Tianxiao Zhao – tzh@kth.se

DD2423 – Lab 1

1.3 Basic functions

Question 1

The following 6 pictures are the results of the coordinates p and q set to (5, 9), (9, 5), (17, 9), (17, 121), (5, 1) and (125, 1) respectively.

From these results, we can draw conclusions below:

1. All 6 pictures have the same amplitude;
2. The plots of the real part and imagine part is almost the same, only with a phase difference of $\pi/2$. Using inverse Fourier transform, we can get:

$$f(m,n) = \frac{1}{\sqrt{MN}}\sum\sum \hat{f}(u,v)e^{2\pi i\left(\frac{mu}{M}+\frac{nv}{N}\right)}$$

$$= \frac{1}{\sqrt{MN}}\sum\sum \hat{f}(u,v)[\cos\left(2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right) + isin\left(2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)\right)]$$

In this case, only when $u = p$ and $v = q$, $\hat{f}(u,v) = 1$ hold. So, all the plots with a single point of value 1 have the same amplitude. And the real and imagine part share the same ω, and exist a pi/2 difference in phase;

3. The direction of the waveform is the same with the direction of the line between (p, q) and the origin (0, 0) in the frequency domain. The further the distance from (p, q) to the origin in frequency domain, the larger the frequency of the wave in spatial domain, which also means that the waveform has a smaller wavelength.

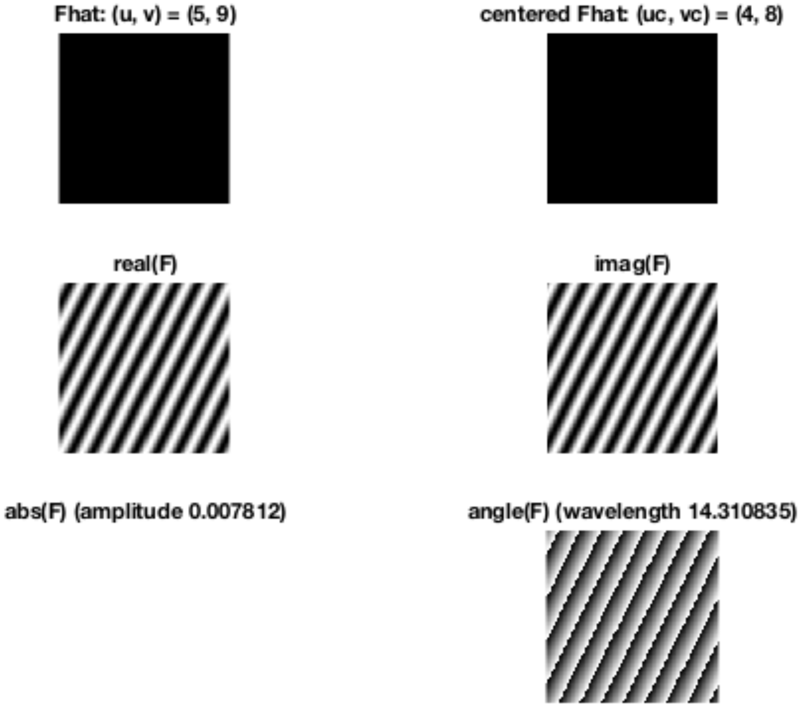

Figure 1: (p, q) = (5, 9)

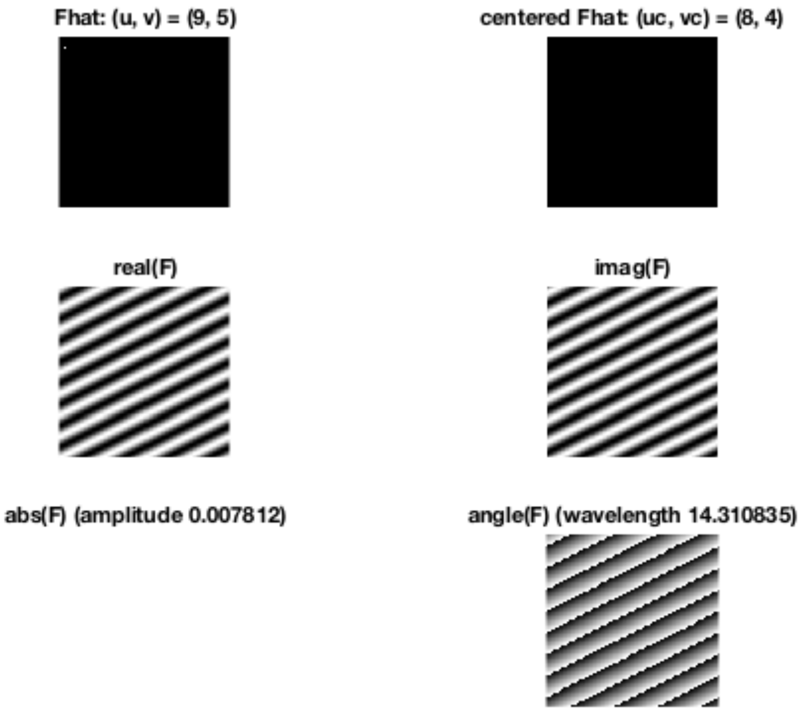

Figure 2: (p, q) = (9, 5)

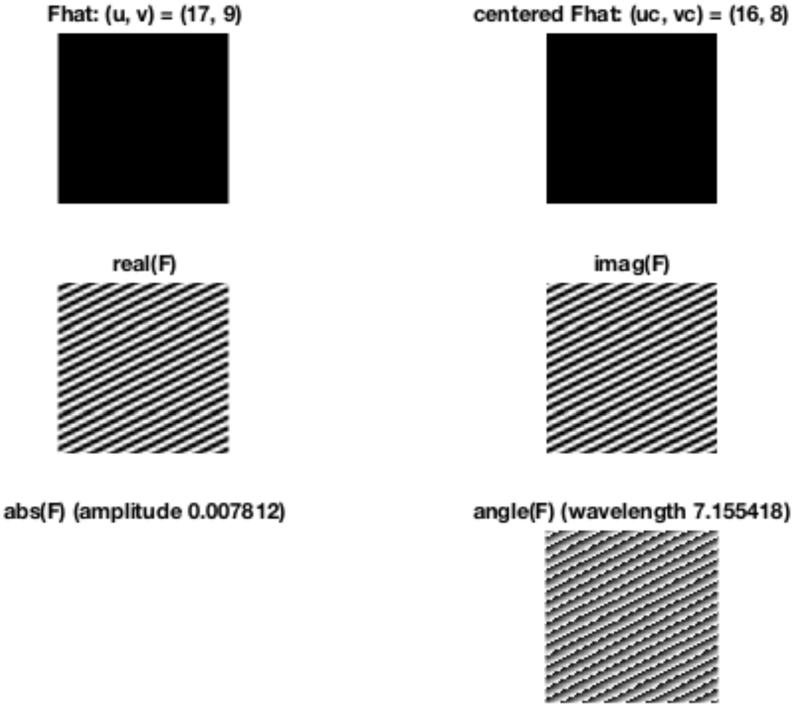

Figure 3: (p, q) = (17, 9)

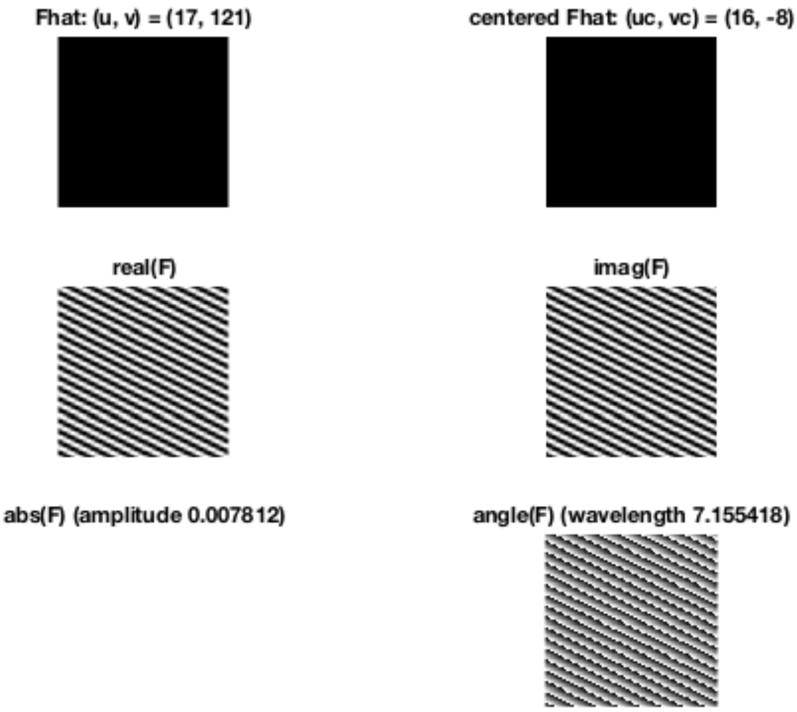

Figure 4: (p, q) = (17, 121)

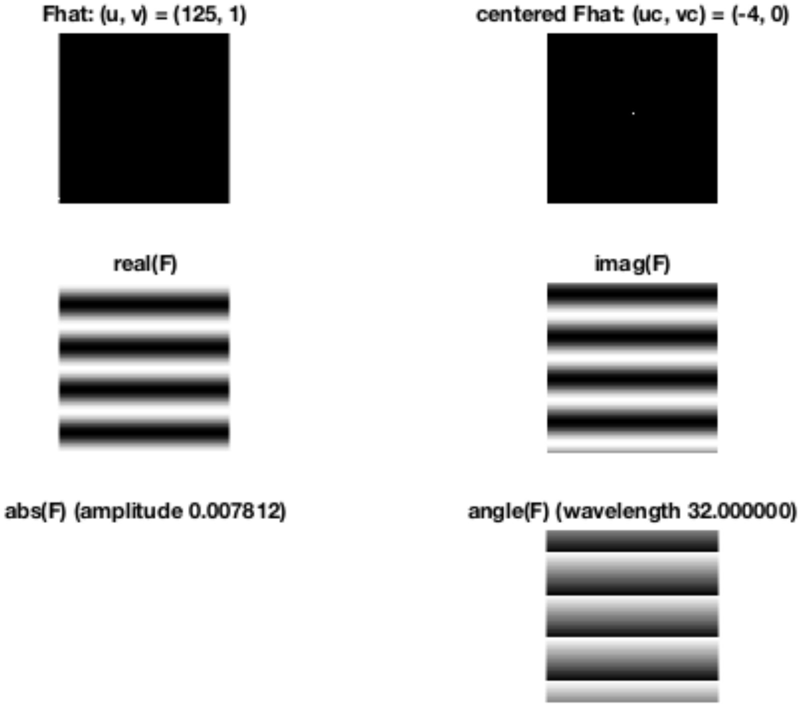

Figure 5: (p, q) = (125, 1)

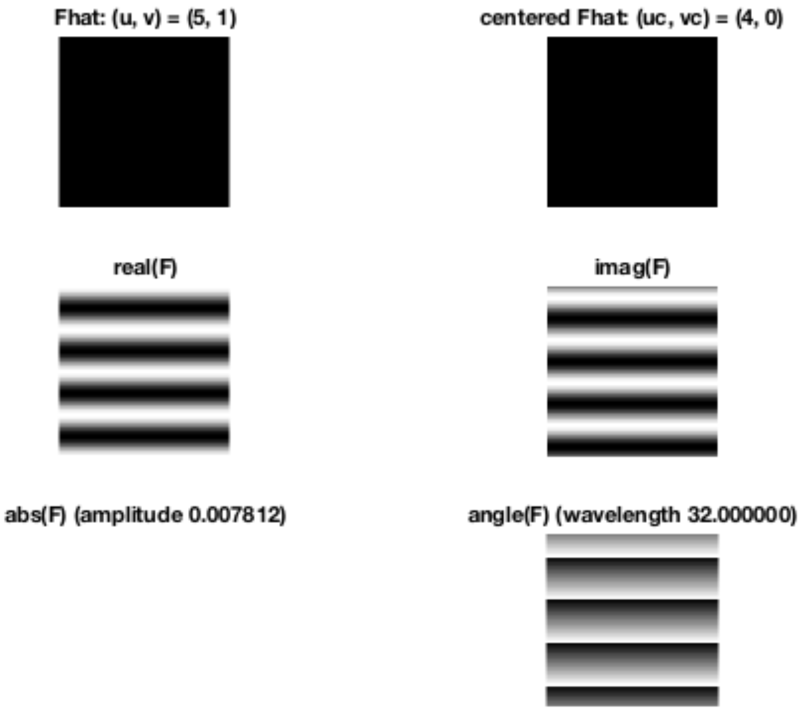

Figure 6: (p, q) = (5, 1)

Question 2

From question 1, we have:

$$f(m,n) = \frac{1}{\sqrt{MN}} \sum \sum \hat{f}(u,v)[\cos\left(2\pi\left(\frac{mu}{M} + \frac{nv}{N}\right)\right) + isin\left(2\pi\left(\frac{mu}{M} + \frac{nv}{N}\right)\right)]$$

For example, when $(\mathrm{p}, \mathrm{q}) = (5,9)$, the real part of the waveform in spatial domain is shown below:

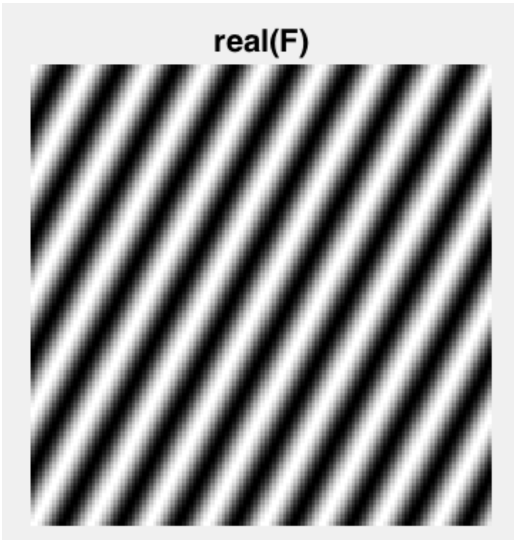

Figure 7: real part of the waveform (2D)

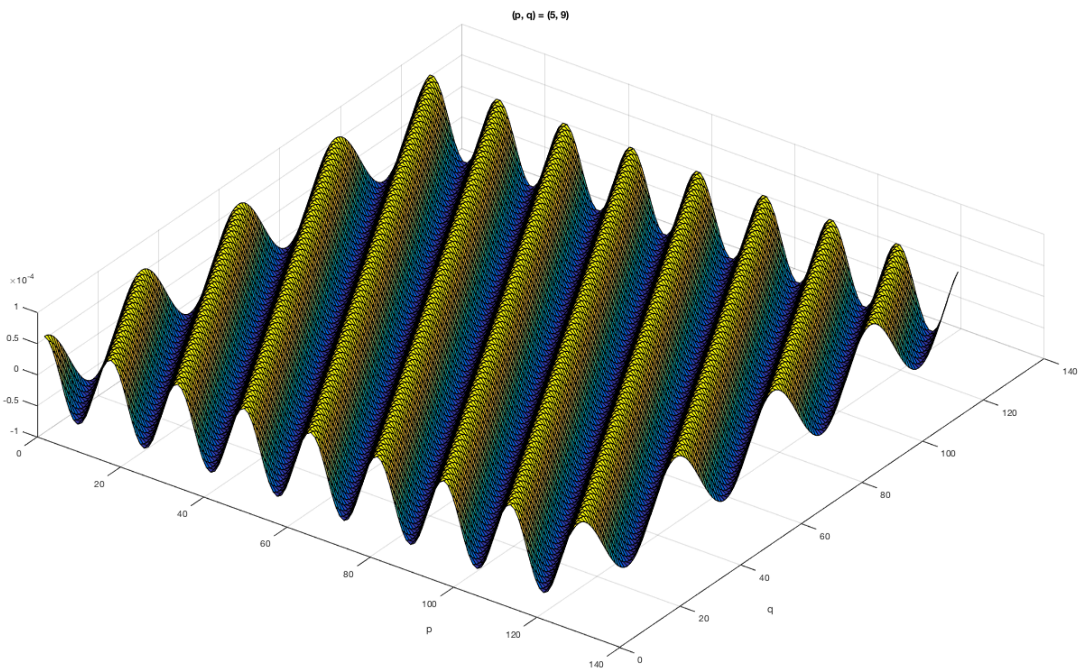

Figure 8: real part of the waveform (3D)

Question 3

From inverse Fourier transform, we can derive:

$$f(m,n) = \frac{1}{\sqrt{MN}} \sum \sum \hat{f}(u,v)[\cos\left(2\pi\left(\frac{mu}{M} + \frac{nv}{N}\right)\right) + isin\left(2\pi\left(\frac{mu}{M} + \frac{nv}{N}\right)\right)]$$

The amplitude should satisfy the relation below:

$$A = \frac{1}{\sqrt{MN}} \cdot \max\left(\hat{f}(u,v)\right) = \frac{1}{\sqrt{MN}} = \frac{1}{128}$$

The code (Bold) is complemented as shown below:

```
...
w1 = 2 * pi * uc / sz;
w2 = 2 * pi * vc / sz;
wavelength = 2 * pi / sqrt(w1^2 + w2^2); % wavelength
amplitude = max(Fhat(:)) / sz; % amplitude
...
```

Question 4

From the lecture notes, we can get:

$$\lambda = \frac{2\pi}{||\omega||} = \frac{2\pi}{\sqrt{\omega 1^2 + \omega 2^2}}$$

According to the inverse Fourier transform, we have $\omega_1 = 2\pi * \frac{u}{M}$ and $\omega_2 = 2\pi * \frac{v}{N}$.

Here, $u = uc$, $v = vc$, $M = N = sz$. The code (Bold) is complemented as shown below:

```
...
w1 = 2 * pi * uc / sz;
w2 = 2 * pi * vc / sz;
wavelength = 2 * pi / sqrt(w1^2 + w2^2); % wavelength
amplitude = Fhat(u, v) / sz^2; % amplitude
...
```

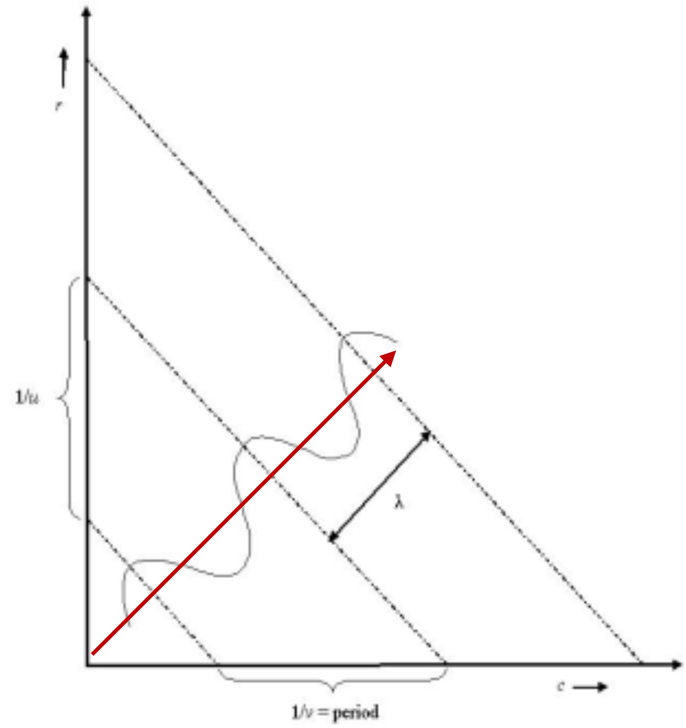

Figure 9: direction of the waveform

As is depicted above, the direction of the waveform can be determined by line between the point (p, q) and the origin (0, 0).

Question 5

The discrete Fourier transform of the image is periodic with interval N. In the frequency domain, an image is repeated periodically in the 2D plain. The figure below shows how the image repeats itself, and the pixel value should satisfy the relation that $(N+x,\ N+y) = (N-x,\ N-y)$. This means that the regions of the original image exchange their position diagonally after the function fftshift is used.

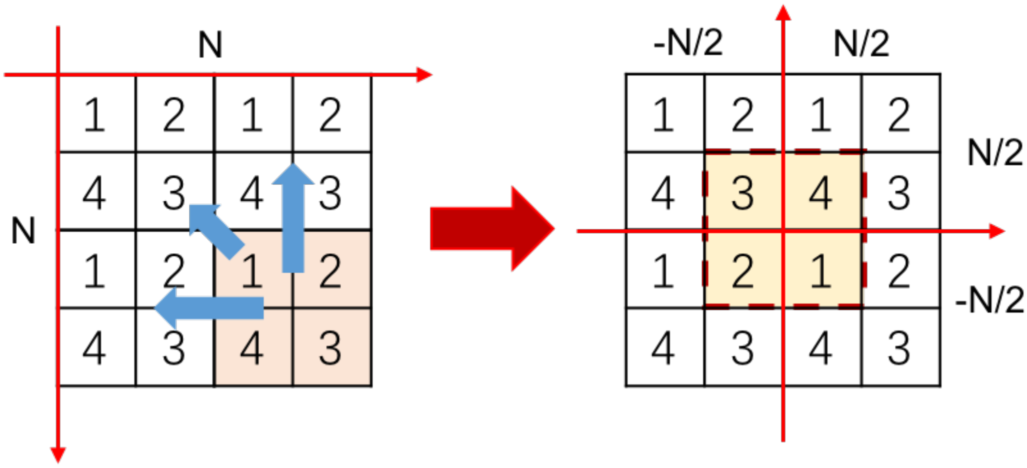

Figure 10: function fftshift

Question 6

The original image in the frequency domain visualizes the frequency within the interval $[0, 2\pi]$. The lines of code in the script computes the coordinates of (p, q) after the function fftshift( ) is applied. The function fftshift( ) shows the image of the frequency within the interval $[-\pi, \pi]$.

1.4 Linearity

After running the code, we get the following three figures. The first one shows the image of F, G and H in spatial domain. The second one shows the images in frequency domain and the one after function fftshift( ).

Here, we find after 'fftshift' command, the position of these white lines change. This can be easily explained since fftshift function helps to move the zero-frequency component to the center of the image in frequency domain.

The third one compares the fourier images and the histograms before and after logarithm is applied.

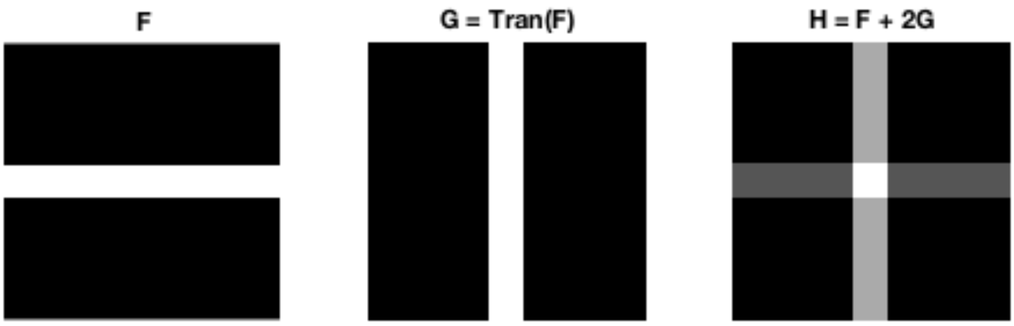

Figure 11： image in spatial domain

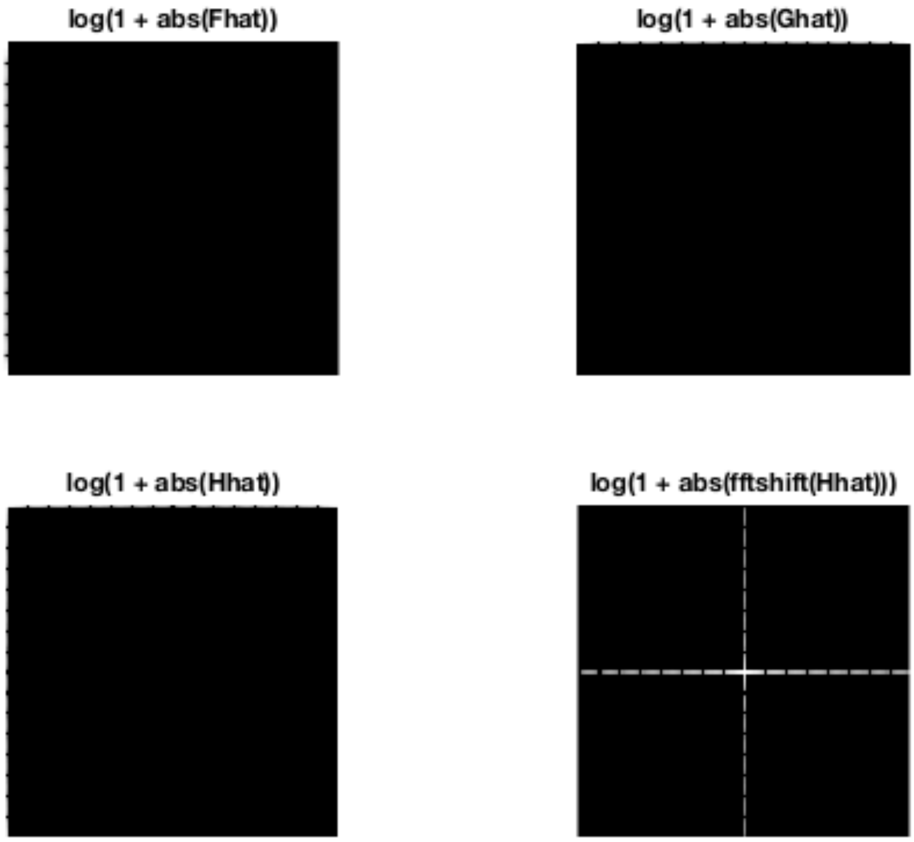

Figure 12: image in frequency domain

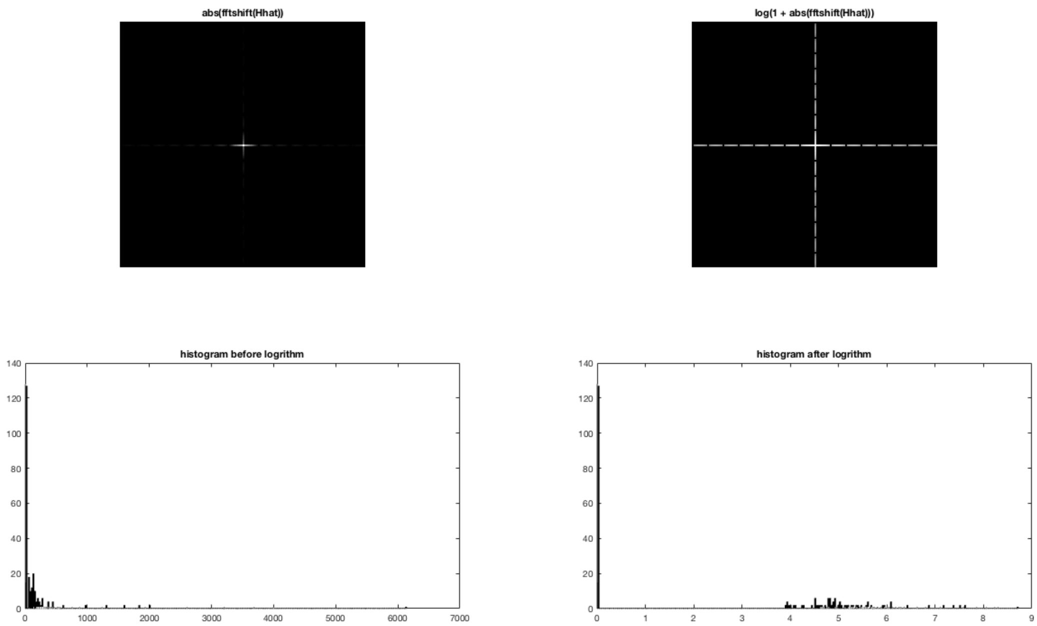

Figure 13: histogram before and after logarithm

Question 7

The Fourier transform has the following form:

$$\hat{F}(m,n) = \frac{1}{N}\sum\sum f(u,v)e^{-2\pi i\left(\frac{mu+nv}{N}\right)}$$

Let's take **F** for example. $f(u,v) = 1$ holds when $57 < u < 73$, otherwise $f(u,v) = 0$. So, we can compute the Fourier transform based on separability and simplify it with the property of Kronecker delta function:

$$\hat{F}(m,n) = \frac{1}{N}\sum_{u=56}^{72} e^{-2\pi i\left(\frac{mu}{N}\right)}\sum_{v=0}^{N-1} e^{-2\pi i\left(\frac{nv}{N}\right)} = \sum_{u=56}^{72} e^{-2\pi i\left(\frac{mu}{N}\right)} \cdot \delta(n)$$

Then, $\hat{F}(m,n)$ is non-zero value only when $n = 0$. This is the reason why the Fourier spectra of F concentrated to the left border. And this could also explain why the Fourier spectra of G is on the upper border.

Question 8

As is shown in Figure 13, the distribution of the pixels' values concentrate on a lower range before logarithm is applied. Since logarithm transformation could expand the range of low pixel values and compress the range of high pixel values, it can redistribute the pixel values on the low range and enhance contrast in the image of frequency domain. After logarithm is used, the detail in the image becomes more visible, and the dynamic range of the pixel value become much smaller, see Figure 13.

Question 9

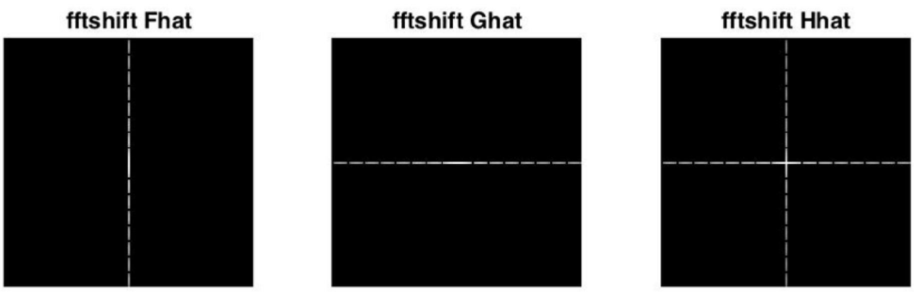

Figure 14 image in frequency domain

From this picture, we can easily conclude that the last figure is the linear combination of the first and the second. According to the definition, we have H = F + 2 * G in spatial domain. So, we can draw this conclusion:

$$F[af_1(m,n) + bf_2(m,n)] = a\hat{f}_1(m,n) + b\hat{f}_2(m,n)$$

1.5 Multiplication

The result is shown below:

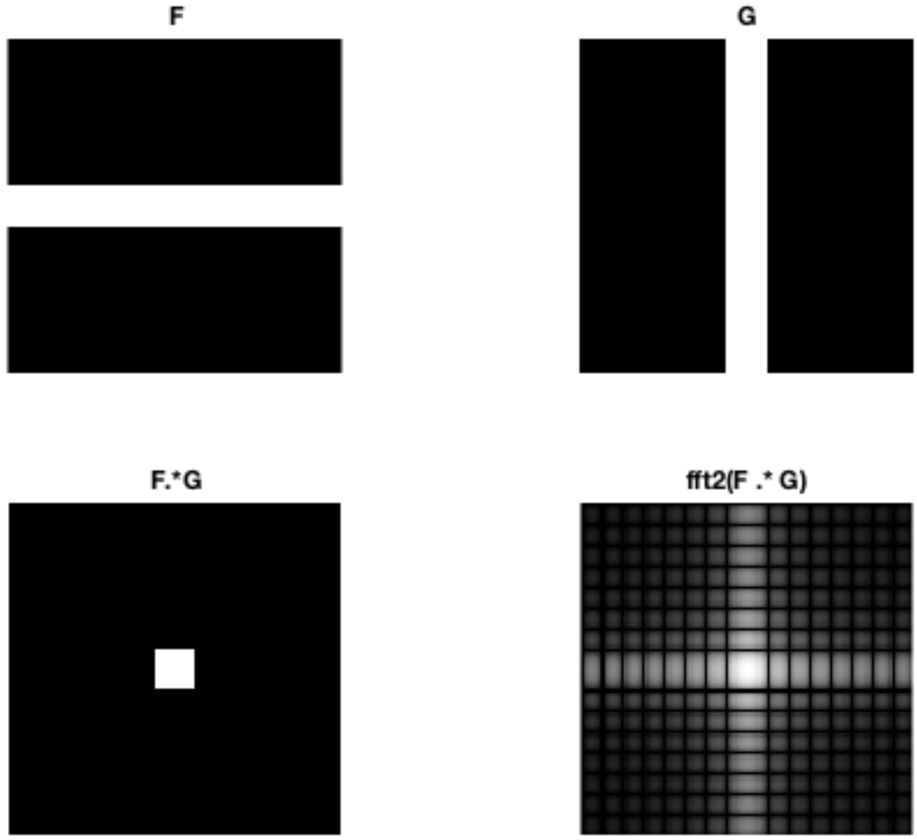

Figure 15 spatial multiplication and Fourier transformation

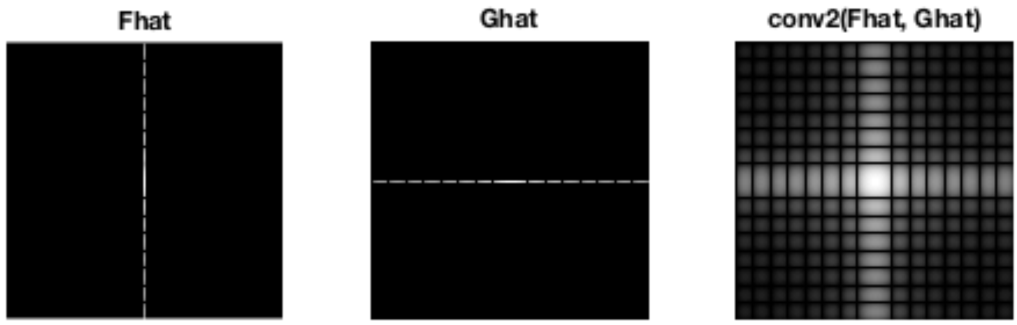

Figure 16 convolution of Fourier image

Question 10

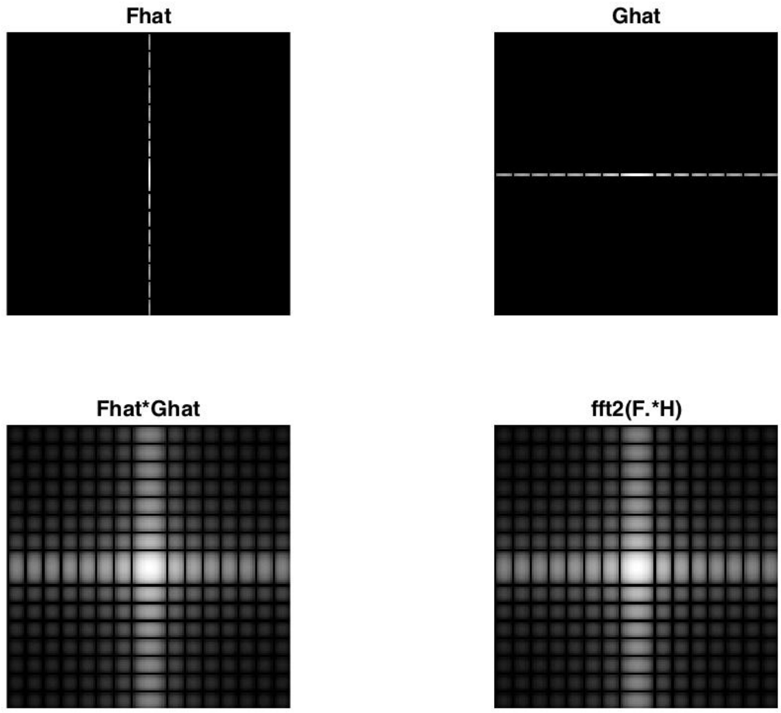

Figure 17: two ways of manipulation

From what we have learned, multiplication in the spatial domain is same as convolution in the Fourier domain:

$$F(fg) = F(f) * F(g)$$

The first image on the second row shows the result of the convolution of the Fourier transform of F and G; The second one is the Fourier transform of the multiplication of F and G. The two same results prove that the relation above should hold.

1.6 Scaling

Question 11

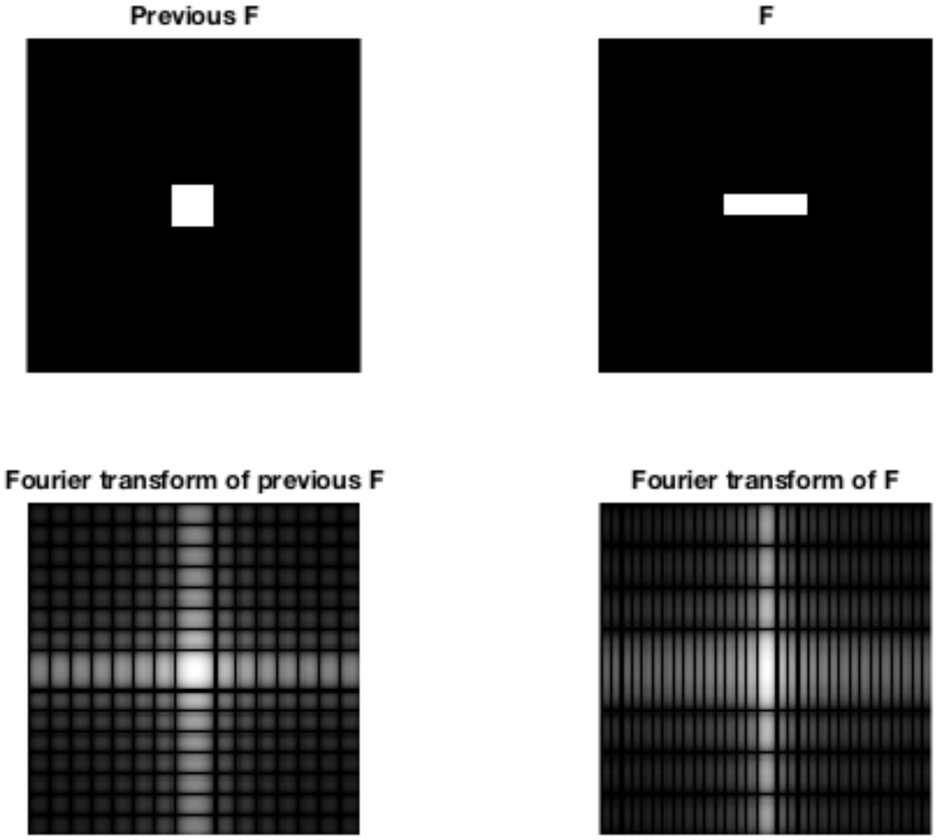

Figure 18: Fourier transform before and after scaling in spatial domain

The previous F is a 16×16 matrix of pixel values. F is an 8×32 matrix of pixel values. They have the relation $f_1(x, y) = f_2\left(2x, \frac{1}{2}y\right)$. Comparing their Fourier image, they should have following relation $\hat{f}_1(\omega_1, \omega_2) = \hat{f}_2\left(\frac{1}{2}\omega_1, 2\omega_2\right)$. Then, we could conclude that the expansion in spatial domain is same as compression in Fourier domain. Mathematically, if $g(x, y)$ is transformed from $f(x, y)$ by scaling: $g(x, y) = f(s_1 x, s_2 y)$, then the Fourier transform of $g(x, y)$ and $f(x, y)$ should have following relation:

$$\hat{g}(\omega_1, \omega_2) = \frac{1}{|s_1 s_2|}\hat{f}\left(\frac{\omega_1}{s_1}, \frac{\omega_2}{s_2}\right)$$

1.7 Rotation

Question 12

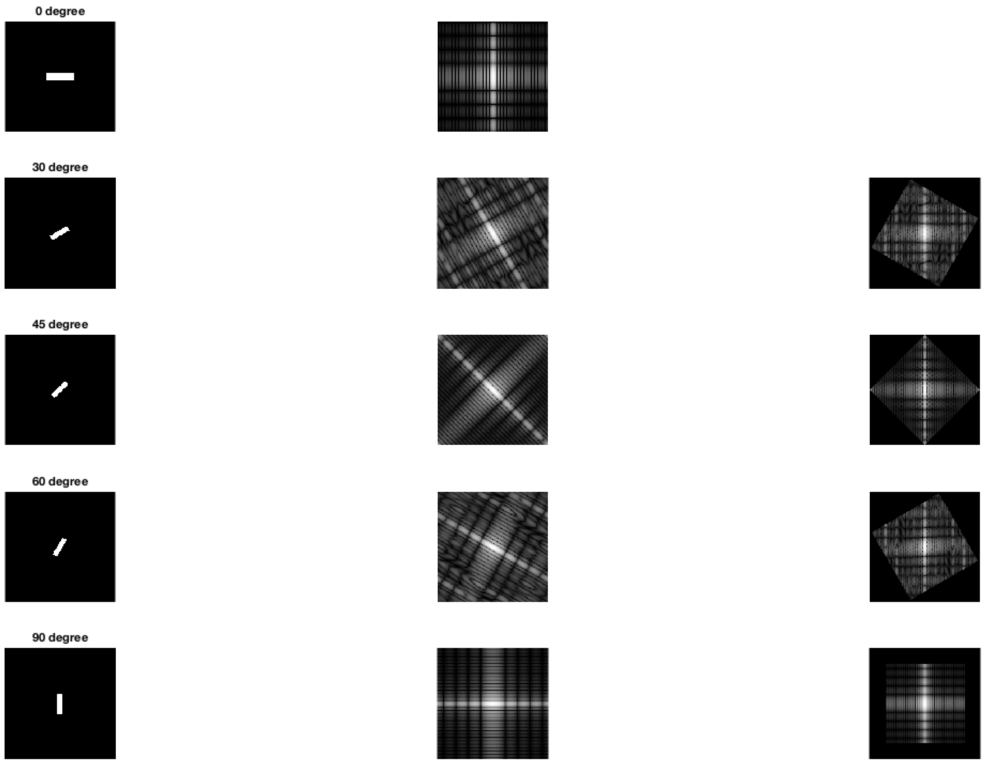

Figure 19: rotation of image in spatial and Fourier domain

Rotation of the original image rotates the image in Fourier domain by the same angle. Suppose a point (x, y) in the spatial domain is rotated by an angle θ, then its coordinates become $(x_1, y_1) = (x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$. Take the Fourier transform here:

$$\mathcal{F}(u,v) = \sum\sum f(x,y)\cdot e^{-\frac{2\pi i(xu+yv)}{N}}$$

$$= \sum\sum f(x_1\cos\theta - y_1\sin\theta, x_1\sin\theta + y_1\cos\theta)\cdot e^{-2\pi i((x_1\cos\theta - y_1\sin\theta)u + (x_1\sin\theta + y_1\cos\theta)v)/N}$$

$$= \sum\sum f(x_1\cos\theta - y_1\sin\theta, x_1\sin\theta + y_1\cos\theta)\cdot e^{-2\pi i(u\cos\theta + v\sin\theta)x_1 + (-u\sin\theta + v\cos\theta)y_1)/N}$$

Using $u_1 = u\cos\theta + v\sin\theta$, $v_1 = -u\sin\theta + v\cos\theta$, then the equation above could be simplified into:

$$\mathcal{F}(u_1, v_1) = \sum\sum f(x_1, y_1)\cdot e^{-\frac{2\pi i(x_1u_1 + y_1v_1)}{N}}$$

This means that the image in frequency domain is rotated by the same angle.

1.8 Information in Fourier phase and magnitude

Question 13

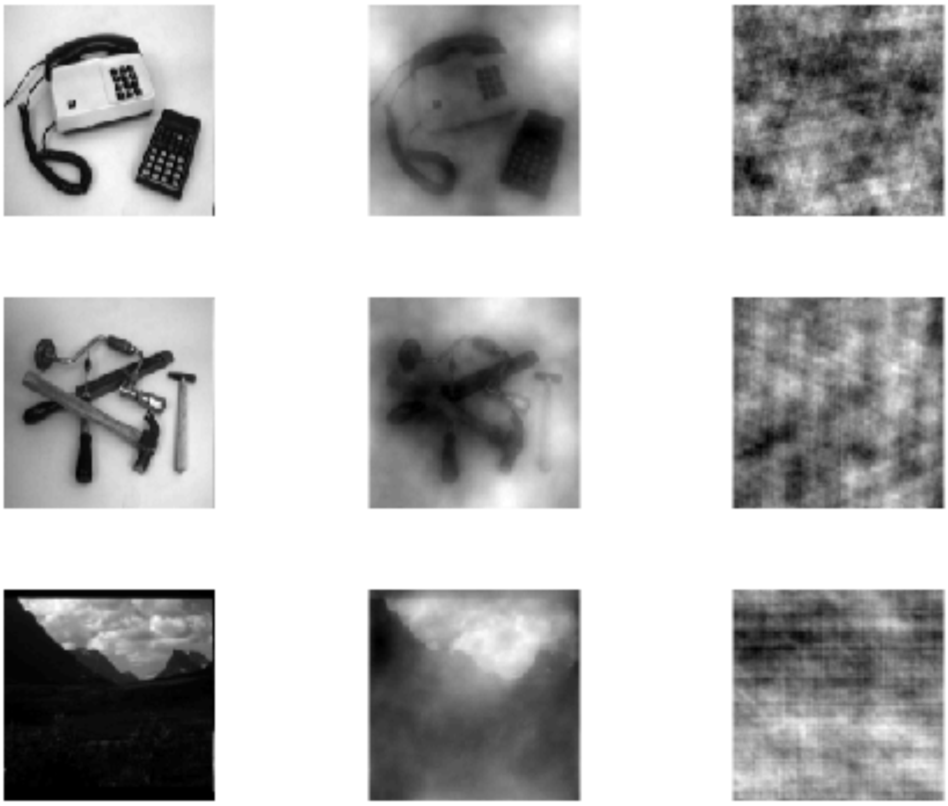

Figure 20: original image, image with phase information and image with magnitude information

The phase information is responsible for the position of the contour. Since the second column of images share the same phase with the first column, we could still distinguish the contours of the objects in the second one even if we change the magnitude; The magnitude is responsible for the brightness of the image. Because the third column keeps the magnitude unchanged and randomizes the phase information, we could still notice that the ratio of the brightness and darkness in the third column is approximately the same as the first column. But the lack of phase information causes that the brightness and darkness are located in different positions, which makes it hard to distinguish what it is in the image.

2.3 Filtering procedure

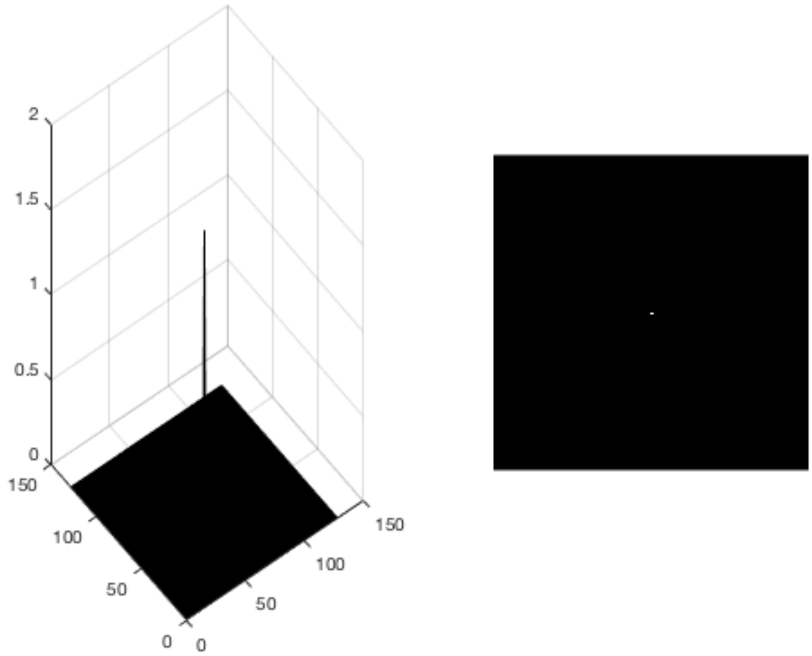

Figure 21: impulse response (t = 0.1)

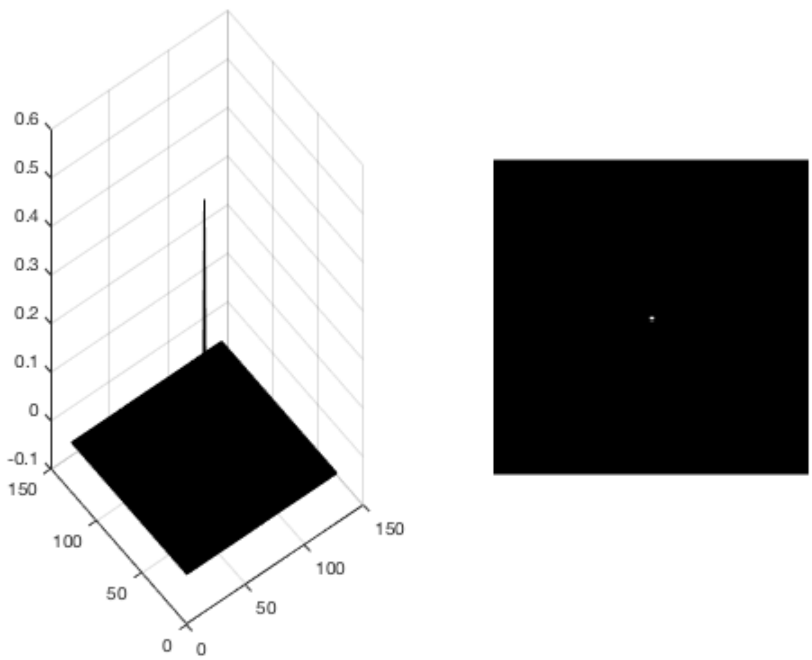

Figure 22 : impulse response (t = 0.3)

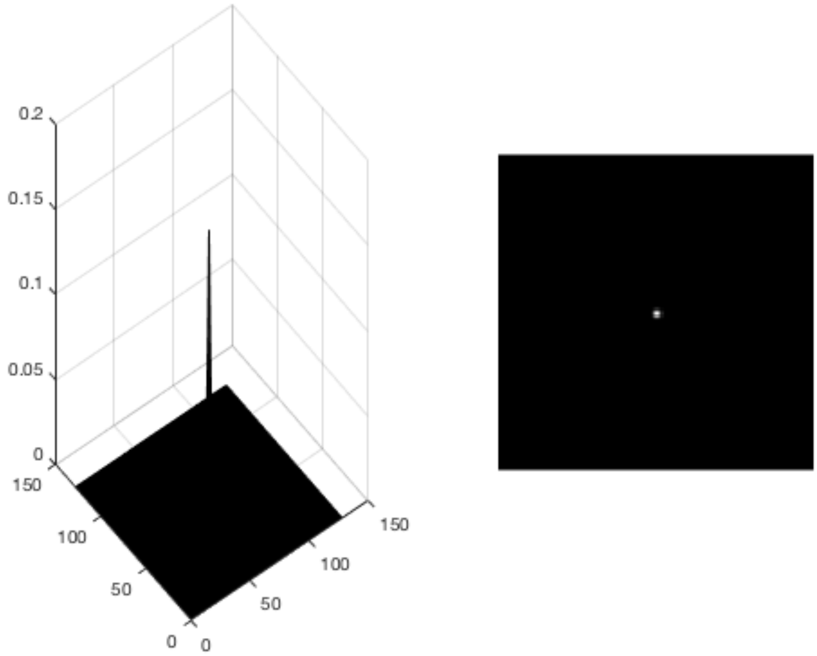

Figure 23: impulse response (t = 1.0)

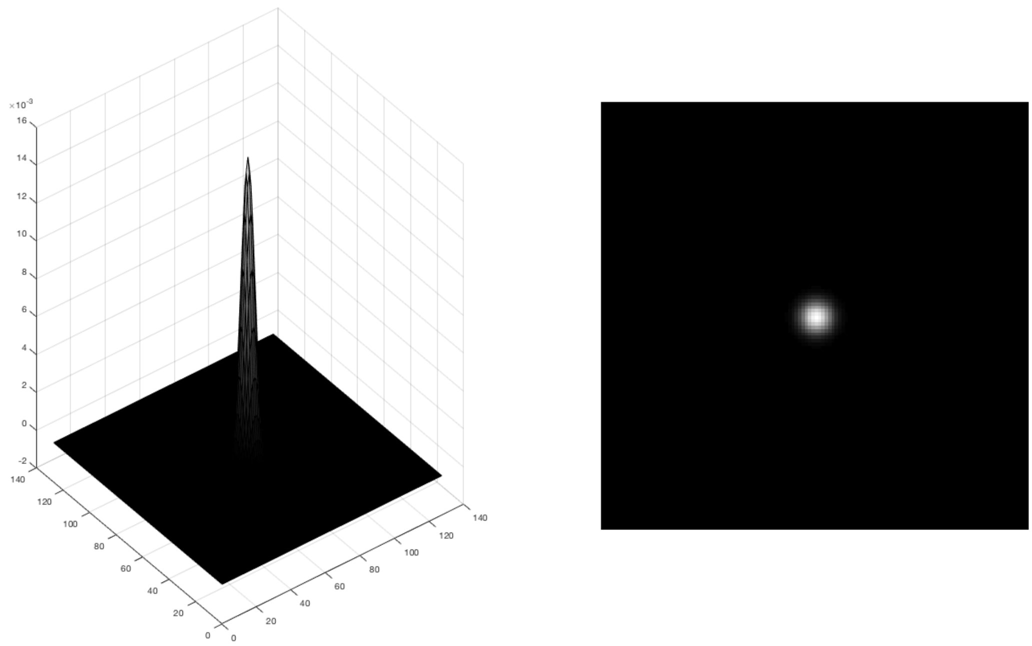

Figure 24: impulse response (t = 10.0)

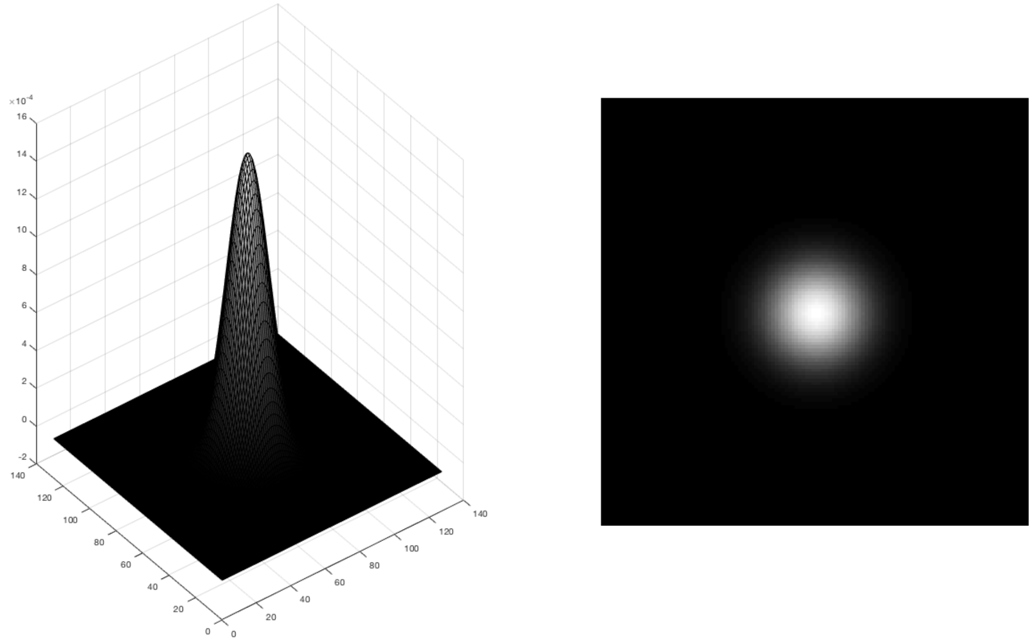

Figure 25: impulse response (t = 100.0)

Question 14

t	Covariance of discretized Gaussian kernel
0.1	$\begin{bmatrix} 0.0133 & 0 \\ 0 & 0.0133 \end{bmatrix}$
0.3	$\begin{bmatrix} 0.2811 & 0 \\ 0 & 0.2811 \end{bmatrix}$
1.0	$\begin{bmatrix} 1.0000 & 0 \\ 0 & 1.0000 \end{bmatrix}$
10.0	$\begin{bmatrix} 10.0000 & 0 \\ 0 & 10.0000 \end{bmatrix}$
100.0	$\begin{bmatrix} 100.0000 & 0 \\ 0 & 100.0000 \end{bmatrix}$

Table 1: covariance of discretized Gaussian kernel

Question 15

When the variance t is below 1.0, the covariance is different to the estimated values; When t is above 1.0, the covariance is similar to the estimated ones. This is because that when the variance is small, the values of the Gaussian filter kernel after sampling have a tendency to become non-Gaussian. The smaller the variance t, the fewer points sampled from the region within 3 standard deviations of the mean.

Question 16

Figure 26: original image

Figure 27: Gaussian functions of variance t = 1.0

Figure 28: Gaussian functions of variance t = 4.0

Figure 29: Gaussian functions of variance t = 16.0

Figure 30: Gaussian functions of variance t = 64.0

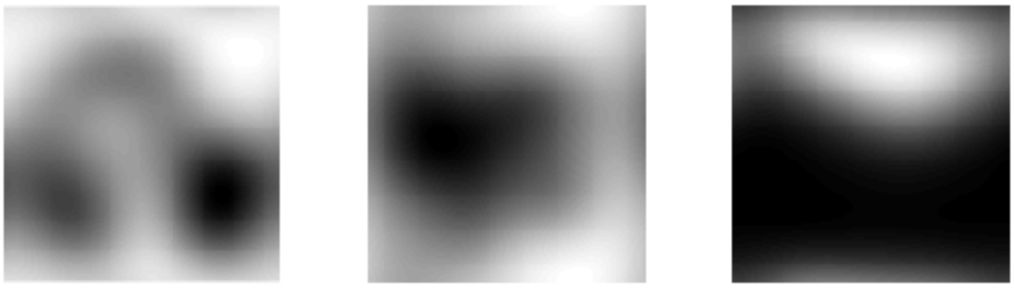

Figure 31: Gaussian functions of variance t = 256.0

As the variance t increases, the images become more blur. This is reasonable because the high the variance t, the lower the cut-off frequency in the frequency domain for the Gaussian function. The lower the cut-off frequency is, the more components with high frequency, such as edges and corners, are lost. That's why we can observe that the images are becoming more blur as the value of t goes up.

3.1 Smoothing of noisy data

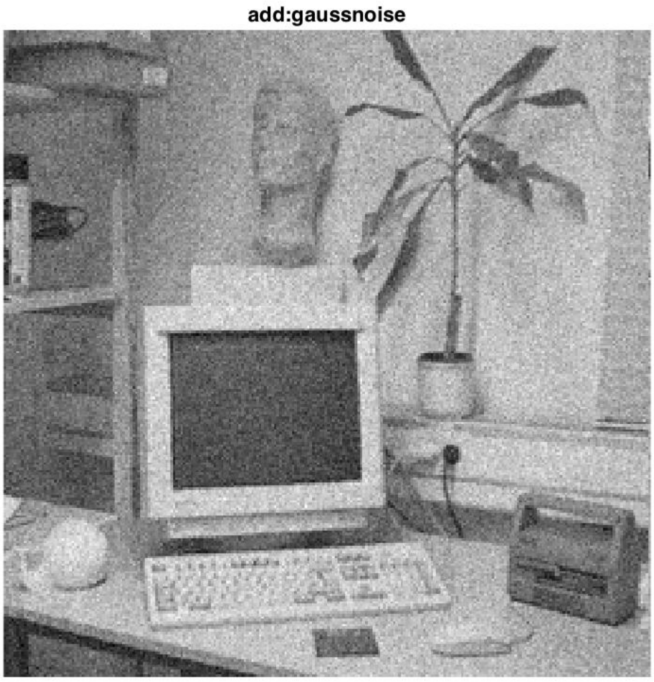

Figure 32: image with Gaussian noise

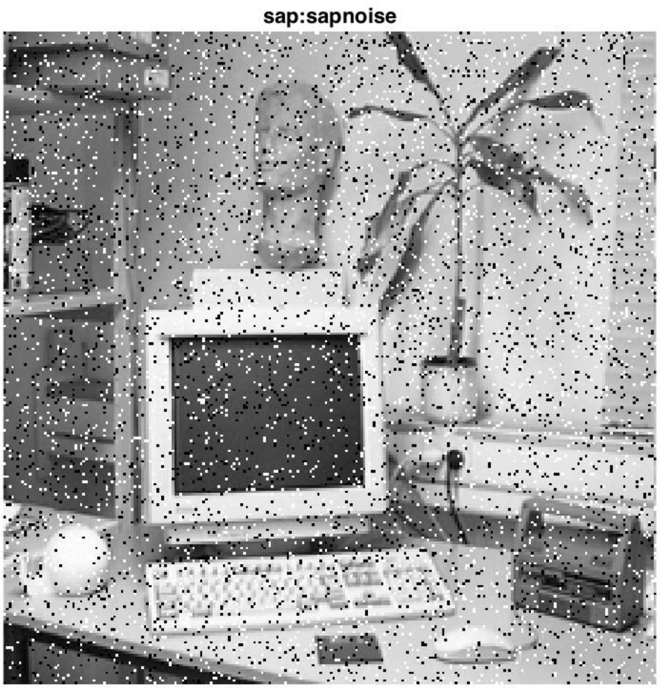

Figure 33: image with salt and pepper noise

Question 17

The following 3 pictures show how the original images with Gaussian noise are processed by Gaussian smoothing, median filtering and ideal low-pass filtering respectively. We could easily find that Gaussian smoothing has the best performances than the others.

Figure 34: image with Gaussian noise processed with Gaussian smoothing

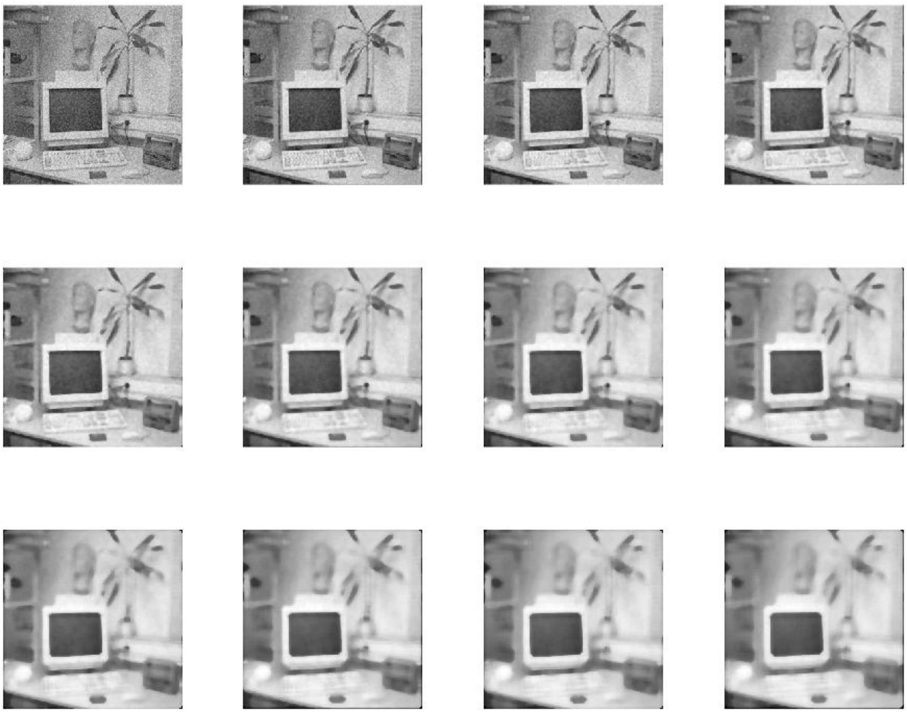

Figure 35: image with Gaussian noise processed with median filtering

Figure 36: image with Gaussian noise processed with ideal low-pass filtering

The following 3 pictures show how the original images with salt and pepper noise are processed by Gaussian smoothing, median filtering and ideal low-pass filtering respectively. And we notice that the median filtering works best in this case.

Figure 37: image with salt and pepper noise processed with Gaussian smoothing

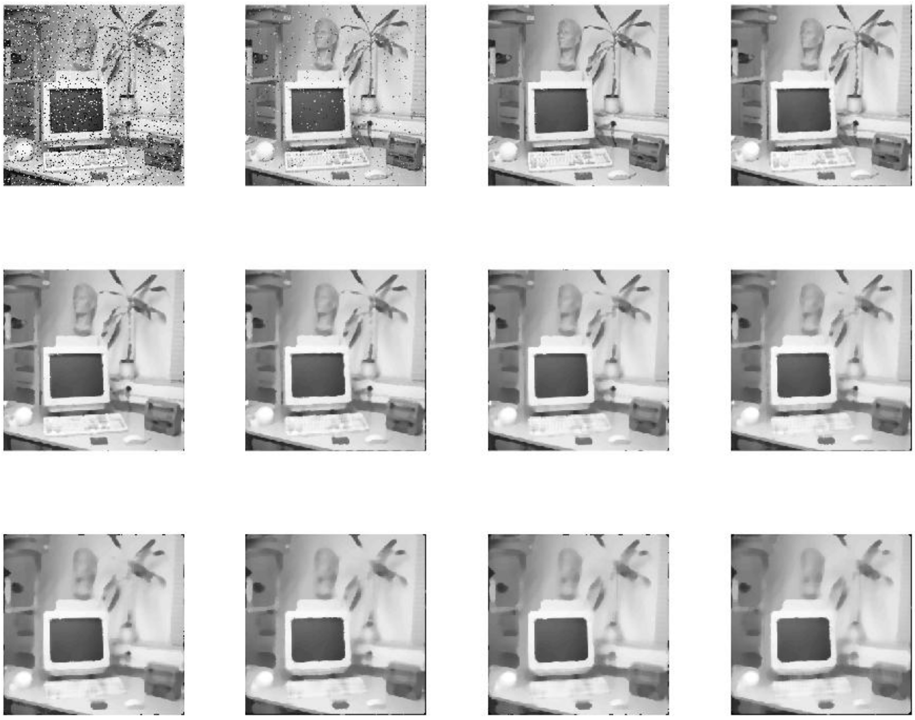

Figure 38: image with salt and pepper noise processed with median filtering

Figure 39: image with salt and pepper noise processed with ideal low-pass filtering

From results above, we could compare these filters:

- Gaussian smoothing
  - Advantage: as a low-pass filter, it is much more smoothing than the others because the Fourier transform of Gaussian is still a Gaussian;
  - Disadvantage: images become blurring when the variance of the kernel goes larger; integrate salt and pepper noise into the image.
- Median filter: non-linear
  - Advantage: eliminate local extreme values, such as salt and pepper noise; keep the sharp edge;
  - Disadvantage: image looks like paintings when the window size goes larger; visible information lost.
- Ideal low-pass filter
  - Advantage: easy and direct in form;
  - Disadvantage: delete all the high frequency; not smoothing enough; ringing.

Question 18

Gaussian smoothing and ideal low-pass filter are both linear while median filtering is not.

- Gaussian smoothing: when the variance increases, the blurring increases too in the image. The Fourier transform of the Gaussian function:

$$\hat{g}(\omega_1, \omega_2; t) = \int_{\omega_1=-\infty}^{\infty} \int_{\omega_2=-\infty}^{\infty} g(x, y; t) e^{-i(\omega_1 x + \omega_2 y)} dxdy = e^{-({\omega_1}^2 + {\omega_2}^2)t/2}$$

- Median filter: it works with the median, not the mean of a neighborhood of pixels; it has better performance in preserving edges, and works well with salt and pepper noise.
- Ideal low-pass filter: when the variance is large in partial domain, means a small variance and low cut off frequency in frequency domain, which indicates more information of high frequency is lost; the sinc function whose one component is comprised by concentric circles in spatial domain causes ringing effect.

3.2 Smoothing and subsampling

Question 19

Figure 40: images with subsampling and smoothing at the same time

From the figure above, we could see that the subsampled images processed by Gaussian filter (second row) and low-pass filter (third row) are much smoother than the original subsampled one. Besides, we can still notice the existence of the ringing effect in the images of the third row.

Question 20

From the phenomenon we could conclude that smoothing before subsampling could prevent from losing information of the image. This can be explained by the sampling theorem. To lower the information loss during the process of the subsampling, the sampling frequency should be above the Nyquist frequency, which is half the maximum frequency. If we first smooth the image by Gaussian filter or low-pass filter, the maximum frequency will decrease, which causes a reduction of the Nyquist frequency. This means that the relatively low sampling frequency could also satisfy the requirement of the sampling theorem. In this way, the information loss will be decreased.