Answers to questions in

Lab 1: Filtering operations

Name: Per Emil Hammarlund Program: TMAIM

Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

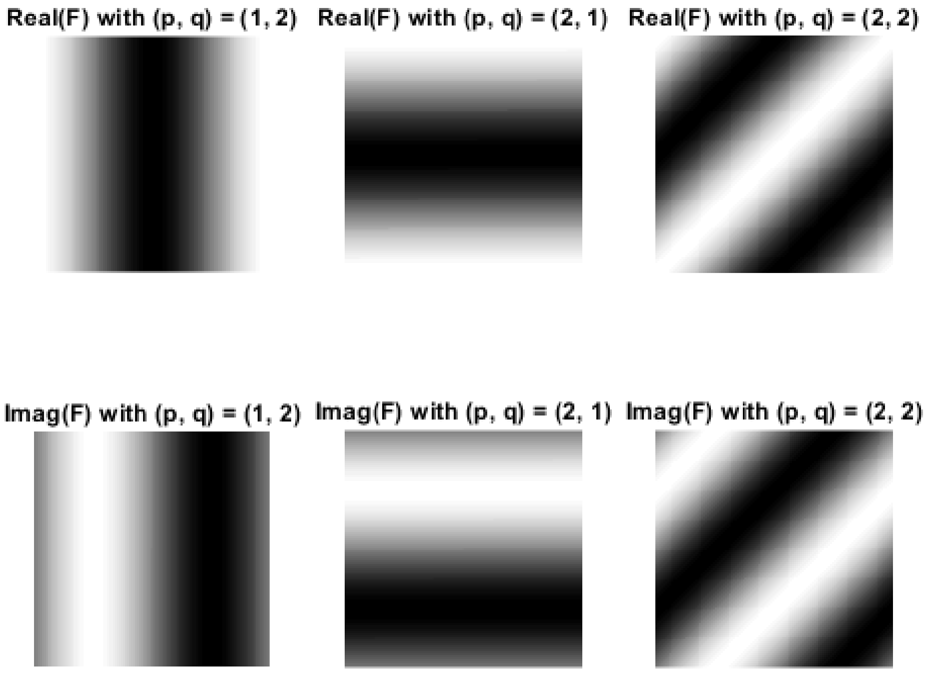
Good luck!

Question 1: Repeat this exercise with the coordinates p and q set to (5, 9), (9, 5), (17, 9), (17, 121), (5, 1) and (125, 1) respectively. What do you observe?

Answers:
Between (5, 9) and (9, 5), we mirror around the diagonal. When the frequency along one dimension is increased, the number of waves along said dimension increases.
Between (17, 9) and (17, 121), we see a change in angle but with the same frequencies.
Between (5, 1) and (125, 1) we have a 180 degree rotation with the same frequency.

Question 2: Explain how a position (p, q) in the Fourier domain will be projected as a sine wave in the spatial domain. Illustrate with a Matlab figure.

Answers:
A position (p, q) in the fourier domain will be a sine wave that is built up by a frequency p in the y-direction and a frequency q in the x-direction.



Question 3: How large is the amplitude? Write down the expression derived from Equation (4) in the notes. Complement the code (variable amplitude) accordingly.

Answers:

The amplitude is the magnitude divided by the number of pixels in the image. For all images with a single value = 1, and 0 on the rest. We get:

1 / N,

where N is the number of pixels in the image.

Question 4: How does the direction and length of the sine wave depend on p and q? Write down the explicit expression that can be found in the lecture notes. Complement the code (variable wavelength) accordingly.

Answers:

The wavelength of the sine wave when transformed to the spatial domain is given by:

lambda = 1 / sqrt(p^2+ q^2)

(where p and q are the frequencies that the indices correspond to)

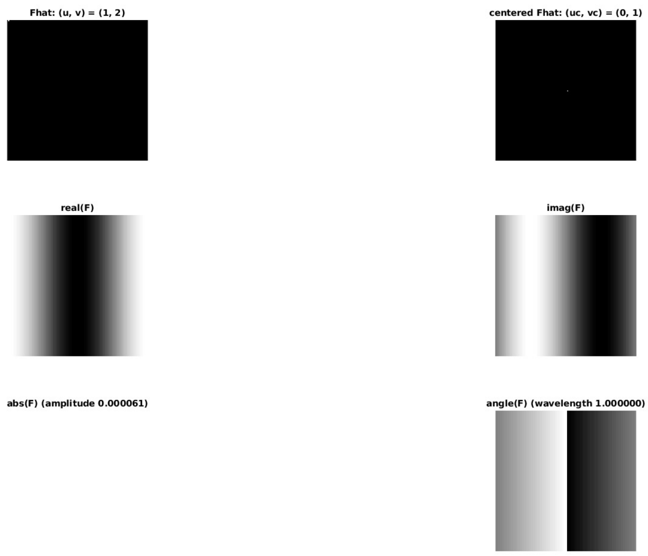
Question 5: What happens when we pass the point in the center and either p or q exceeds half the image size? Explain and illustrate graphically with Matlab!

Answers:

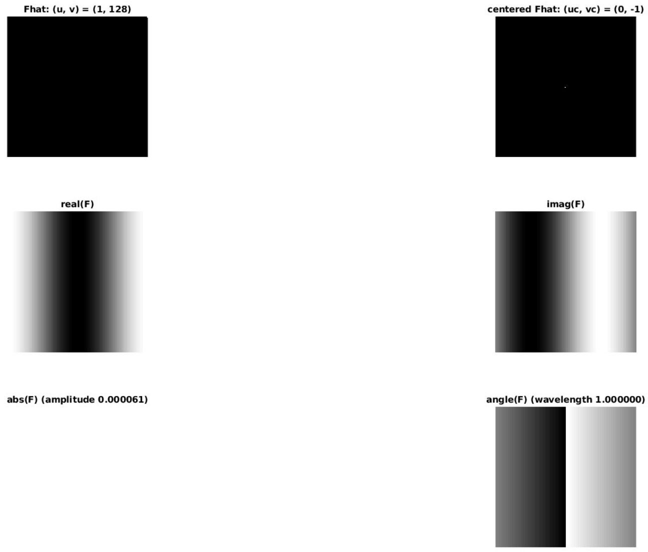
When one of the dimensions exceeds half the image size, the image that gets reproduced in the spatial domain is simply the negative side of the same frequency starting from the edge of the right side and increasing as we go inwards towards the middle of the image in the fourier domain when we have the non shifted version of the fourier domain.

When the fourier domain is shifted s.t. the zero frequency is in the middel of the image crossing the middle of the image is the same as getting the negative version of the same frequency.

When we are left of center:



When we are right of center:



Question 6: What is the purpose of the instructions following the question *What is done by these instructions?* in the code?

Answers:

The purpose is to center the fourier domain around the zero frequency, which makes sure that we represent each frequency in the original Fhat with the index that is closest to zero.

Question 7: Why are these Fourier spectra concentrated to the borders of the images? Can you give a mathematical interpretation? Hint: think of the frequencies in the source image and consider the resulting image as a Fourier transform applied to a 2D function. It might be easier to analyze each dimension separately!

Answers:

The Fourier spectra are concentrated along the borders since variation in the source image is only found in one of the dimensions in F and G respectively.

Question 8: Why is the logarithm function applied?

We use the logarithm in order to rescale the image data into a more readable format.

Question 9: What conclusions can be drawn regarding linearity? From your observations can you derive a mathematical expression in the general case?

Answers:

We can see that the operations that have been performed before the foruier transform can be seen after the zero pad fourier transform. For example, take the adding of the original image with its transpose. We see that the original image and its transpose are reflections of each other and that their sum simply results in the combination of the two.

$$\mathcal{F}\left[a\, \mathrm{f}_1(m,n) + b\, \mathrm{f}_2(m,n))\right] = a\, \hat{f}_1(u,v) + b\, \hat{f}_2(u,v)$$

$$a\, \mathrm{f}_1(m,n) + b\, \mathrm{f}_2(m,n)) = \mathcal{F}^{-1}\left[a\, \hat{f}_1(u,v) + b\, \hat{f}_2(u,v)\right]$$

Question 10: Are there any other ways to compute the last image? Remember what multiplication in Fourier domain equals to in the spatial domain! Perform these alternative computations in practice.

Answers:
Multiplication in the Fourier domain is equal to a convolution in the spatial domain and vice versa. In the given code, we are multiplying in the spatial domain and then converting to the Fourier domain. We can alternatively perform the transformation to the Fourier domain on each of the components first and then perform a convolution in the Fourier domain to get the same result.

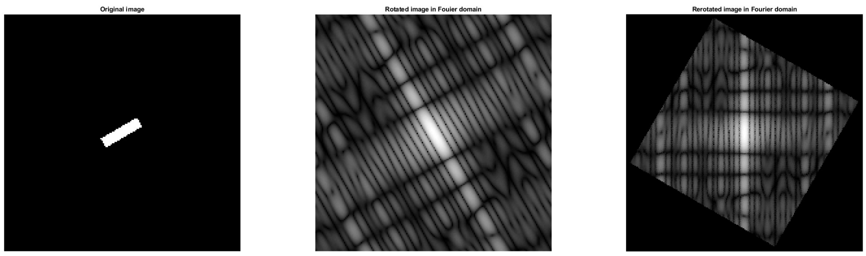
Question 11: What conclusions can be drawn from comparing the results with those in the previous exercise? See how the source images have changed and analyze the effects of scaling.

Answers:
Stretching in one of the domains gives a shrinkage in the other and vice versa.

Question 12: What can be said about possible similarities and differences? Hint: think of the frequencies and how they are affected by the rotation.

Answers:
Rotating the original image results in the same rotation in the Fourier domain. And rotating the Fourier domain image back to it's original rotation gives the same image as in question 11, but slightly distorted which is a consequence of us not being able to perfectly represent the rotated image in discrete form.

30 degree rotation in spatial domain:



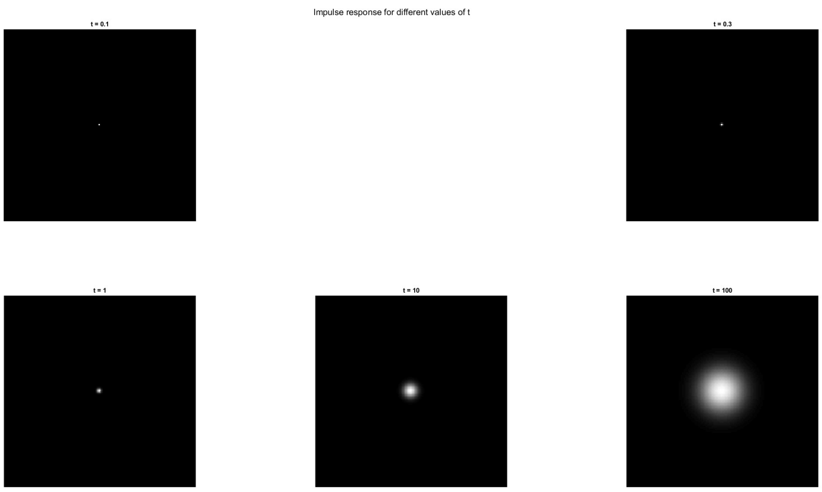
Question 13: What information is contained in the phase and in the magnitude of the Fourier transform?

Answers:
The magnitude of the Fourier transform contains information about how large the waveforms are i.e. how large the pixel values will be in the spatial domain.

The phase contains information about how the waveforms are shifted along their directions i.e. where the images edges will end up in the spatial domain.

Question 14: Show the impulse response and variance for the above-mentioned t-values. What are the variances of your discretized Gaussian kernel for t = 0.1, 0.3, 1.0, 10.0 and 100.0?

Answers:
The impulse responses took the following form:



And the variances where:

var_0_1 =

```
    0.0133         0
         0    0.0133
```

var_0_3 =

```
    0.2811    0.0000
    0.0000    0.2811
```

var_1 =

```
    1.0000    0.0000
    0.0000    1.0000
```

var_10 =

```
    9.9995   -0.0000
   -0.0000    9.9995
```

var_100 =

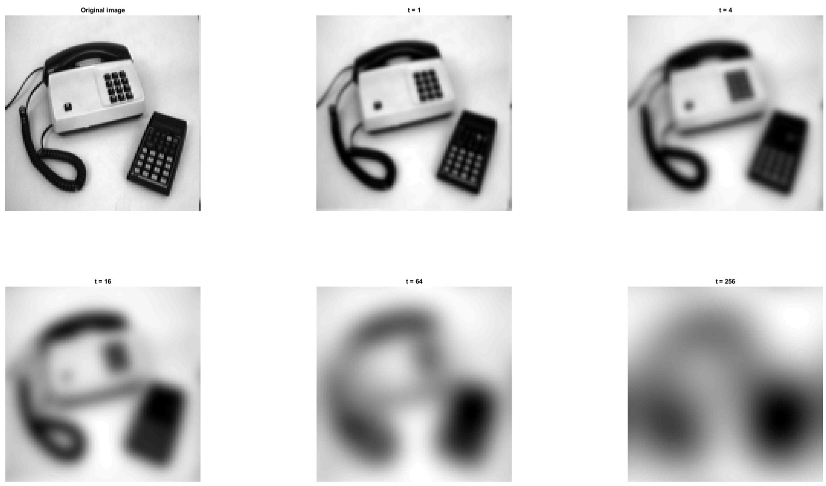
```
   99.9871    0.0000
    0.0000   99.9871
```

Question 15: Are the results different from or similar to the estimated variance? How does the result correspond to the ideal continuous case? Lead: think of the relation between spatial and Fourier domains for different values of t.

Answers:
The variances are close to the ideal continuous case, but not exact due to discretization.

Question 16: Convolve a couple of images with Gaussian functions of different variances (like t = 1.0, 4.0, 16.0, 64.0 and 256.0) and present your results. What effects can you observe?

Answers:



The larger the variance of the kernel, the more blurry the image.

Question 17: What are the positive and negative effects for each type of filter? Describe what you observe and name the effects that you recognize. How do the results depend on the filter parameters? Illustrate with Matlab figure(s).

Answers:

The gaussian filter is good att removeing white noise, but does make the image more blurry. When using the gaussian filter, the larger the variance the more blurry the image and the smaller the variance the less of an effect the filter has on the image. As the variance approaches zero, the image tends towards the original unaltered image.

Median filtering is good at removing noise such as salt and pepper noise. As the size of the kernel for the image increases, we lose details in the image and the image starts to look like a painting. With the smallest possible kernel 1, the image simply becomes the original image.

Low pass filtering gives good results when removing white noise from the image, but lets the low part of salt and pepper noise through the filter creating an image that looks like it has been distorted with white noise. As the cutoff frequency decreases, the images produced by the filter start getting a “wavyness” to them and as the cutoff increases towards the entire range, we get the original image.

Question 18: What conclusions can you draw from comparing the results of the respective methods?

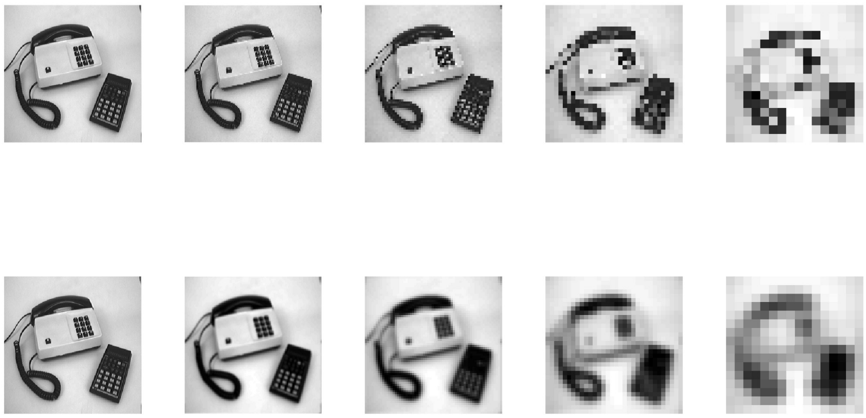
Answers:
That depending on the type of noise that we have in our image, the filter that will enhance the image best will depend on what kind of noise there is in the image.

For white noise, both the gaussian and the low.pass filter perform well. The low-pass filter is better at preserving details in the image such as edges, while the gaussian gets rid of the white noise better. Median filtering is not a good choice when faced with white noise in the image.

For salt-and-pepper noise, the best filter of choice is the median filter and neither the gaussian nor the low-pass filter perform particularly well on salt-and-pepper noise.

Question 19: What effects do you observe when subsampling the original image and the smoothed variants? Illustrate both filters with the best results found for iteration i = 4.

Answers:
Non smoothed above and smoothed below, subsampling increases as we move from left toi right in the image below:



The smoothed version of the image manages to keep the general structure of the original image longer then the non-smoothed version of the subsampling of the image.

Question 20: What conclusions can you draw regarding the effects of smoothing when combined with subsampling? Hint: think in terms of frequencies and side effects.

Answers:
That the smoothed version of the image is able to maintain the general structure of the image better since information from neighboring pixels is spread out across pixels in the image. So the subsampled image will contain information from pixels that where lost in the subsampling.
