding zero will cause some unexpected consequences stated as following.



Given the formula of the periodic noise, we can easily get the corresponding frequency spectrum by the following derivation:

$$N(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} A \cdot \sin\left[2\pi\left(\frac{u_0 x}{M} + \frac{v_0 y}{N}\right)\right] e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$= \frac{A}{2j}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[e^{j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)} - e^{-j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)}\right] \cdot e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$= \frac{A}{2j}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} 1 \cdot e^{j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)} \cdot e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$- \frac{A}{2j}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} 1 \cdot e^{-j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)} \cdot e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$= \frac{A}{2j}\Im\{1 \cdot e^{j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)}\} - \frac{A}{2j}\Im\{1 \cdot e^{-j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)}\}$$

$$= \frac{A}{2j}(\delta(u-u_0, v-v_0) - \delta(u+u_0, v+v_0))$$

Hence, the expected 'impulse' is on the coordinate (u_0, v_0) and $(-u_0, -v_0)$

But if we padded zero on the image to make its size doubled, the expected 'impulses' are totally changed. The noise is described as following.

$$\eta(x,y) = \begin{cases} A \cdot \sin\left[2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)\right] & ,x < M \text{ and } y < N \\ 0 & ,others \end{cases}$$

, which can be regarded as the multiplication of sinusoid noise and box noise. That is if

$$\eta_1(x,y) = A \cdot \sin\left[2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)\right]$$

$$\eta_2(x,y) = \begin{cases} 1, x < M \text{ and } y < N \\ 0, \quad others \end{cases}$$

, then the noise could be expressed as following.

$$\eta(x,y) = \eta_1(x,y) \cdot \eta_2(x,y)$$

So when we transform to frequency domain, that is

$$N(u,v) = N_1(x,y) * N_2(x,y)$$

, where * denotes the operation of convolution.

$$N_1(u,v) = \sum_{x=0}^{2M-1}\sum_{y=0}^{2N-1} \mathrm{A} \cdot \sin\left[2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)\right] e^{-j2\pi\left(\frac{ux}{2M}+\frac{vy}{2N}\right)}$$

$$= \frac{A}{2j}\sum_{x=0}^{2M-1}\sum_{y=0}^{2N-1}[e^{j2\pi\left(\frac{2u_0 x}{2M}+\frac{2v_0 y}{2N}\right)} - e^{-j2\pi\left(\frac{2u_0 x}{2M}+\frac{2v_0 y}{2N}\right)}]$$

$$\cdot e^{-j2\pi\left(\frac{ux}{2M}+\frac{vy}{2N}\right)}$$

$$= \frac{A}{2j} \sum_{x=0}^{2M-1} \sum_{y=0}^{2N-1} 1 \cdot e^{j2\pi\left(\frac{2u_0 x}{2M} + \frac{2v_0 y}{2N}\right)} \cdot e^{-j2\pi\left(\frac{ux}{2M} + \frac{vy}{2N}\right)}$$

$$- \frac{A}{2j} \sum_{x=0}^{2M-1} \sum_{y=0}^{2N-1} 1 \cdot e^{-j2\pi\left(\frac{2u_0 x}{2M} + \frac{2v_0 y}{2N}\right)} \cdot e^{-j2\pi\left(\frac{ux}{2M} + \frac{vy}{2N}\right)}$$

$$= \frac{A}{2j}(\delta(u - 2u_0, v - 2v_0) - \delta(u + 2u_0, v + 2v_0))$$

And we can get $N_2(x, y)$ from the textbook, which is of the form of

$$F(\mu) = AW \frac{sin(\pi\mu W)}{\pi\mu W}$$

The detailed continuous formula is derived as following:

$$N_2(u, v) = \int_0^{2M} \int_0^{2N} \eta_2(x, y) e^{-j2\pi\left(\frac{ux}{2M} + \frac{vy}{2N}\right)} dxdy$$

$$= \int_0^{M} e^{-j2\pi\left(\frac{ux}{2M}\right)} dx \int_0^{N} e^{-j2\pi\left(\frac{vy}{2N}\right)} dy$$

$$= \left[\frac{-M}{j\pi u}\left(e^{-j2\pi\left(\frac{ux}{2M}\right)}\right)\Big|_0^M\right] \cdot \left[\frac{-N}{j\pi v}\left(e^{-j2\pi\left(\frac{vy}{2N}\right)}\right)\Big|_0^N\right]$$

$$= \frac{-M}{j\pi u}\left(e^{-j2\pi\left(\frac{u}{2}\right)} - 1\right)\frac{-N}{j\pi v}\left(e^{-j2\pi\left(\frac{v}{2}\right)} - 1\right)$$

$$= \frac{-MN}{\pi^2 uv}\left[\left(e^{-j\pi\left(\frac{u}{2}\right)} - e^{j\pi\left(\frac{u}{2}\right)}\right)e^{-j\pi\left(\frac{u}{2}\right)}\right]\left[\left(e^{-j\pi\left(\frac{v}{2}\right)} - e^{j\pi\left(\frac{v}{2}\right)}\right)e^{-j\pi\left(\frac{v}{2}\right)}\right]$$

$$= \frac{-MN}{\pi^2 uv}\left[\left(-2j \cdot \sin\left(\frac{\pi u}{2}\right)\right)e^{-j\pi\left(\frac{u}{2}\right)}\right]\left[\left(-2j \cdot \sin\left(\frac{\pi v}{2}\right)\right)e^{-j\pi\left(\frac{v}{2}\right)}\right]$$

$$= \mathrm{MN}e^{-j\pi\left(\frac{u+v}{2}\right)}\left[\frac{\left(\sin\left(\frac{\pi u}{2}\right)\right)}{\frac{\pi u}{2}}\right]\left[\frac{\left(\sin\left(\frac{\pi v}{2}\right)\right)}{\frac{\pi v}{2}}\right]$$

The multiplication in spatial domain is corresponding to the convolution in frequency domain. Hence, make the convolution of impulse $N_1(x, y)$ and the sinc-wise $N_2(x, y)$. And we've know that the convolution between impulse and any other matrix is equivalent to the copy of the second matrix to the central of the impulse, which

exactly explain what we get from padding zero to the sine wave before Fourier transformation. The centers of the impulses in 2 direction are moved to $(2u_0, 2v_0)$ and $(-2u_0, -2v_0)$, and since the size of image is doubled, the proportion of the distance from the impulse to the center of the image is still $\frac{1}{8}$.

2. The effect of D_0 to PSNR

   The parameter D_0 is used to control the pass area the notch filter. If the window is too large, thus we might reject some areas belonging to the original image, decreasing PASNR; if the window is too small, thus the impulse of the noise is not rejected also decreasing PSNR.

   In this case, since the image is large enough, to make the 'impulse' almost in one pixel. So then by some trials, D_0 =1.5 become the best candidate (since sqrt(2)=1.414).

D0	PSNR
1.4	-37.0239*
1.5	74.4394
2	73.1298
2.5	69.9884
3	67.8407
4	65.4145

   *the window is too small (< sqrt(2)) and doesn't work, so there is a huge difference between too images and PSNR become negative.

3. The effect of noise frequency (u0, v0) on PSNR

   Since the notch filter in frequency domain is just to make the values of area where the noise frequency locates zero regardless of the origin value. Thus different frequency of the noise will cause different results after notch filter.

   To illustrate it, some sample frequency pairs are chosen to get this idea. And note that the lower the frequency is, the more the notch filter will corrupt the image.

u_0	v_0	PSNR
86	86	74,.4384
100	100	75.7273
150	150	89.3048
172	172	94.4801
200	200	95.4130
300	300	97.5851

But, we can also note that no matter how far it is from the DC component, the PSNR will never tend to infinite.

5-4 Parametric Wiener filter

做法說明

```
function [ output_f, H ] = addMotionBlur( input_f, T, a, b )
```

Use 2 for loops to assign each values from top-left to bottom-right. Since the input frequency spectrum has been centralized (required), the matrix H, which is the transform for motion blur in frequency domain, also needs to be centralized. Thus, for every coordinate (x, y), first fix it to (u, v) by the following codes:

```
u = x-M/2;
v = y-N/2;
```

And then, apply the formula to (u, v). That is,

$$H(x,y) = H(u,v) = \frac{T}{\pi(au+bv)}\sin(\pi(au+bv))e^{-j\pi(au+bv)}$$

, except for the frequency pairs which will make the phase term $\pi(au+bv)$ be zero. In this exception, use the limit of $H(u,v)$. That is

$$\lim_{\alpha\to 0}\frac{T\sin(\pi\alpha)e^{-j\pi\alpha}}{\pi\alpha} = \lim_{\alpha\to 0}\frac{T[\sin(\pi\alpha)\,e^{-j\pi\alpha}]'}{\pi(\alpha)'}$$

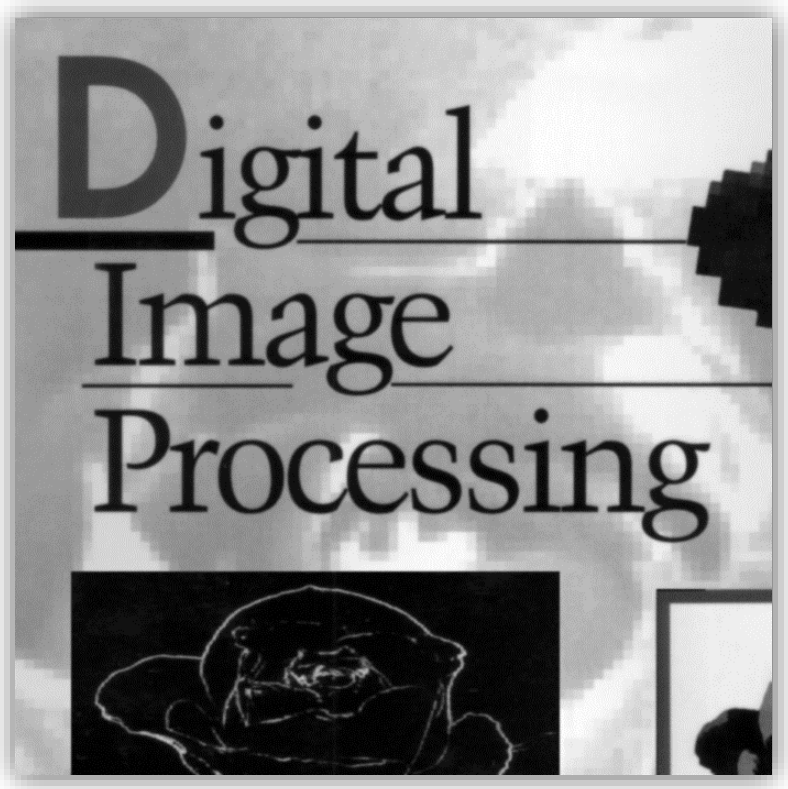
$$= \lim_{\alpha\to 0}\frac{T[\pi\cos(\pi\alpha)\,e^{-j\pi\alpha} + (-j\pi)\sin(\pi\alpha)\,e^{-j\pi\alpha}]}{\pi} = \lim_{\alpha\to 0}\frac{T\pi}{\pi} = T$$

```
function [ output_f ] = wienerFilter( input_f, H, K )
```

Just apply the formula derived to the input frequency with the estimated degradation model and the tested proportion of noise spectrum and original image spectrum.

```
W = conj(H)./(abs(H).^2+K);
output_f = W.*input_f;
```

結果圖片



Original image

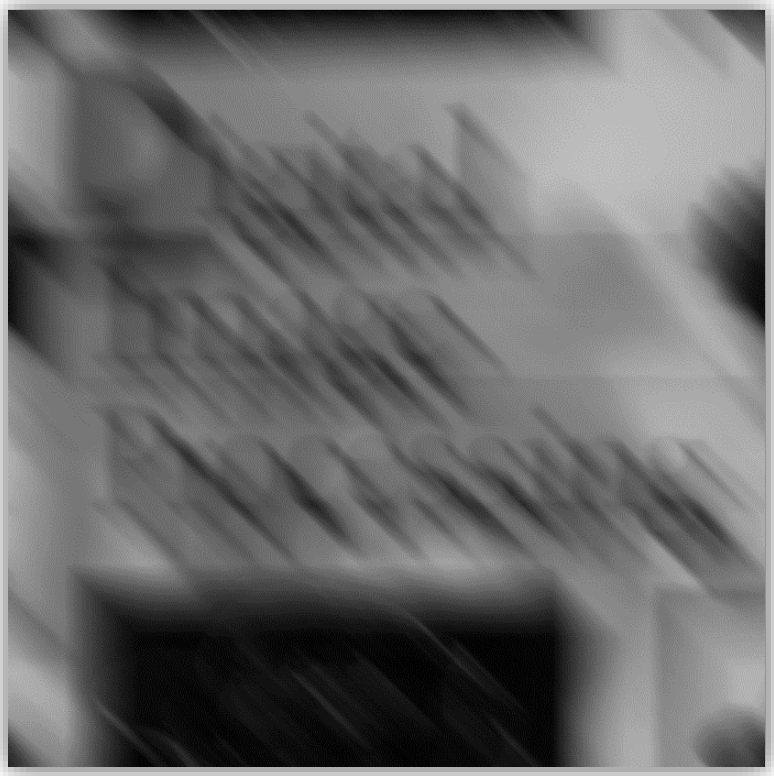


Image corrupted by motion blur

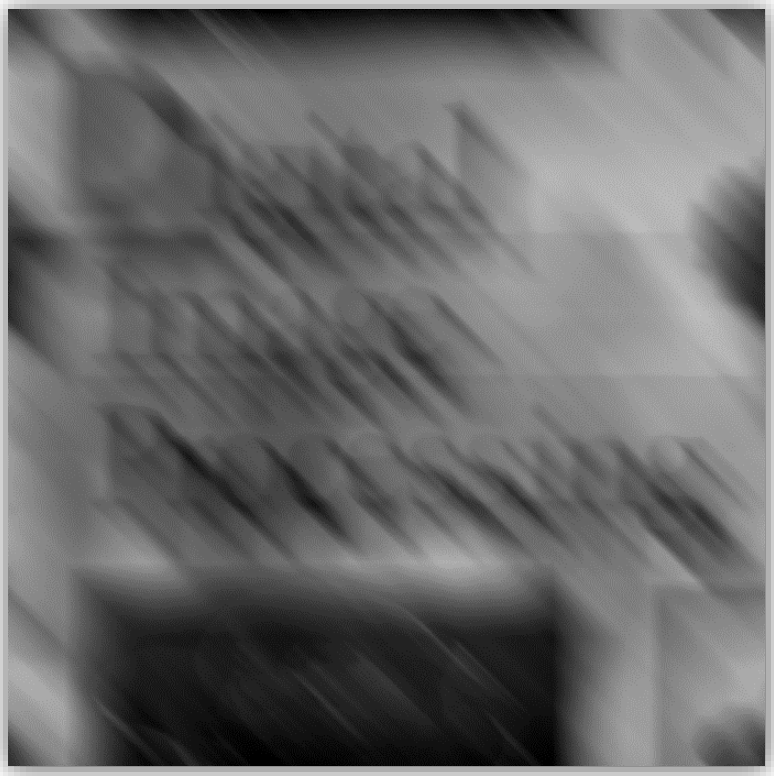


Image corrupted by motion blur and sine noise

Restored by Weiner with K = 0.001 (PSNR=11.3937)

Restored by Weiner with K = 0.005 (PSNR=10.9178)

Restored by Weiner with K = 0.01 (PSNR=10.5905)

Restored by Weiner with K = 10^{-16} (PSNR=11.7475)

分析以及討論

The image above is degraded by the uniform and linear motion blur of T=1, a=b-0.1and sine noise of frequency (50/688, 50/688) and magnitude A = 10^{-3}. And then apply the Weiner to restore the corrupted image. As we can see from the above resulted images with NSR K varying from 0.001 to 0.1 (one more case for K = 10^{-16}), the noise component and the origin image get the opposed effects. When the NSR down to be near zero, the original image become clearer, while the noise seems be amplified. In the additional extreme case of K = 10^{-16}, we can see the words in the image is close to the original image, but with the sine noise more visible.

To explain this phenomenon theoretically, it need to refer to the formula of Weiner filter. The degradation model of motion blur, as what the name suggests, is a smoothing operator in frequency domain. And inversely, its inverse is a high pass filter in frequency domain. So when the K is relatively large, the blurred component is

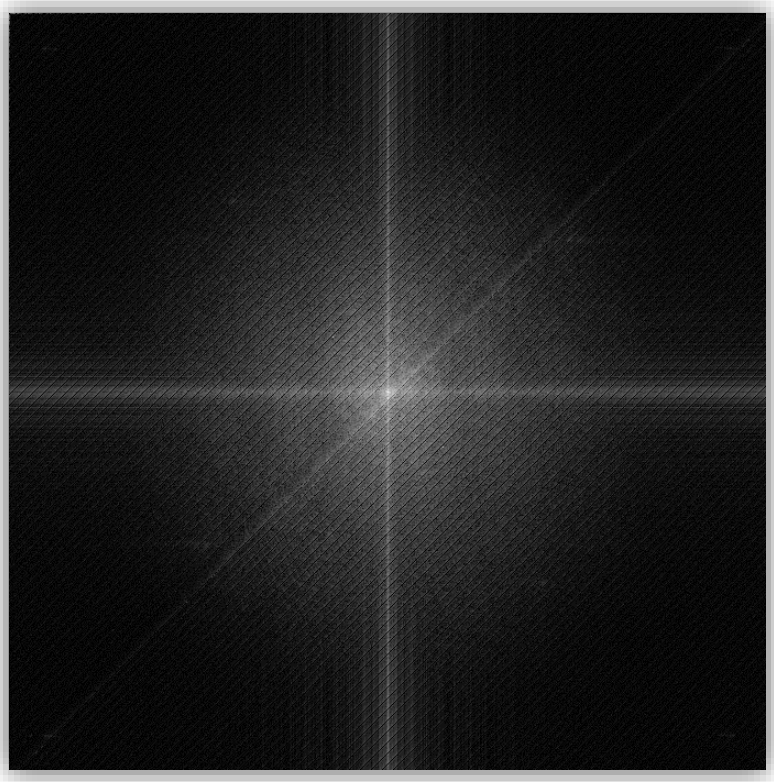
not fully de-blurred and the words appear to be blurred. And when K is estimated to tend to zero, the origin image is fully de-blurred but the since the inverse of the H is a high pass filter, the noise part is sharpened, making the noise much more visible.

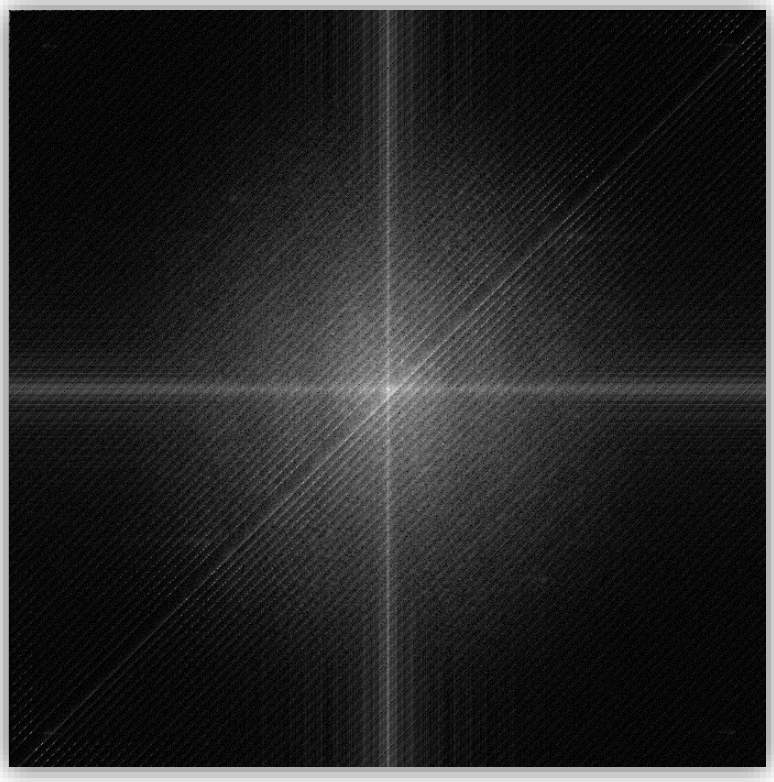
Note that K is never be estimated as zero even if there is no noise, which will not restore the blurred image. To find the reason, I inspect the content of the model H, where some value tending to zero is found, which means the inverse of H will include some values tending to infinity. And the corresponding extremely small value in the blurred frequency spectrum was approximated and loose the exact order for the precision requirement of storage and then if we apply the inverse to it, the resulted matrix will also contain the almost infinite values, failing the restoration.

Finishing the visible effect of different estimated NSR K, we can turn to the numerical check of the measurement PSNR.

K	PSNR
0.1	8.2591
0.01	10.5905
0.005	10.9178
0.001	11.3937
0.0001	11.6664
10^{-16}	11.7475
10^{-18}	10.0708

Here, we can see when K decrease below 10^{-16}, the PSNR also declines, which seems to originate from the nearly zero values in H, which is more credible from the spectrum of them. The following spectrums are K=10^{-16}(former) and 10^{-18}(latter).





From the 2 spectrum image, we can see the periodic dark stripes in the former spectrum as opposed to the bright strips in the latter spectrum which is from the

infinite value is the inverse of H.