

Homework 2:

Introduction to Image and Video Processing

Huang Yiping

1 Periodicity, Frequency Filtering

1.1 Create a two dimensional periodic signal with one frequency (e.g. sin, cos...).

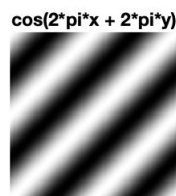
(a) Calculate its 2D FT analytically ("on paper" with math) and by applying the 2D FFT

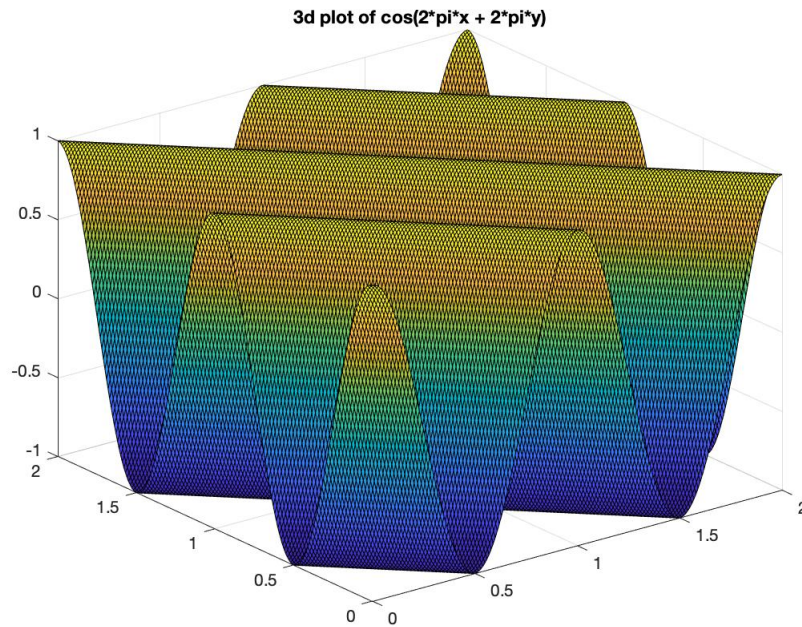
Answer:

$$\begin{aligned}
 &\text{choose } f(x,y) = \cos(2\pi x + 2\pi y) \\
 &F(f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-j2\pi(ux+vy)} dx dy \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{e^{j2\pi(x+y)} + e^{-j2\pi(x+y)}}{2} e^{-j2\pi(ux+vy)} dx dy \\
 &= \frac{1}{2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-j2\pi(ux+vy)} e^{j2\pi(x+y)} dx dy \\
 &\quad + \frac{1}{2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-j2\pi(ux+vy)} e^{-j2\pi(x+y)} dx dy \\
 &= \frac{1}{2} \delta(u-1, v-1) + \frac{1}{2} \delta(u+1, v+1) \quad \leftarrow \text{frequency shifting}
 \end{aligned}$$

Notice the FFT of 1 is $2\pi\delta(\omega)$, which can be calculated by the FFT of $e^{-a|t|}$ ($a > 0$), and then calculate the limit as $a \rightarrow 0$. And if $\omega = 0$, the limit is ∞ , and for $\omega \neq 0$, the limit is 0. And the integral from $-\infty$ to ∞ is 2π , so the fft of 1 is $2\pi\delta(\omega)$ in angle frequency form. For the frequency u , this simplifies to $\delta(u)$

The image $\cos(2\pi x + 2\pi y)$ is shown below in 2D and 3D.

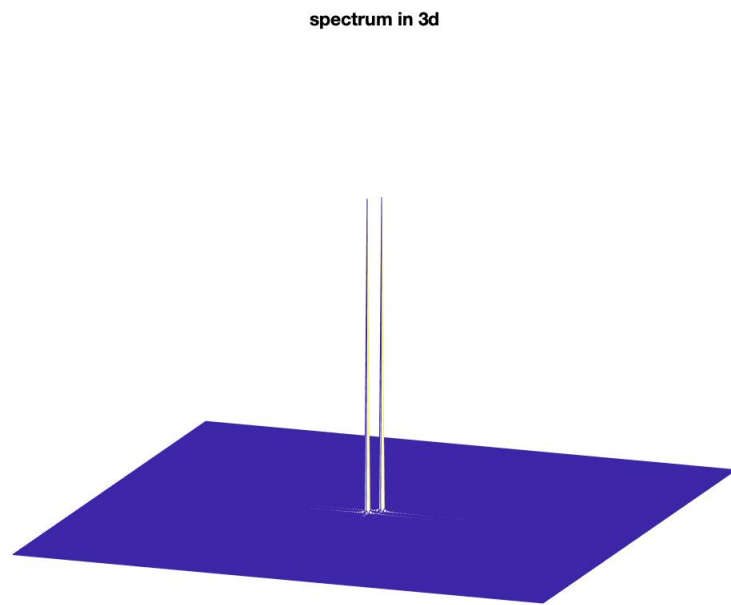
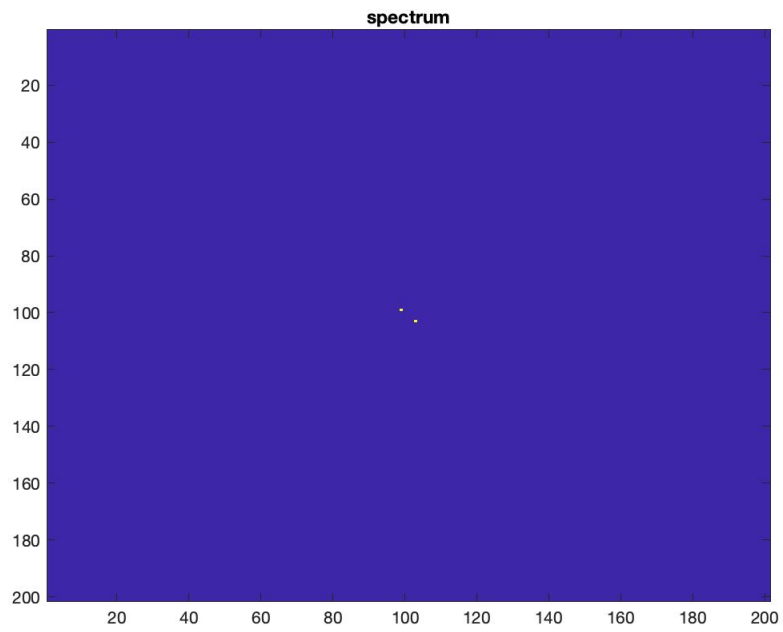


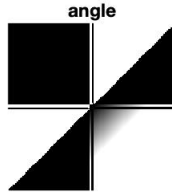


(b) Depict its 2D magnitude and phase after shifting the center of frequency coordinates to the center of your image. What do you observe?

Answer:

The spectrum in 2d and 3d and the phase diagrams are plotted below. It is clear from both the 2d and 3d plot that there are two impulses in the spectrum, which are even symmetric about the origin. This confirms that the previous calculation is correct. The phase diagram shows that the phase is odd symmetric about the origin. In fact, for a real function, its fourier transform is conjugate symmetric. That is $F^*(u, v) = F(-u, -v)$. And the above observation can be derived from this property.





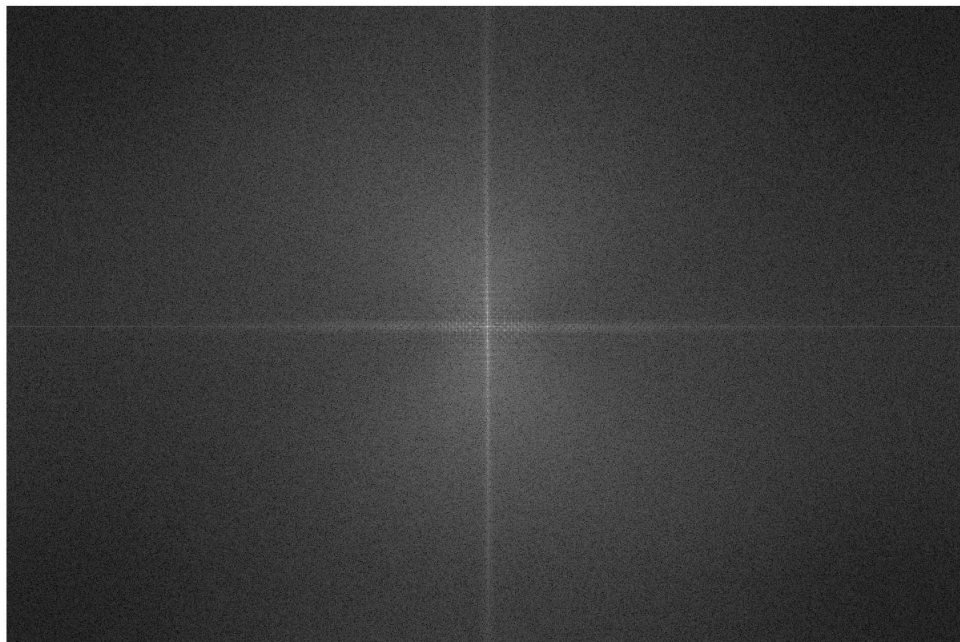
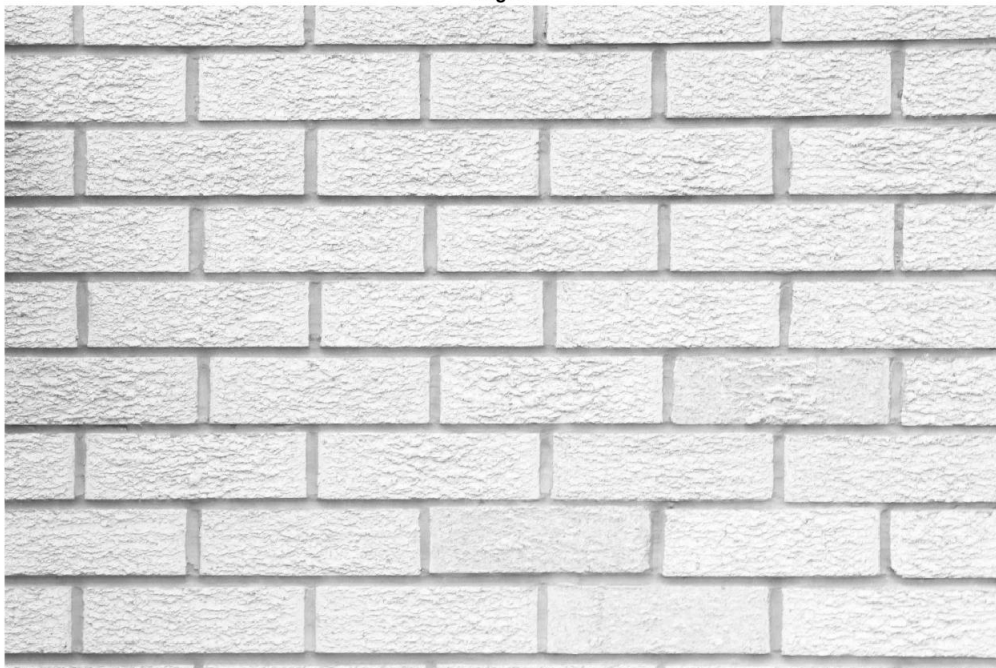
1.2 Create or find a clearly periodic image with a repeating pattern in the x, y or both directions. This should be a more complex 2D periodic function than in the previous question.

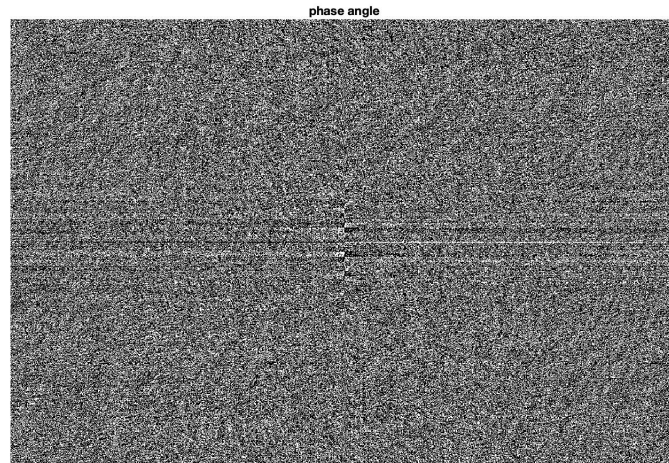
(a) Depict its 2D magnitude and phase after shifting the center of frequency coordinates to the center of your image. What do you observe?

Answer:

The below image of brick wall is chosen to be representative of a periodic image. We see that many bright spots in the spectrum along the vertical and horizontal axis. This corresponds to the repeated intensity value changes along the vertical and horizontal directions in the original image. From the phase diagram, there's not much we can see. However, phase provides important information about the image, namely the phase shift of each frequency component. If you reconstruct the image by only the phase, you can still see the original image, just the intensity values will go away.

original



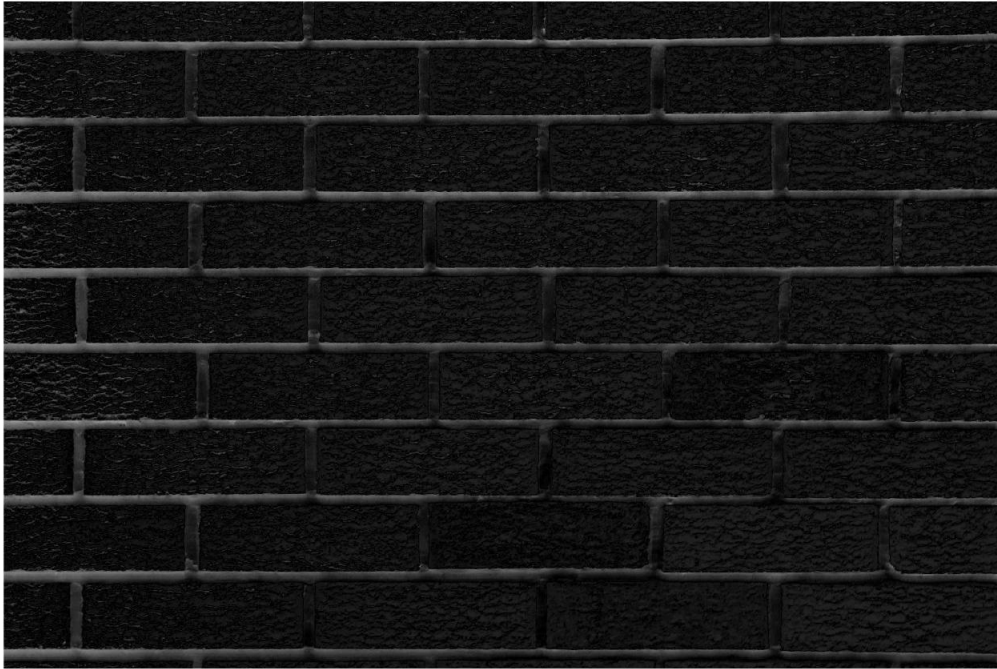


(b) Remove the strongest frequency from its FT and then find the inverse 2D FFT. Depict the resulting image and discuss your results.

Answer:

We see below the image by inverse FFT after removing the strongest frequency in the spectrum. The image is much darker than it was since the average intensity becomes 0 after the change. In fact, the image now also contains negative values. The reason behind is that for an uncentered FFT, the $F(0,0)$ corresponds to the DC value, i.e., the average value. This can be directly calculated by DFT formula.

Average intensity of 0



2 Periodic noise removal

- 2.1 Choose an image of your liking and add periodic noise to it. You have to decide what kind of periodic noise you want to add. For the rest of this exercise you should assume you have been given only the noisy image and that this noise is unknown to you.**

Answer:

The image chosen is the same as from project 1 as shown below. The periodic noise function chosen is $f(x, y) = 100 * \sin(2\pi 50x/m + 2\pi 100y/n)$. This is added to the original image, and the image with noise is shown next.

original



with noise

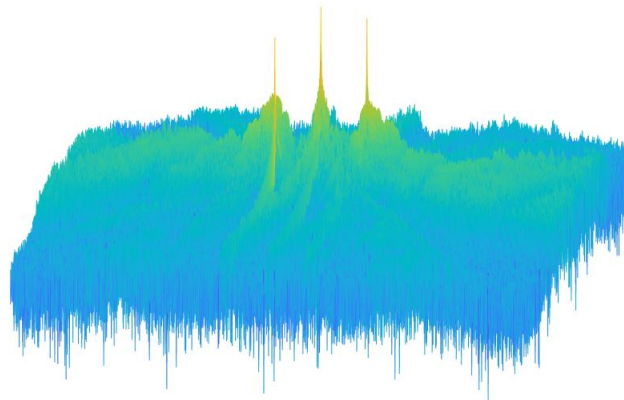


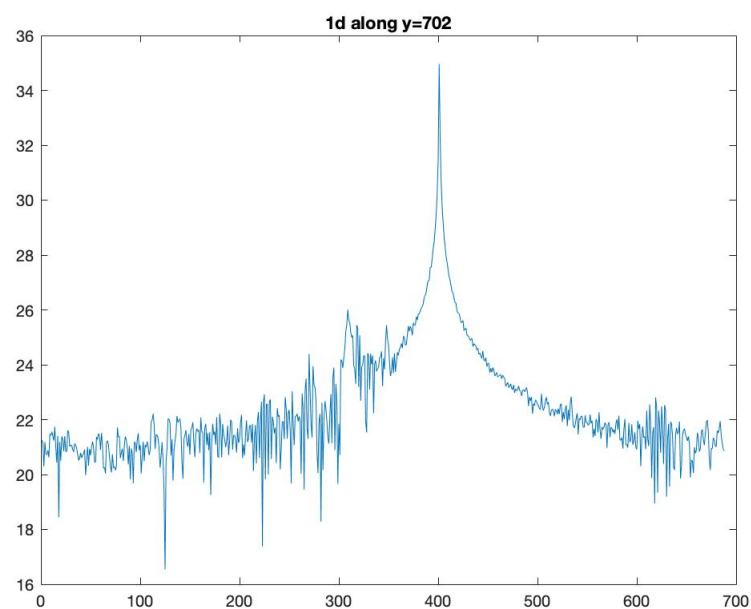
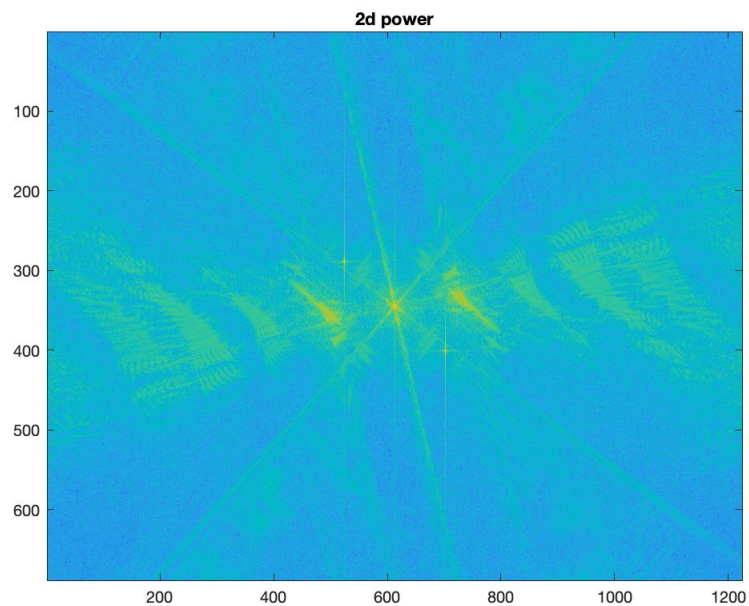
2.2 2. Calculate the 2D FFT of the noisy image(using an inbuilt function like `fft2`, `numpy.fft. fft2`). Display the noisy image's power spectrum in 1D, 2D, 3D and comment on it. What does it reveal about the noise?

Answer:

The power spectrum in 1D, 2D and 3D are shown below. We can clearly see from the 2D and 3D plot that there are two bright spots near the center of the image, one at the top left, the other at the bottom right both relative to the center. They correspond to the frequencies of the periodic noise pattern. For the 1D image, the cross section is chosen along $v = 702$ vertical line. The impulse is also clearly shown around $u = 400$. We know that for a sine wave, its image in the frequency domain are two impulses that forms a conjugate pair. These images verify this fact.

3d power



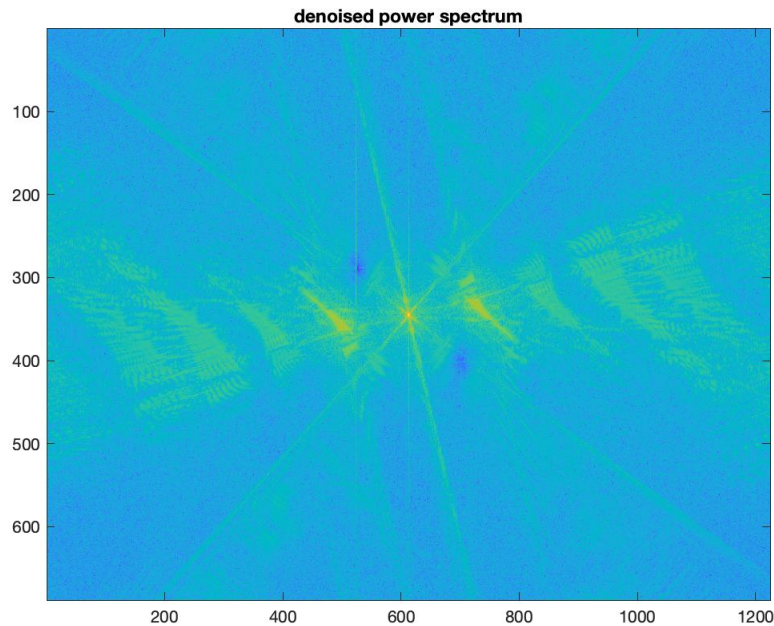


2.3 Find a way to remove the periodic noise in the frequency domain. Then show the de-noised image (in space) and its power spectrum. Discuss your approach and results.

Answer:

To remove the periodic noise, we use notch reject filter to filter the unwanted spots. The general notch filter form is $H_{NR}(u, v) = \prod_{k=1}^Q H_k(u, v)H_{-k}(u, v)$. These unwanted spot showcase themselves in the locations where discontinuity in intensity can be observed in the power spectrum, usually those spots that are much brighter than the nearby points. They are found interactively with cursor location on the coordinate system of the image. The result contains two locations, one at point $x=702, y=401$, the other at point $x=527, y=289$. To specify the notch reject filter, we also need to specify the desired cutoff frequency. In this case, we choose d_0 to be 20 for both two locations. Since we are using notch reject filter, so we need to choose highpass filter as individual component of the notch filter. The highpass filter is 1 minus the lowpass filter. In this case, we use gaussian, so the lowpass filter is of the form $e^{-(D^2(u,v)/2D_0^2)}$.

The product of frequency response and the FFT of the image is as shown below. We notice that the bright spots have been covered up with these notch filters. The inverse transform is shown next. We found with this setting, the noise has been basically removed except at the boundary. This is perhaps due to the lack of padding in the image.



after restoration



3 Image restoration

- 3.1 Choose an image of your liking and blur it with a spatial kernel h of your choice. Add random noise n following a specific noise distribution.

Answer:

The same image as in problem 2 is chosen, and is shown again for convenience.

original



A gaussian spatial kernel with size 5×5 is chosen with variance of 2. The resulting kernel looks like this:

$$M_{Gaussian} = \begin{pmatrix} 0.0232 & 0.0338 & 0.0383 & 0.0338 & 0.0232 \\ 0.0338 & 0.0492 & 0.0558 & 0.0492 & 0.0338 \\ 0.0383 & 0.0558 & 0.0632 & 0.0558 & 0.0383 \\ 0.0338 & 0.0492 & 0.0558 & 0.0492 & 0.0338 \\ 0.0232 & 0.0338 & 0.0383 & 0.0338 & 0.0232 \end{pmatrix}$$

This kernel is used to blur the image using the matlab `imfilter` function. The blurred image is shown below.

with blur



Next, gaussian noise is added on top of the blurred image. We use gaussian distribution with mean of 0 and a variance of 0.01 to generate the noise. The result image after both blur and noise is shown below.

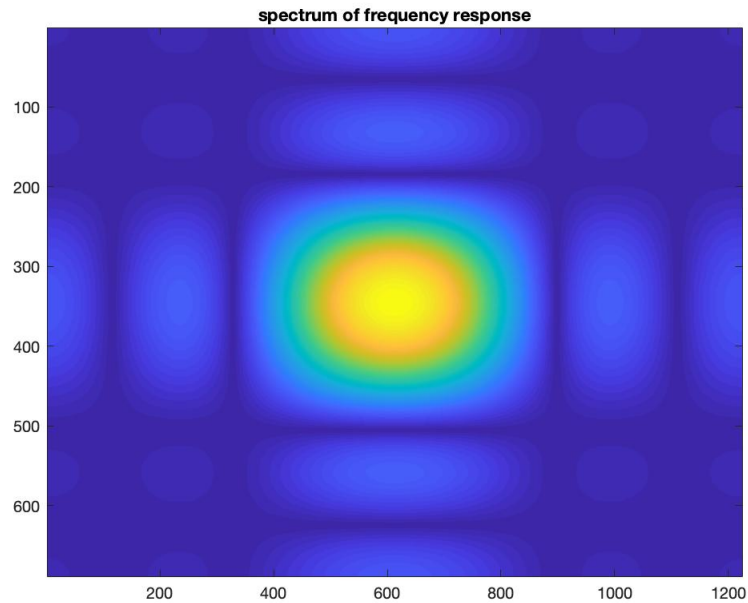
with blur and noise



3.2 Calculate the frequency transform of your blurring function h

Answer:

The FFT of the blurring function h is shown below. We know that the array multiplication in the frequency domain requires the frequency response and the FFT of the image to be of the same size, so we pad the blurring function to the size of F before FFT. The image of H also shows one property of the gaussian function. That is the FFT of gaussian is still gaussian and vice versa.

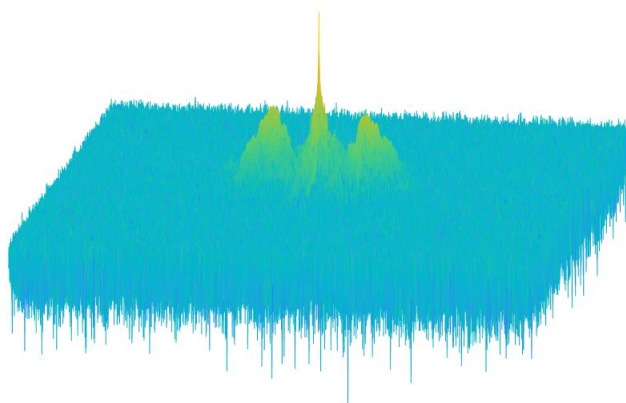


3.3 Calculate the 2D FFT of the degraded image(using an inbuilt function like `fft2`, `numpy.fft`, `fft2`). Display the degraded image's power spectrum in 1D, 2D, 3D and comment on it. What does it reveal about the blurring and noise?

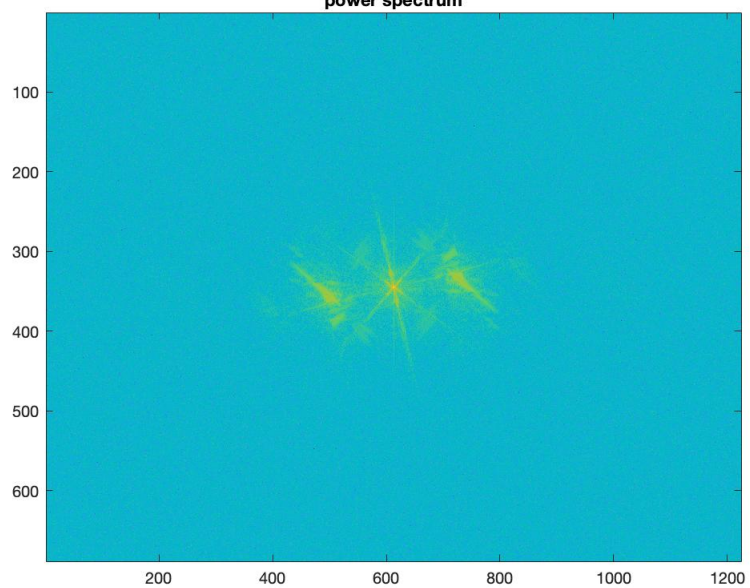
Answer:

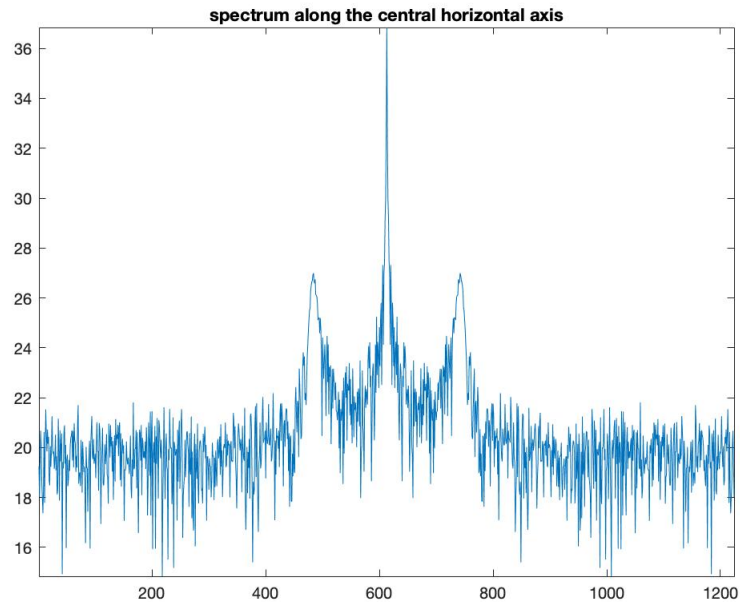
The power spectrum of 1D, 2D and 3D are plotted below. We see that due to the blurring, the high frequency components are reduced. This is as expected, because the blurring is essentially smoothing the rate of change which corresponds to high frequencies. Due to the noise, we should expect to see more high frequency values since the noise corresponds to abrupt change of intensity. However, because of the blurring, the effect cancels out so it is not so obvious in the image.

3d power



power spectrum





3.4 Find a way to remove the blurring in the frequency domain, using the inverse filtering methods we discussed in class. Then show the denoised image (in space) and its power spectrum. Discuss your approach and results.

Answer:

We use in the inverse filtering. The formular for inverse filtering is $\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)}$. The major drawback is that in the presense of noise, and when $H(u, v)$ is small, the term will be dominated by the Noise, since the previous formular is also equivalent in form to $\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$. One quick solution is to only filter frequencies that are near the origin. As we saw from the previous question, the terms near $f(0,0)$ has the biggest power/energy, thus we can limit the influence of noise by the big term in the denominator.

For the cutoff values, we experiment with different values in the range from 100 to 200 with an interval of 10, as value beyond 200 results in completely noise covered image, any value below 100 causes too much blurring.

Another thing is to consider is how to transition after specifying the cutoff value. The simplest is to use the ideal filter, i.e., everything beyond the cutoff is 0 by default. Another way is to use another smoothing filter. In this case, the butterworth filter with an order of 10 is chosen as a second pass filtering after the inverse filter, or we can view the product of two to be the filter in use. For the ideal version, the best cutoff value is found to be 150, and for the butterworth version, the best value is found to be 140. Both results are shown below. The difference is small. The second image gives us less noise, while the first image looks more sharp. They all do a relatively good job at recovering the original image.

restored image with cutoff = 150



restored image with cutoff = 140 using butterworth smoothing



The power spectrum of both recovered images are shown below. For the first image, we see that the frequency is cutoff within the specified radius, while for the second image, the values have a smooth transition at the cutoff radius, as expected from the general shape of the corresponding filter in use.

