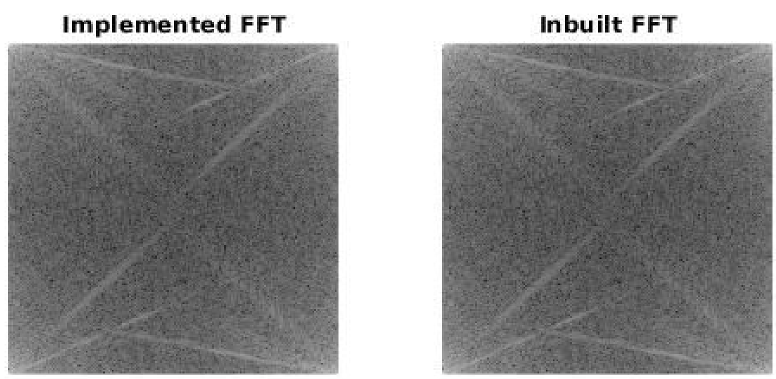
DIP Assignment 3

Report

Siddharth Gaur (20171198)

Q1. The implemented 2D FFT is applied on ‘cameraman.tif’ and also apply the inbuilt fft function, ‘fft2’ and get the following output:

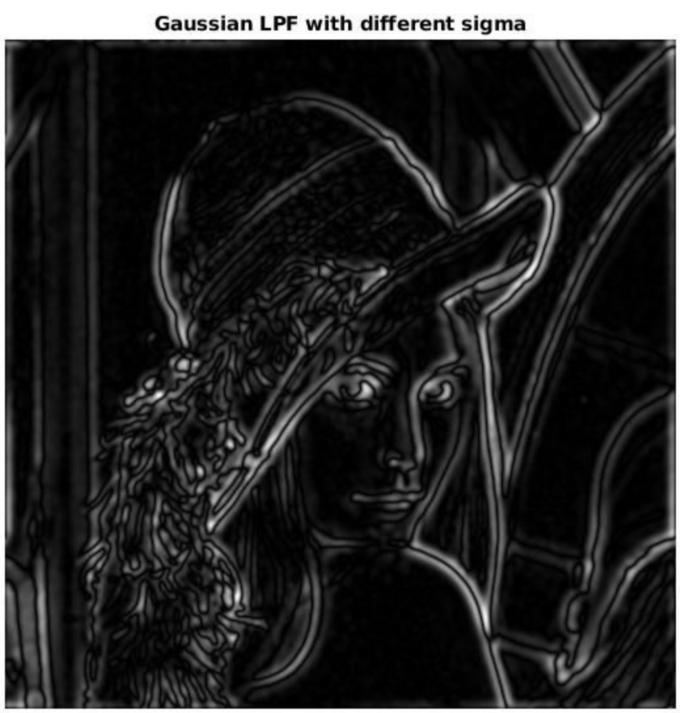


Q2. 1) Ideal, Butterworth and Gaussian low pass filters with D_o = 30 and n = 2(for Butterworth)

Ideal Low Pass Filter

Butterworth Filter

2) When the Gaussian LPF is applied with two different values of sigma(D_o), it acts as a bandpass filter as it only passes the values which lie between [D_{o1},D_{o2}] i.e. the two values of sigma.



Q3. The solution is as follows:

3 Image 1

$g_1 = f_1 + h_2 * f_2 \longrightarrow g_1(x,y) = f_1(x,y) + (h_2 * f_2)(x,y)$

Image 2

$g_2 = f_2 + h_1 * f_1 \longrightarrow g_2(x,y) = f_2(x,y) + (h_1 * f_1)(x,y)$

Applying fourier transform to both imges :-

$G_1(u,v) = F_1(u,v) + H_2(u,v) F_2(u,v)$ —— ①

$G_2(u,v) = F_2(u,v) + H_1(u,v) F_1(u,v)$ —— ②

from eq ②

$F_2(u,v) = G_2(u,v) - H_1(u,v) F_1(u,v)$

putting value of F_2 in eq ①

$G_1(u,v) = F_1(u,v) + H_2(u,v)\{G_2(u,v) - H_1(u,v) F_1(u,v)\}$

$G_1(u,v) = F_1(u,v) + H_2(u,v) G_2(u,v) - H_2(u,v) H_1(u,v) F_1(u,v)$

$G_1(u,v) = F_1(1 - H_1(u,v) H_2(u,v)) + H_2(u,v) G_2(u,v)$

$F_1(u,v)(1 - H_1(u,v) H_2(u,v)) = G_1(u,v) - H_2(u,v) G_2(u,v)$

$$F_1(u,v) = \frac{G_1(u,v) - H_2(u,v) G_2(u,v)}{1 - H_1(u,v) H_2(u,v)} = \hat{F}_1(u,v)$$

Similarly we can find $F_2(u,v)$ by putting F_1 in eq ① we get,

$$F_2(u,v) = \frac{G_2(u,v) - H_1(u,v) G_1(u,v)}{1 - H_1(u,v) H_2(u,v)} = \hat{F}_2(u,v)$$

now, $f_1(x,y)$ & $f_2(x,y)$ can be obtained by taking inverse fourier transform

$f_1(x,y) = F_1^{-1}(\hat{F}_1(u,v))$

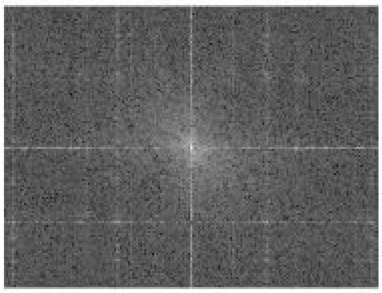
$f_2(x,y) = F_2^{-1}(\hat{F}_2(u,v))$

Since H_1 & H_2 are LPF, so when $H_1(u,v) H_2(u,v) = 1$

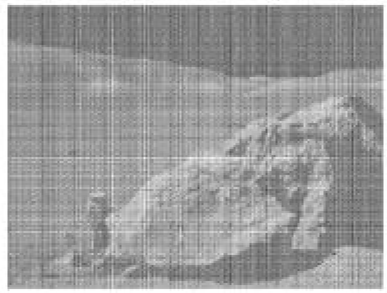
Our solution is undefined.

Q4. For the given image, Notch Filter is used to denoise it.

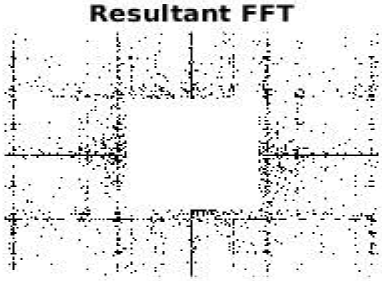
Original FFT



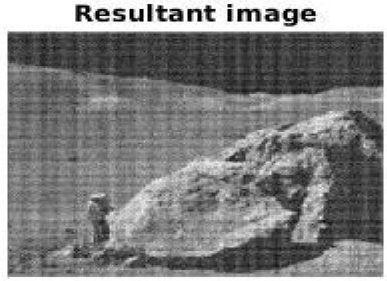
Original image



Resultant FFT

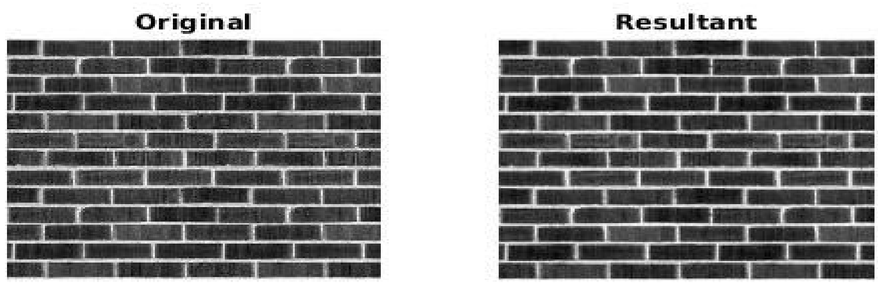


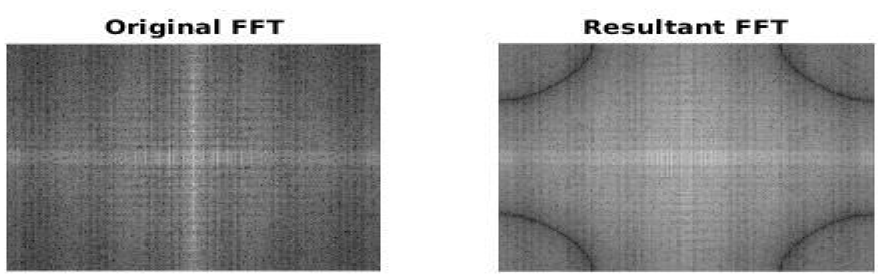
Resultant image

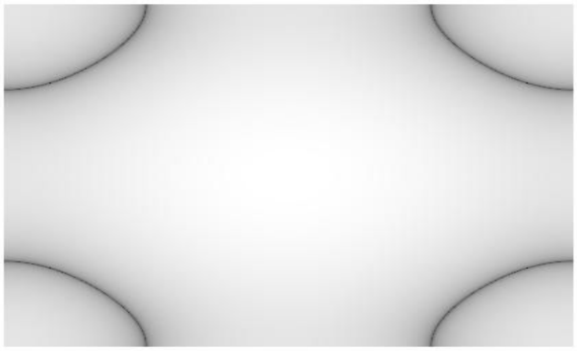


Repetitive noise in an image is seen as a bright peak somewhere other than the origin. We can suppress such noise effectively by carefully erasing the peaks. One way to do this is to use a notch filter to simply remove that frequency from the picture. In general notch filtering is an ad hoc procedure requiring to determine what frequencies need to be removed to clean up the signal.

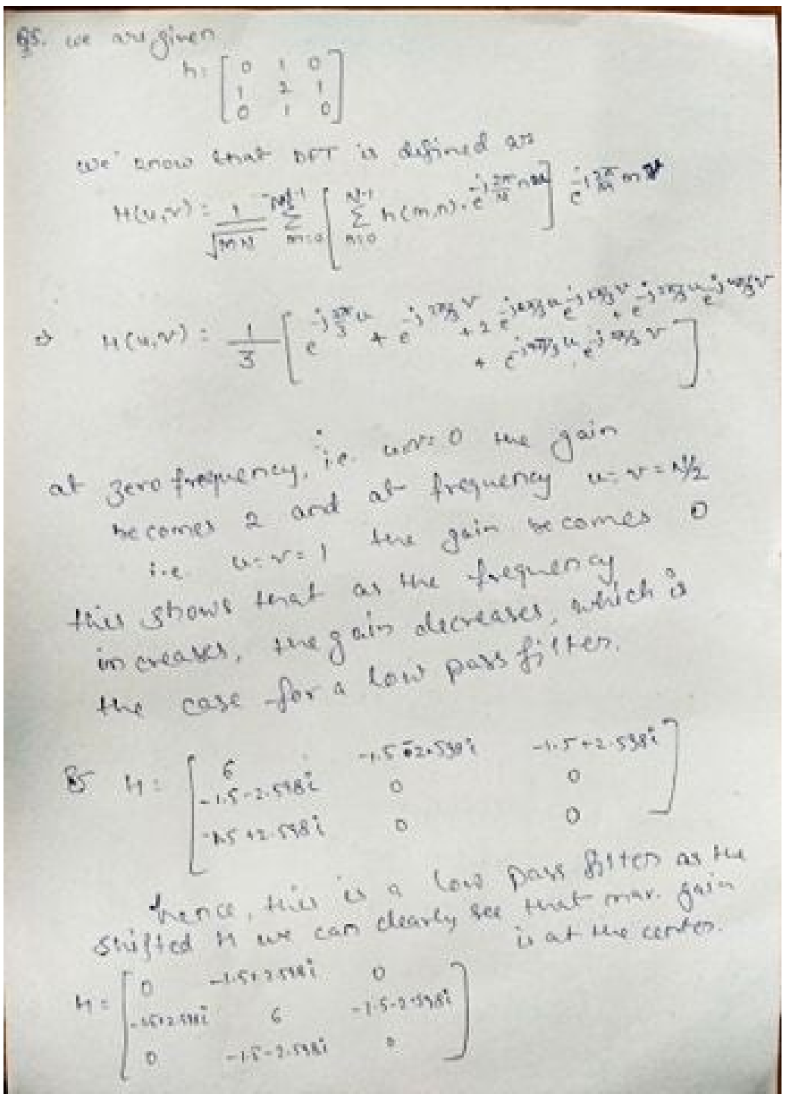
Q5.







This is the FFT of the filter that is given to us, the mathematical representation is as:



Q5. we are given

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

we know that DFT is defined as

$$H(u,v) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} h(m,n) \cdot e^{-j\frac{2\pi}{N} nv} \right] e^{-j\frac{2\pi}{M} mu}$$

$$\Rightarrow H(u,v) = \frac{1}{3} \left[e^{-j\frac{2\pi}{3}u} + e^{-j\frac{2\pi}{3}v} + 2 + e^{-j\frac{2\pi}{3}u} e^{-j\frac{2\pi}{3}v} + e^{-j\frac{2\pi}{3}u} e^{-j\frac{2\pi}{3}v} + e^{-j\frac{4\pi}{3}u} e^{-j\frac{4\pi}{3}v} \right]$$

at zero frequency, i.e. $u=v=0$ the gain becomes 2 and at frequency $u=v=N/2$ i.e $u=v=1$ the gain becomes 0

this shows that as the frequency increases, the gain decreases, which is the case for a low pass filter.

$$H = \begin{bmatrix} 6 & -1.5-2.598i & -1.5+2.598i \\ -1.5-2.598i & 0 & 0 \\ -1.5+2.598i & 0 & 0 \end{bmatrix}$$

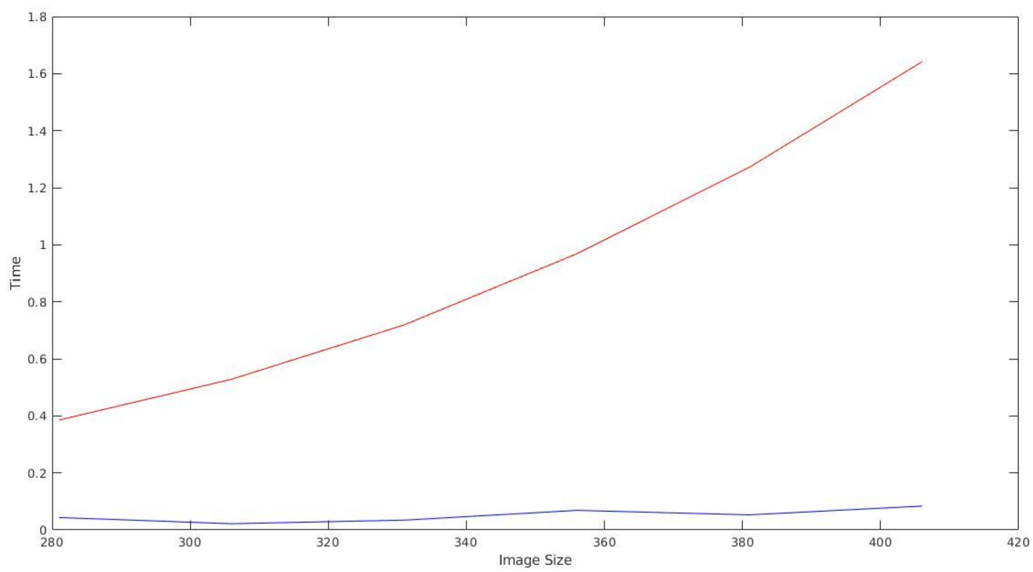
hence, this is a low pass filter as the shifted H we can clearly see that max. gain is at the center.

$$H = \begin{bmatrix} 0 & -1.5+2.598i & 0 \\ -1.5+2.598i & 6 & -1.5-2.598i \\ 0 & -1.5-2.598i & 0 \end{bmatrix}$$

Q6. 1) Two different images, ‘cameraman.tif’ and ‘bricks.jpg’ are used to verify the **Convolution Theorem**, i.e. DFT[f*h] = $F_Z H_Z$, to verify this, error between both the methods is calculated and it comes out to be negligible,
Error = 5.2659e-14 .

2) Observations:

- FFT is faster for bigger images as compared to convolution.
- As the size of image increases the difference between the times of fft and convolution increases.

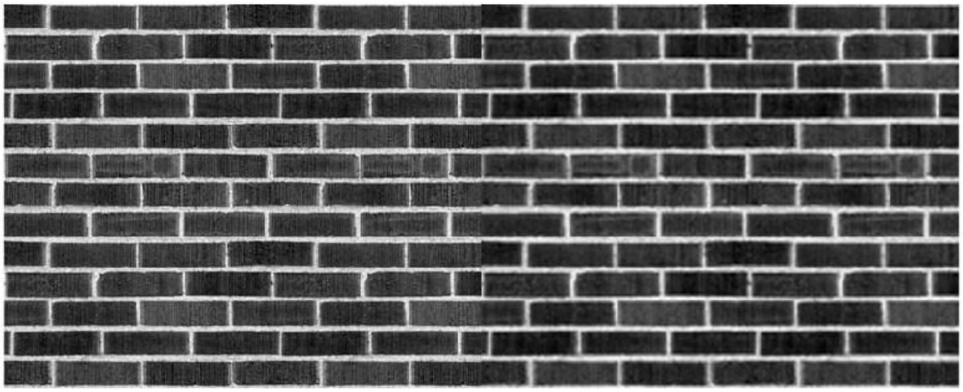


Here, Red and Blue represents Convolution and FFT respectively.

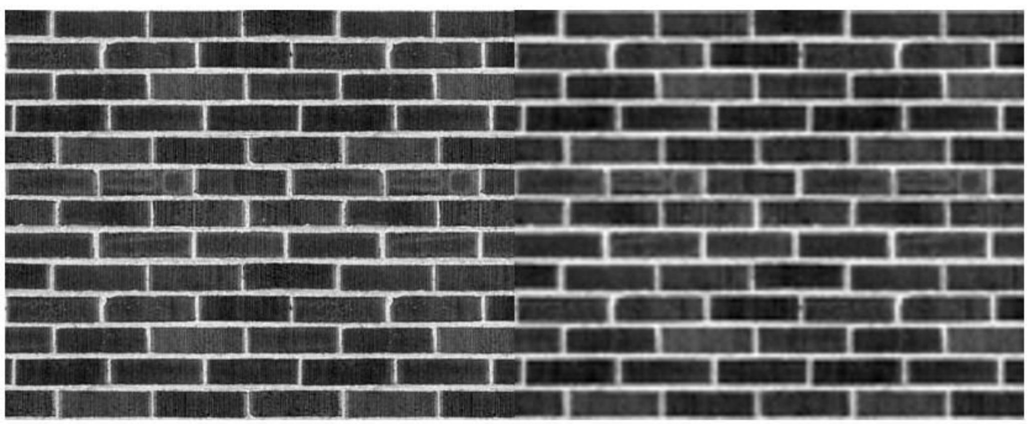
Q7. 1) After plotting the reconstructed images from the sampled images, it is clearly visible that after nx=ny=4, there is aliasing and that the original image cannot be recovered as the reconstructed image has aliasing and hence has lost some information. Since aliasing occurs at nx=ny=5, the nyquist rate for ‘bricks.jpg’ is **nx=ny=4**.

The output images are as follows:

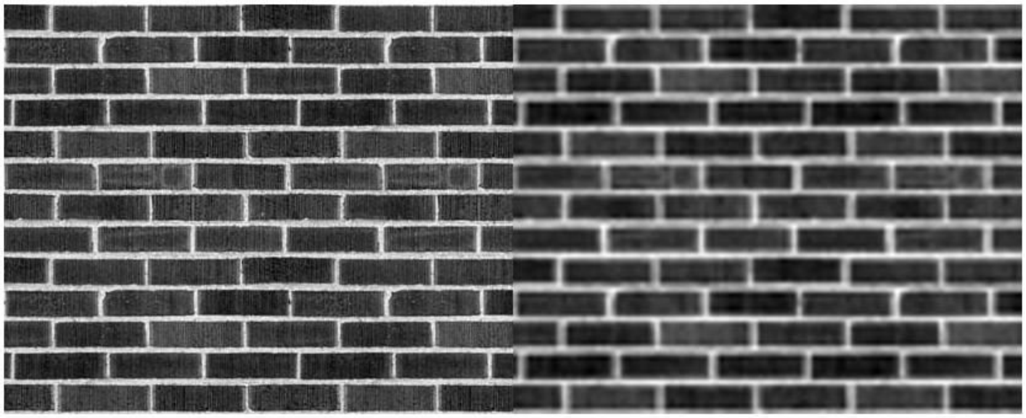
nx=ny=2



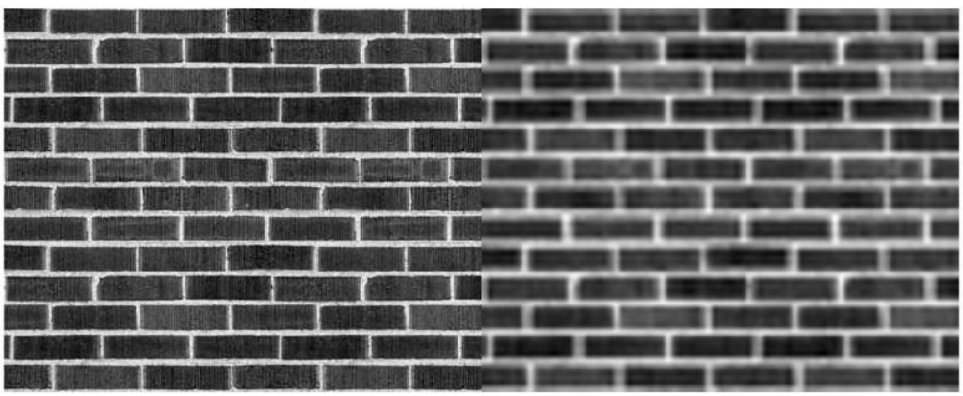
nx=ny=3



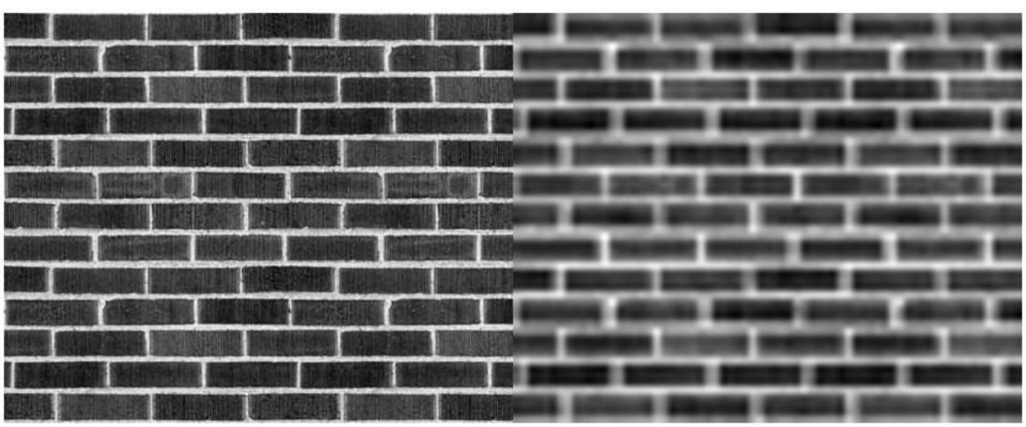
nx=ny=4 (Nyquist Rate)



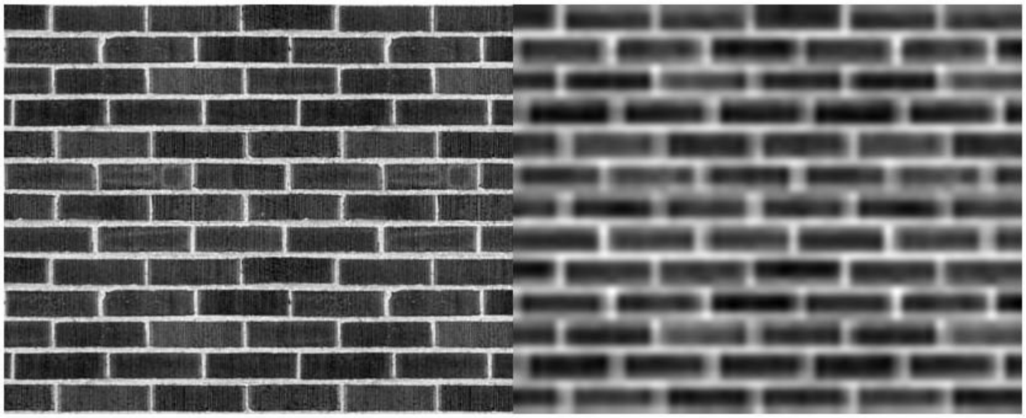
nx=ny=5 (1st aliasing occurs here)



nx=ny=6

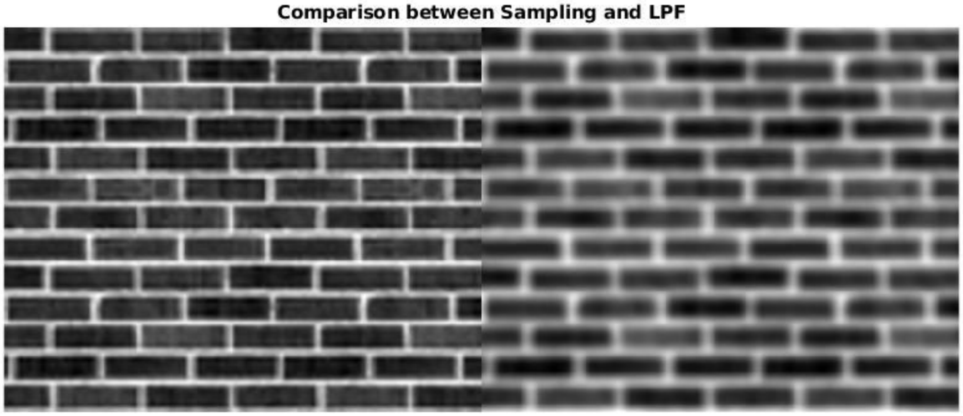


nx=ny=7

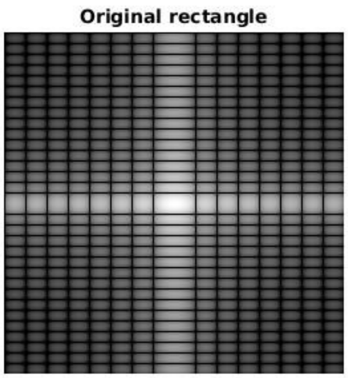


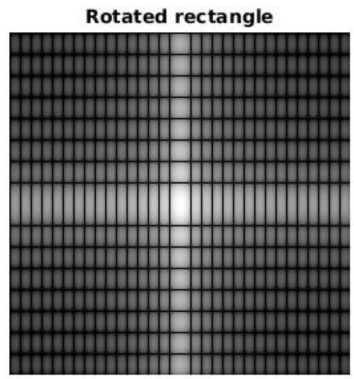
2) The image is blurred using a gaussian low pass filter with sigma = nyquist rate = 4. As alias result from sampling, the sampling effects cannot be avoided by blurring. Rather, blurring results in an approximation to first aliasing the picture to the current resolution, the proper resampling it to a still lower resolution and then properly upsampling the result to the current resolution.

Comparison between Sampling and LPF



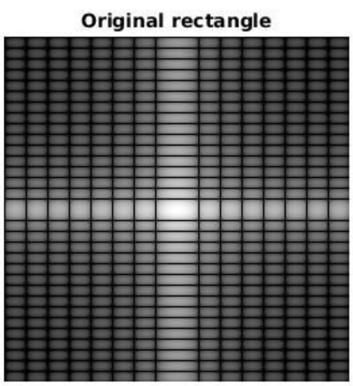
Q8. 1) The image ‘rectangle.jpg’ when rotated in spatial domain brings change in frequency domain too, the outputs are as follows:

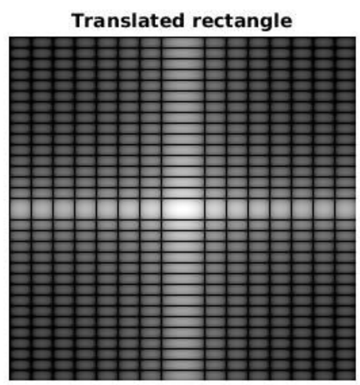




Here Θ = 90° so the fft is also rotated by 90°. Proof for this is as follows:

2) The image when translated by few pixels in the spatial domains gives rise to a phase shift in the frequency domain and since fftshift is being used in the code, the phase shift doesn’t affect the fft of the image. Here x and y are shifted by 20 and 50 respectively.





Proof for 1) :

2D Rotation of Image ('Rectangle.jpg')

Let us suppose image is $g(x,y)$

& $g(x,y) \longleftrightarrow G(f_x, f_y)$

Now, when we rotate the image by an angle θ, the corresponding function $g(x,y)$ changes as.

$$g(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$$

as the x & y coordinates changes when the θ changes.

Now, the new coordinates $(\hat{x}, \hat{y})$ can be written as

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

OR,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} \hat{x}\cos\theta - \hat{y}\sin\theta \\ \hat{x}\sin\theta + \hat{y}\cos\theta \end{bmatrix}$$

$$\therefore \mathcal{F}\{g(\hat{x},\hat{y})\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\hat{x},\hat{y}) e^{-j2\pi(xf_x + yf_y)} dx\, dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\hat{x},\hat{y}) e^{-j2\pi[\hat{x}(f_x\cos\theta + f_y\sin\theta) + \hat{y}(-f_x\sin\theta + f_y\cos\theta)]} dx\, dy$$

now $d\hat{x}\, d\hat{y} = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix} dx\, dy = dx\, dy$

$$\therefore \mathcal{F}\{g(\hat{x},\hat{y})\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\hat{x},\hat{y}) e^{-j\pi[\hat{x}(f_x\cos\theta + f_y\sin\theta) + \hat{y}(-f_x\sin\theta + f_y\cos\theta)]} d\hat{x}\, d\hat{y}$$

$$\Rightarrow \boxed{\mathcal{F}\{g(\hat{x},\hat{y})\} = G(f_x\cos\theta + f_y\sin\theta, -f_x\sin\theta + f_y\cos\theta)}$$

It means that rotation in spatial domain causes rotation by same angle in the frequency domain.

Proof for 2):

Translation of Image in Spatial Domain

Lets assume an image $g(x,y)$

& $g(x,y) \longleftrightarrow G(f_x, f_y)$

now, $g(x,y)$ is shifted in spatial domain by few pixels, i.e.

$g(x,y) \longrightarrow g(x-a, y-b)$

now, the fourier transform of this image,

$$\mathcal{F}\{g(x-a, y-b)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-a, y-b)\, e^{-j2\pi((x-a)f_x + (y-b)f_y)}\, dx\,dy$$

$$\Rightarrow \mathcal{F}\{g(x-a, y-b)\} = \cancel{e^{j2\pi(a+b)}}$$

$$= e^{j2\pi(af_x + bf_y)}\, G(f_x, f_y)$$

$$\Rightarrow g(x,y) \longleftrightarrow G(f_x, f_y)$$

$$\therefore g(x-a, y-b) \longleftrightarrow e^{j2\pi(af_x + bf_y)}\, G(f_x, f_y)$$

So, translation in spatial domain by a few pixels gives rise to phase change in the frequency domain,