Answers to questions in Lab 1: Filtering operations

Name: Jonathan Pernow   Program: TIPUM

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.

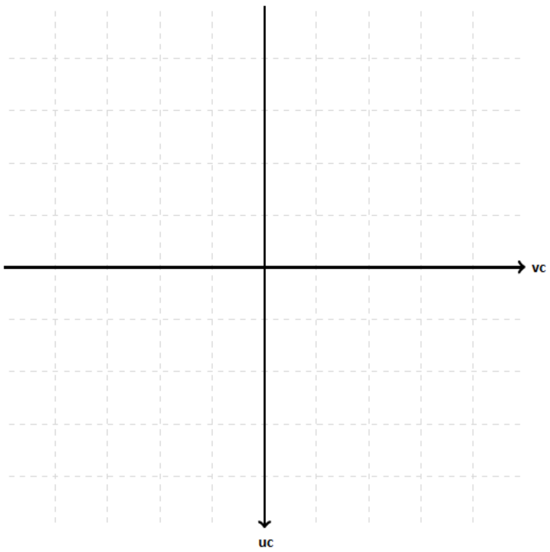
Good luck!

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:

The main observation is that points closer to the origin (in the shifted coordinate system) result in a wave with lower frequency than points further away. The direction of the wave is directed towards the origin. A point on the uc-axis corresponds to a wave in uc-direction, and a point on the vc-axis corresponds to a wave in vc-direction.

(u, v)	(uc, vc)
(5,9)	(4,8)
(9,5)	(8,4)
(17,9)	(16,8)
(17,121)	(16,-8)
(5,1)	(4,0)
(121,1)	(-4,0)

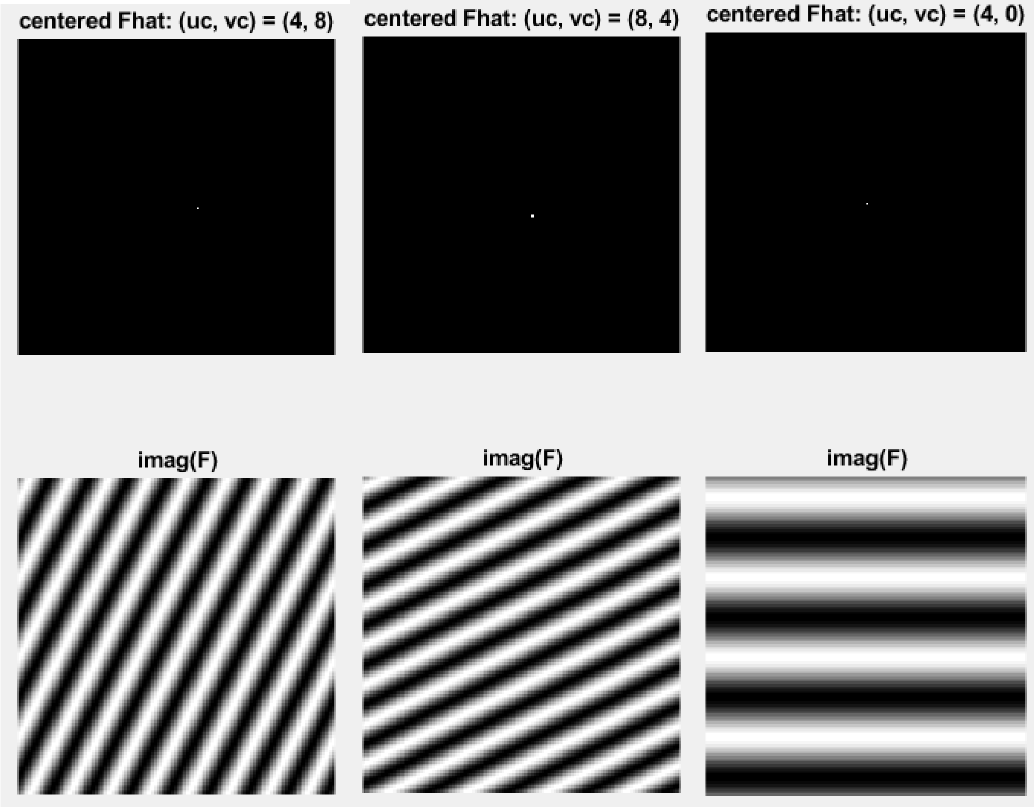


Question 2: Explain how a position (p, q) in the Fourier domain will be projected as a sine wave in the spatial domain. Illustrate with a Matlab figure.

Answers:
The discrete inverse Fourier transform can be rewritten from the usual form to a form with sine and cosine.

$$F(x,y) = \frac{1}{N}\sum_{u,v} \hat{F}(u,v)\ e^{i\frac{2\pi}{N}(ux+vy)}$$
$$F(x,y) = \frac{1}{N}\sum_{u,v} \hat{F}(u,v)\ \left(\cos\left(\frac{2\pi}{N}(ux+vy)\right) + i\sin\left(\frac{2\pi}{N}(ux+vy)\right)\right)$$

Since we're only displaying one white pixel against a black surrounding $\hat{F}(u,v)$ only has one non-zero value. This means that the imaginary part of F can be projected as a sine wave in the spatial domain. As discussed in the previous question the direction and frequency of this sine wave depends on the point (u_c, v_c). When $u_c = 0$ we get a sine wave in the horizontal direction, and when $v_c = 0$ we get a sine wave in the vertical 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:
Since Matlab doesn't normalize in the Fourier transform we must divide by N^2 when doing the inverse Fourier transform.

$$|F(x,y)| = \left|\frac{1}{N^2}\sum_{u,v} \hat{F}\ (u,v)e^{i\frac{2\pi}{N}(ux+vy)}\right|$$

This can be simplified if we consider that $\hat{F}(u,v)$ only has one non-negative value and that the absolute value of the exponent is always one.

$$|F(x,y)| = \left|\frac{1}{N^2}\hat{F}(u,v)\right| = \frac{1}{128^2} = 0.000061$$

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:
From the lecture notes we get an expression for the wavelength

$$\lambda = \frac{2\pi}{\sqrt{\omega_1^2 + \omega_2^2}}$$

Replacing ω_1 and ω_2 with $\omega_1 = \frac{2\pi u_c}{N}$ and $\omega_2 = \frac{2\pi v_c}{N}$ respectively gives us

$$\lambda = \frac{N}{\sqrt{u_c^2 + v_c^2}}$$

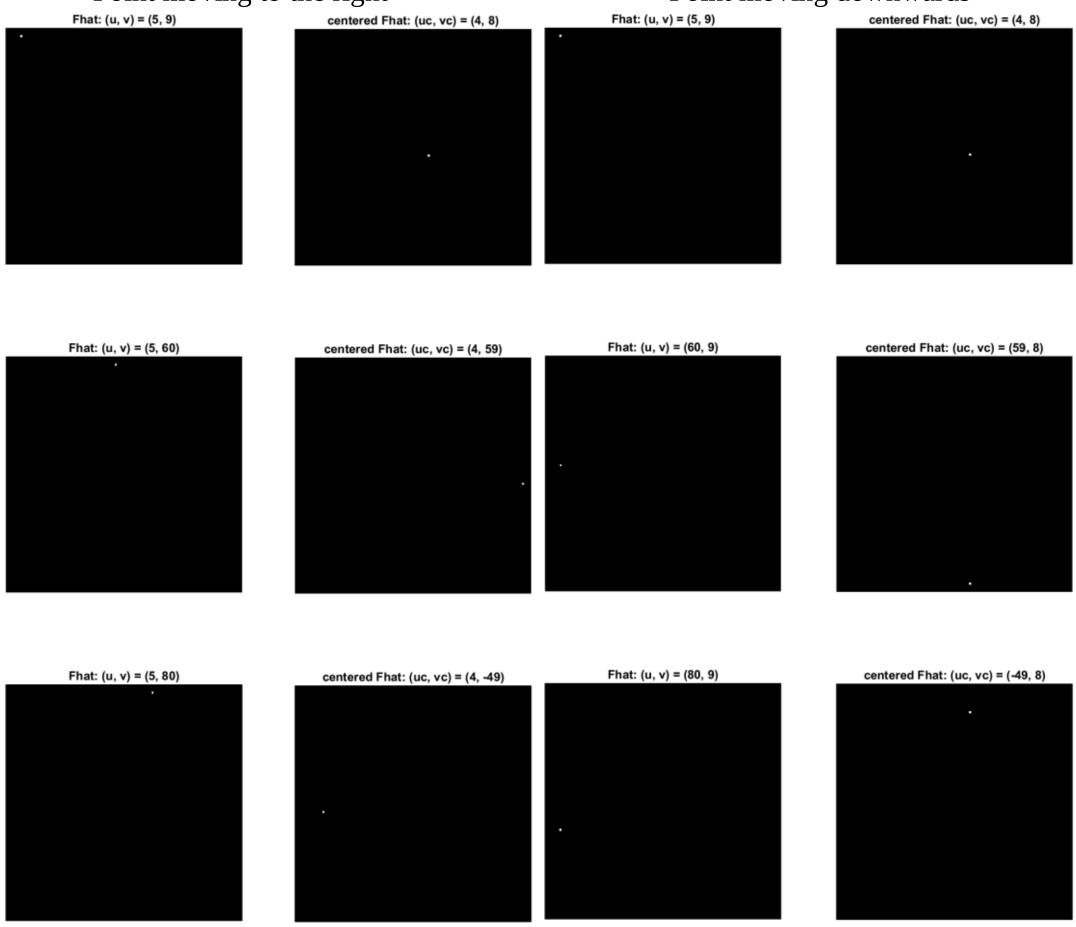
The direction of the wave is given by

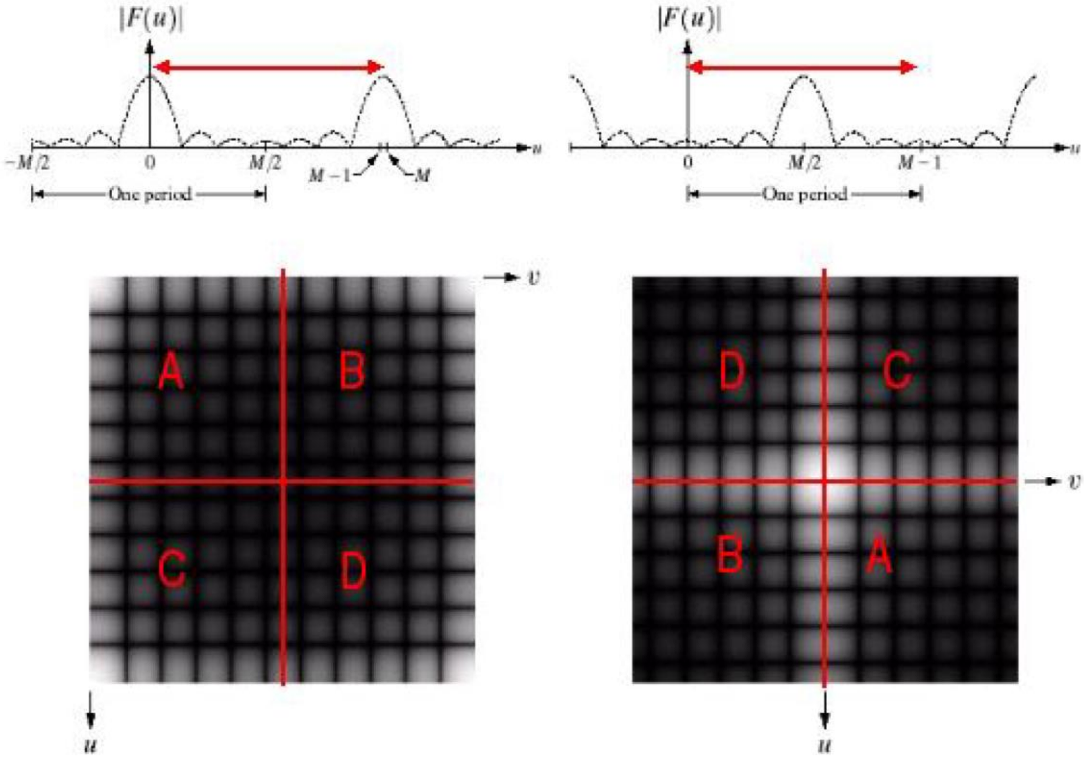
$$\phi(\omega_1, \omega_2) = \tan^{-1}\frac{Im(\omega_1, \omega_2)}{Re(\omega_1, \omega_2)}$$

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:

Point moving to the right　　　　Point moving downwards





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

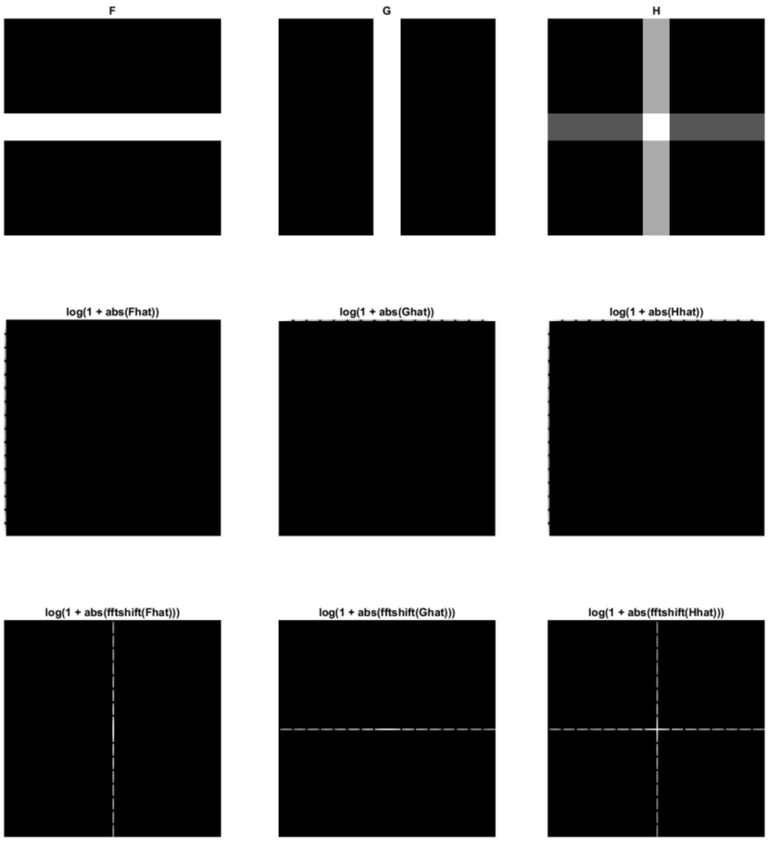
Answers:

It changes coordinates from $u, v \in [1, \dots, N]$ to $u_c, v_c \in \left[-\frac{N}{2}, \dots, \frac{N}{2} - 1\right]$ which results in shifted quadrants illustrated by question 5. The origin has been translated from the upper left corner to the center.

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:

Because horizontal and vertical lines in the spatial domain are represented as sine waves in either x or y direction. They are along the edges because the origin is place in the upper left corner.



$$\hat{f}(u,v) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-2\pi i\left(\frac{mu}{M} + \frac{nv}{N}\right)}$$

Image F:

$$f(m,n) = \begin{cases} 1, & 57 \leq m \leq 72 \\ 0, & else \end{cases}$$

Splitting Fourier transform into two sums

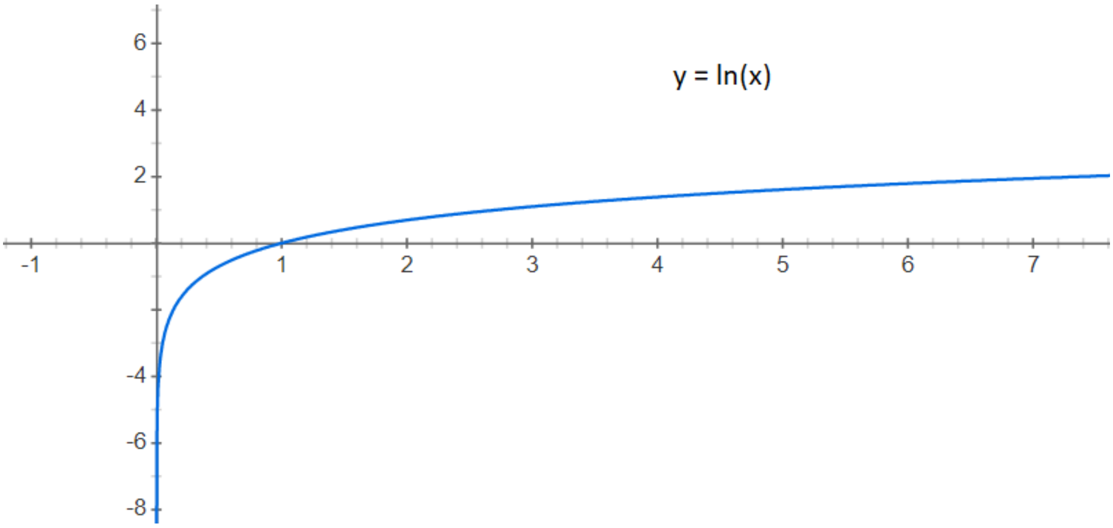
$$\hat{F}(u,v) = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} e^{-2\pi i\left(\frac{nv}{N}\right)} \sum_{m=57}^{72} e^{-2\pi i\left(\frac{mu}{M}\right)}$$

Question 8: Why is the logarithm function applied?

Answers:

Applying the logarithm functions results in a dynamic range compression for the display of the Fourier transform. This is done when there are a few points with magnitude significantly larger than the rest.

This type of compression enhances the low intensity pixel values, while compressing high intensity values into a relatively small pixel range. Hence, if an image contains some important high intensity information, applying the logarithmic operator might lead to loss of information.

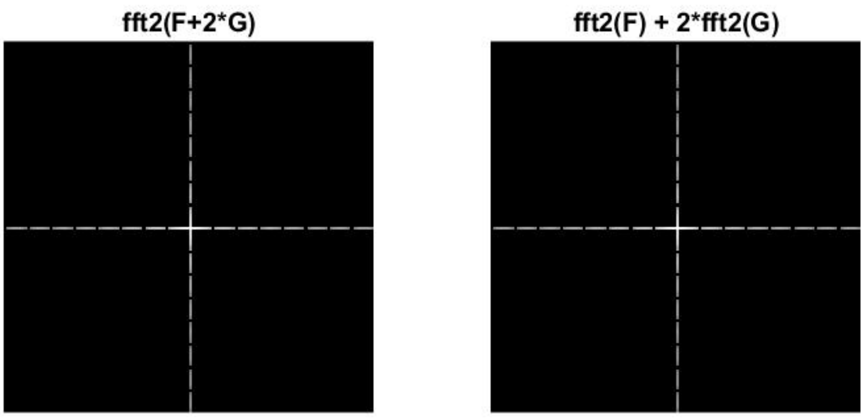


Question 9: What conclusions can be drawn regarding linearity? From your observations can you derive a mathematical expression in the general case?

Answers:
You can add two functions (images) or rescale a function, either before or after computing the Fourier transform. It leads to the same result. Shown by following equations and figures.

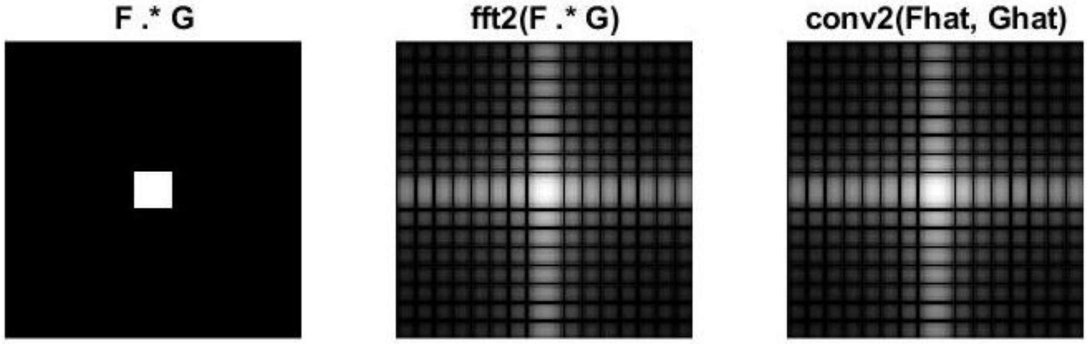
$$\mathcal{F}\left[a\,f_1(m,n) + b\,f_2(m,n)\right] = a\,\hat{f}_1(u,v) + b\,\hat{f}_2(u,v)$$

$$a\,f_1(m,n) + b\,f_2(m,n) = \mathcal{F}^{-1}\left[a\,\hat{f}_1(u,v) + b\,\hat{f}_2(u,v)\right]$$



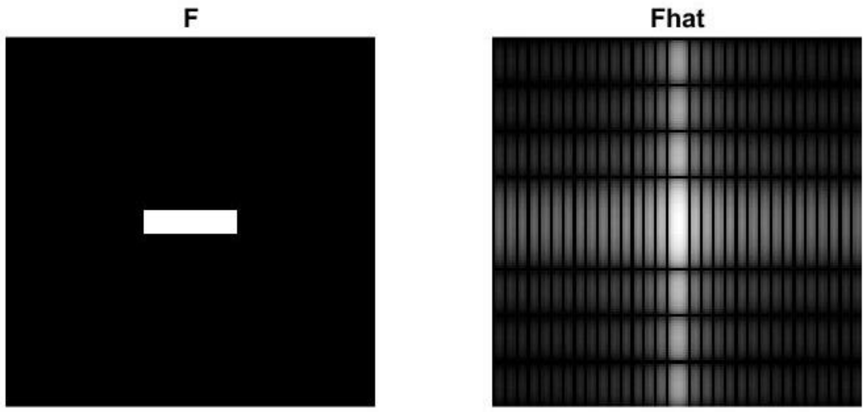
Question 10: Are there any other ways to compute the last image? Remember what multiplication in Fourier domain equals to in the spatial domain! Perform these alternative computations in practice.

Answers:
Since multiplication in the spatial domain equals convolution in the Fourier domain the image can also be computed by using the function *conv2* in Matlab. Instead of using FFT on the sum of F and G you can apply FFT to F and G separately and then convolve F with G. In order to get the same image, you also have to normalize the image with $\left(\frac{1}{N^2}\right)$ in the Fourier domain.



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.

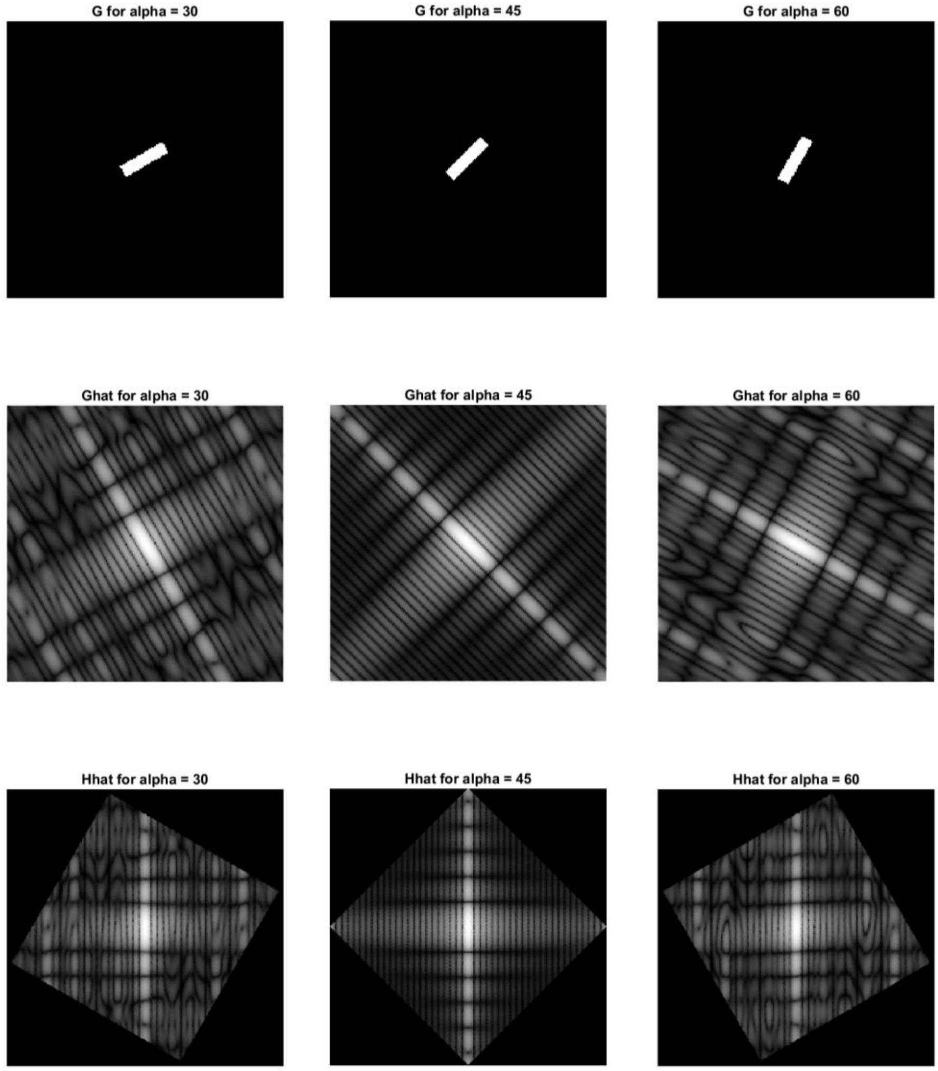
Answers:
The source image has been scaled up in the horizontal axis and scaled down in the vertical axis, which gives the opposite effect in the frequency domain. The frequency domain image has been scaled down in the horizontal axis and scaled up in the vertical axis.



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

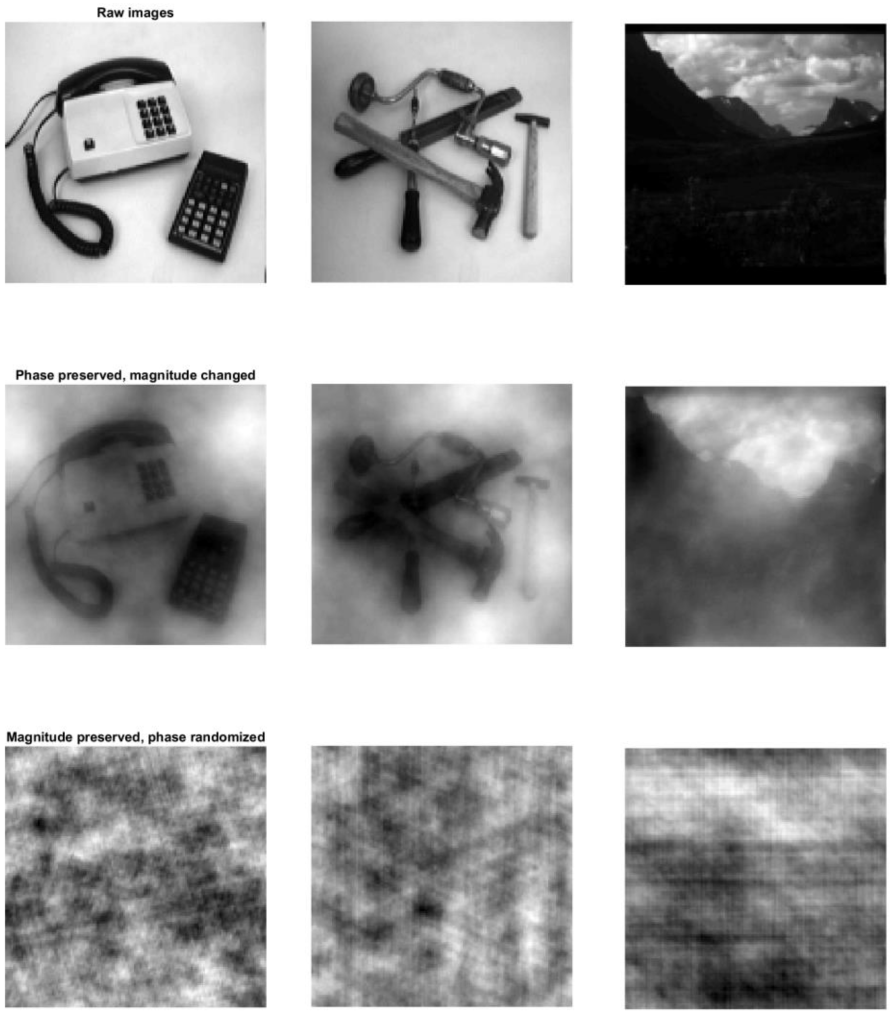
Answers:
When rotating the image, the waves in the Fourier domain also rotate but don’t change in magnitude and phase.

The edges of the rectangle become jagged when we rotate due to poor resolution (aliasing). The edges are therefore no longer perfect lines and show up as waves in the frequency domain.



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

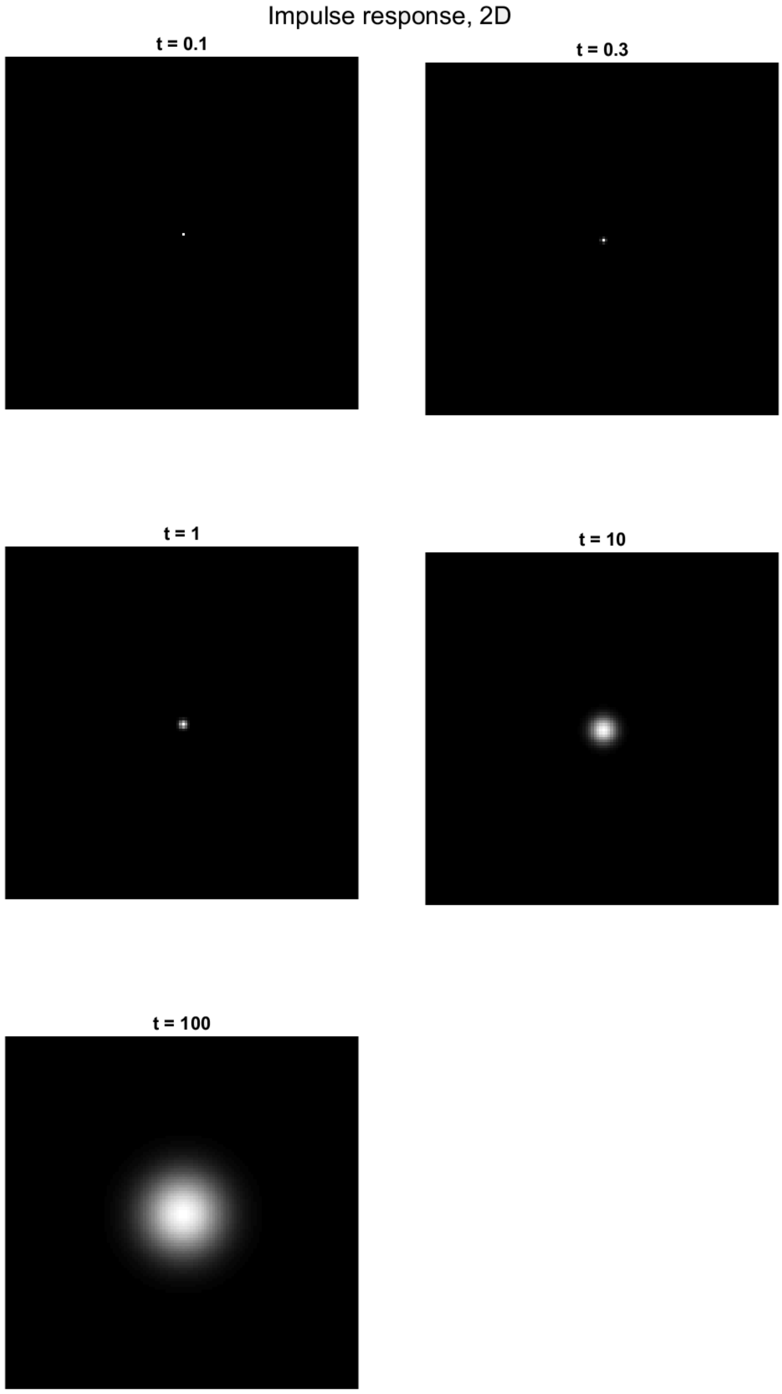
Answers:

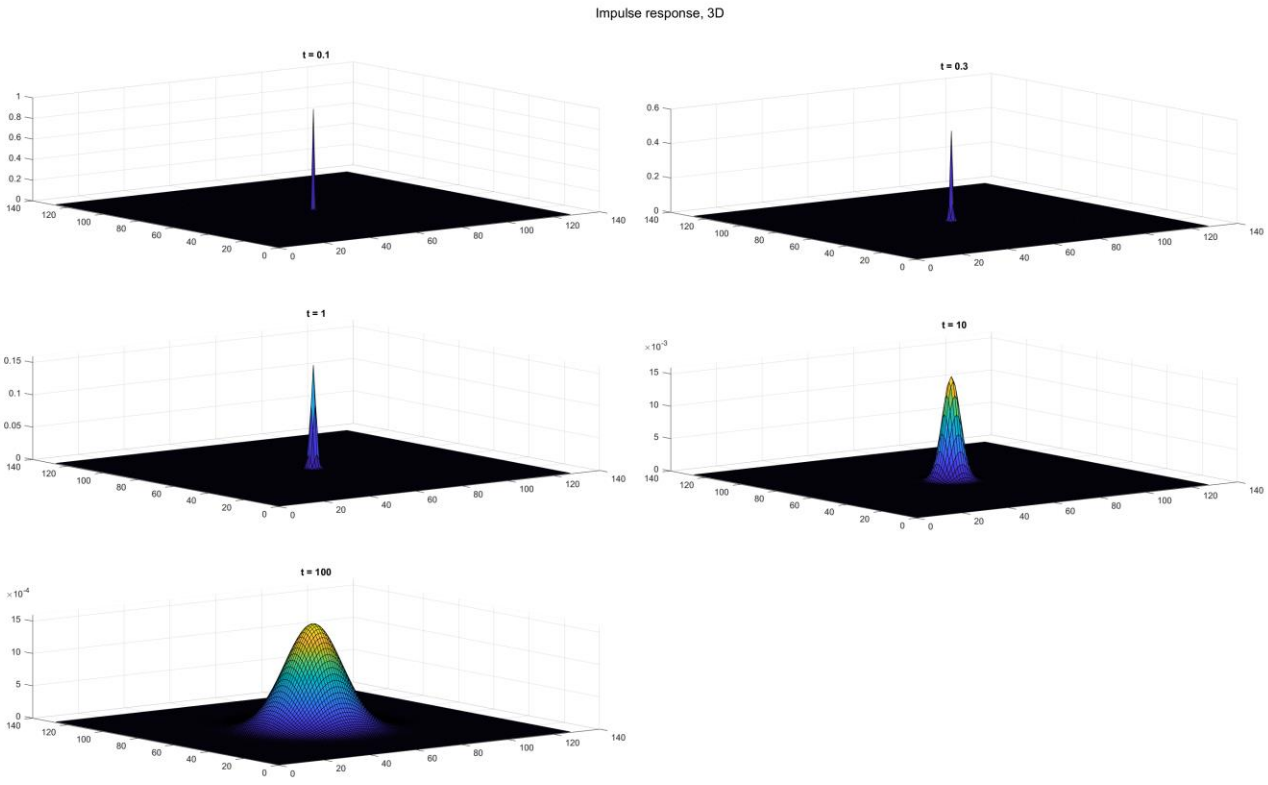


Phase defines how waveforms are shifted along its direction. Where edges will end up in the image (most important).
Magnitude defines how large the waveforms are. What grey-levels are on either side of edge (less important).

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:
Impulse responses from the Gaussian function.


Impulse response, 2D
t = 0.1
t = 0.3
t = 1
t = 10
t = 100



Comparing the resulting variances to the ideal case we see that only the first differ from the ideal case, with the difference increasing with decreased variance t.

$$C\big(g(\cdot,\cdot;t)\big) = t\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$
$$C\big(g(\cdot,\cdot;0.1)\big) = 0.0133\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$
$$C\big(g(\cdot,\cdot;0.3)\big) = 0.2811\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$
$$C\big(g(\cdot,\cdot;1)\big) = 1\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$
$$C\big(g(\cdot,\cdot;10)\big) = 10\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$
$$C\big(g(\cdot,\cdot;100)\big) = 100\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

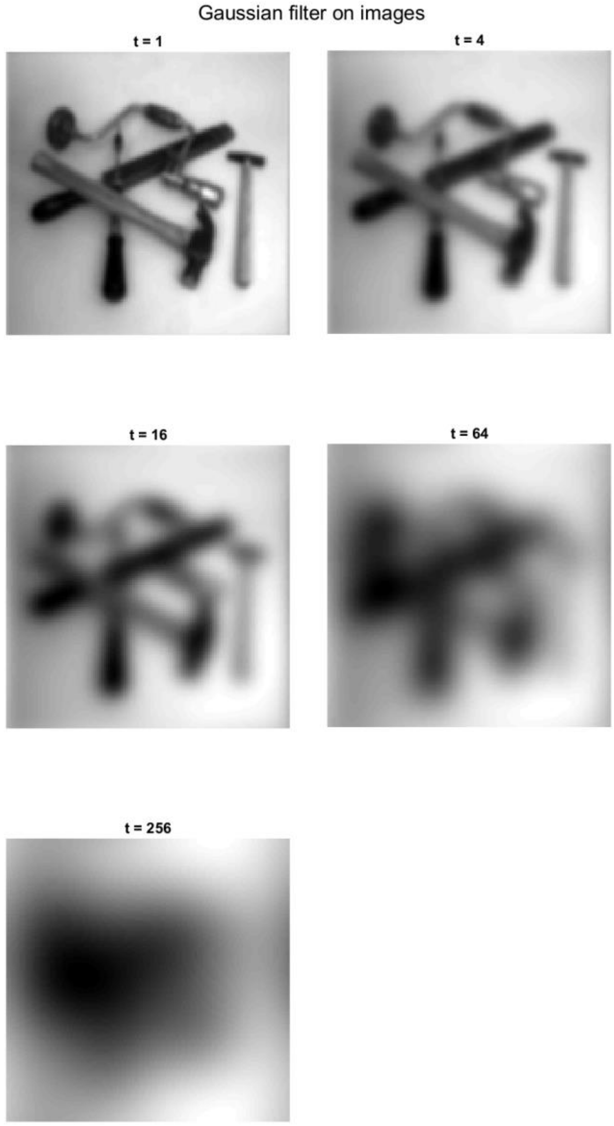
Question 15: Are the results different from or similar to the estimated variance? How does the result correspond to the ideal continuous case? Lead: think of the relation between spatial and Fourier domains for different values of t.

Answers:

Comparing the resulting variances to the ideal case we see that only the first differ from the ideal case, with the difference increasing with decreased variance t. This is due to discretization of a small amount of pixels, thus errors will be more noticeable.

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:

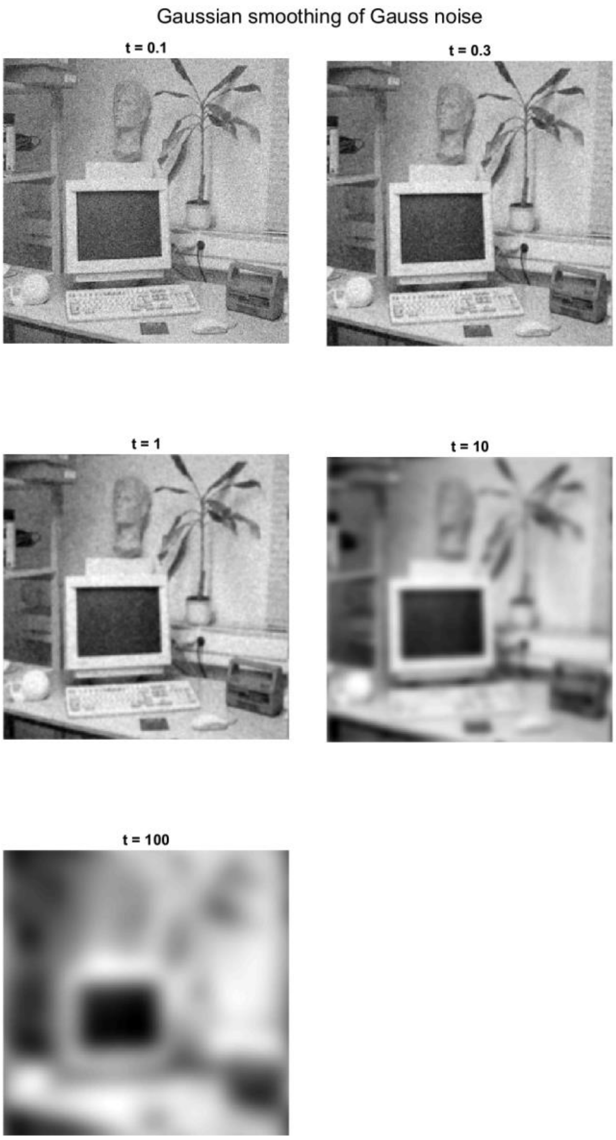


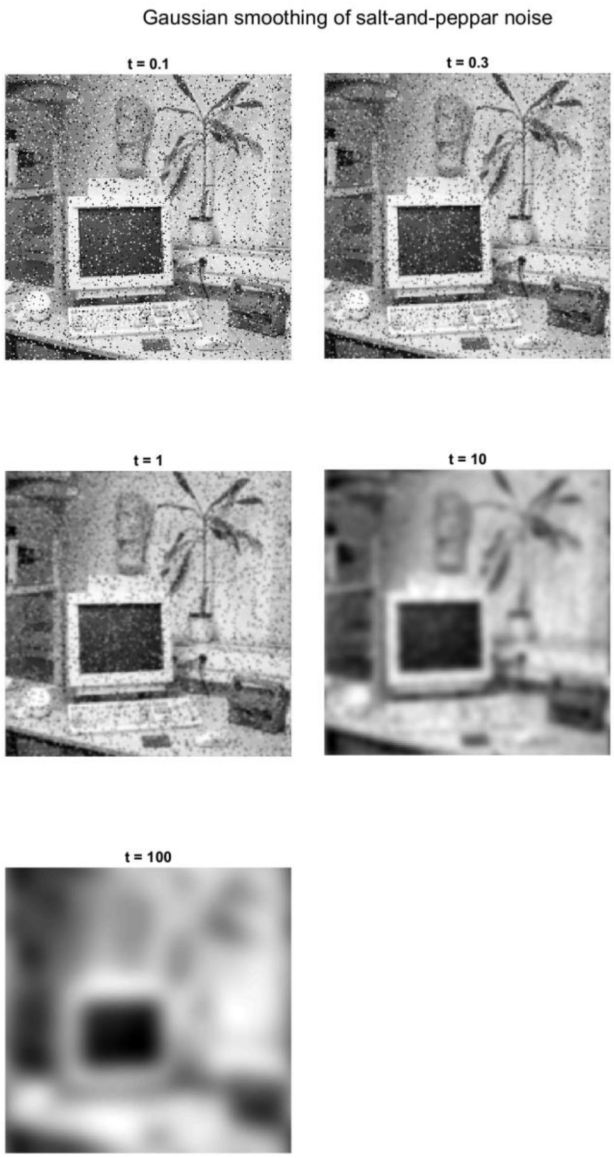
The higher the t-value is the more blurred the image will be. This is because an increasing t will increase the area of the image that will be affected by the Gaussian filter, meaning it will affect more frequencies.

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).

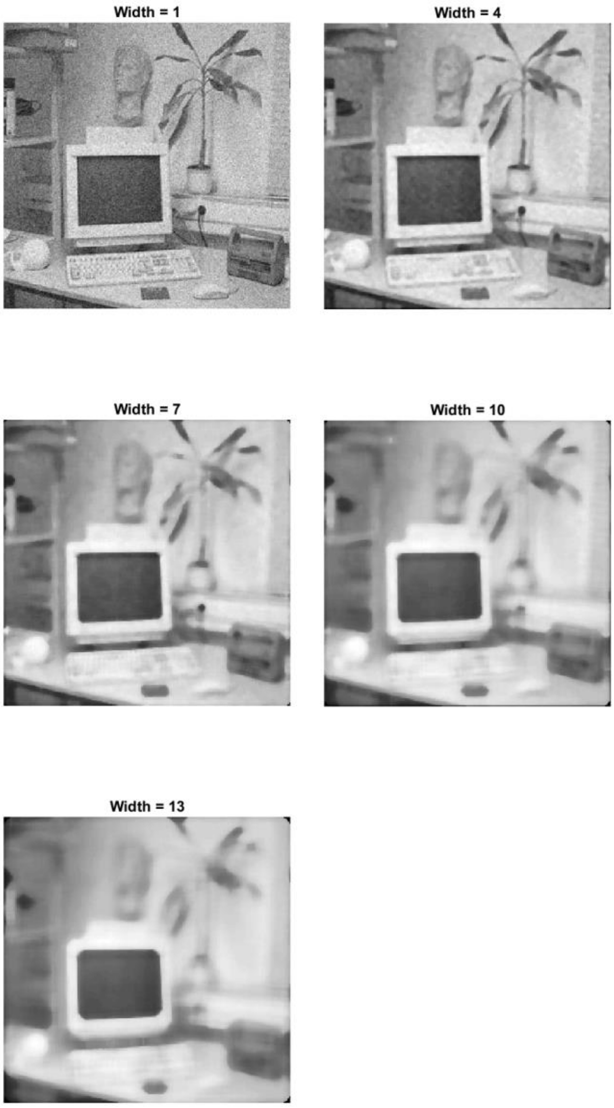
Answers:

The image to the left has a Gauss noise and the image to the right has a salt-and-pepper noise.

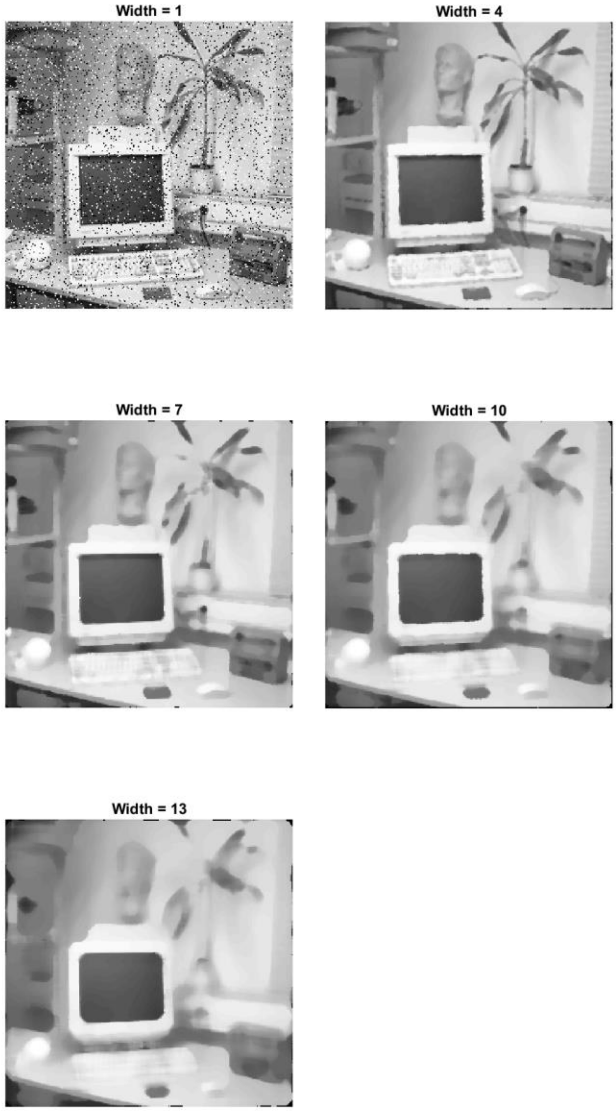


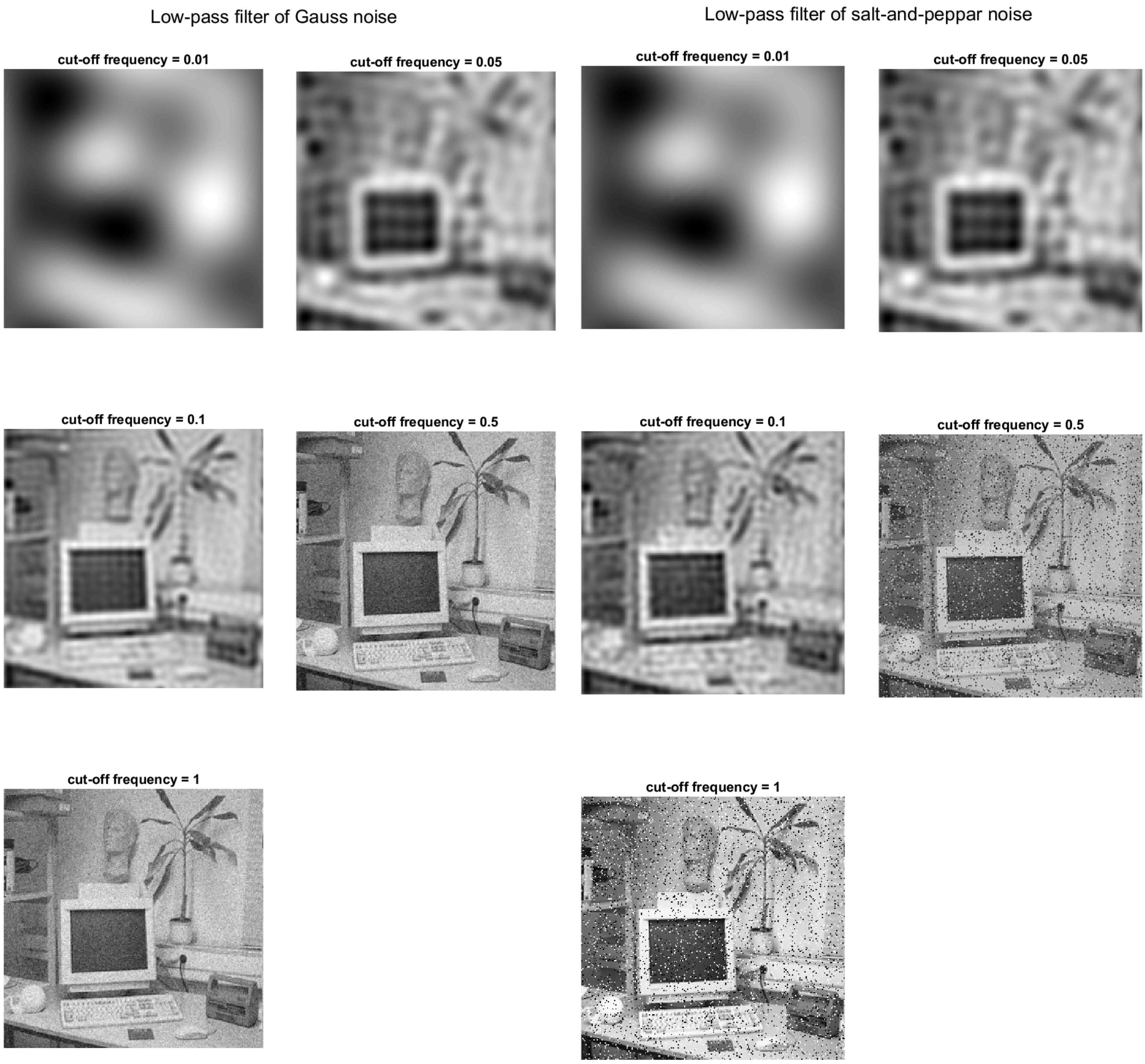


Median fitlering of Gauss noise



Median fitlering of salt-and-peppar noise





At a certain filter parameter value the noise is removed whilst the image not being too blurry to comprehend what is in the image. For the Gaussian filter a larger variance will result in a blurrier image. For the median filtering an increasing width will result in a blurrier image. For the low-pass filter a smaller cut-off frequency will result in a blurrier image since it will let lower frequencies through the filter.

Gaussian filter
+ Smooths the image
+ Good at reducing Gaussian noise
- Does not preserve edges
- Bad at reducing salt-and-pepper noise

Median filter
+ Preserves edges well
+ Good at reducing Gaussian and especially salt-and-pepper noise
- Some objects almost disappear
- Image looks more and more “cartoonish”

Ideal low-pass filter
+ Noise usually consists of high-frequencies which the low-pass removes. Real images have mostly low-frequencies and are therefore mostly unchanged. This assumes appropriate cut-off frequency.
- Creates a ringing effect
- Poor effect on Gaussian and salt-and-pepper noise

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

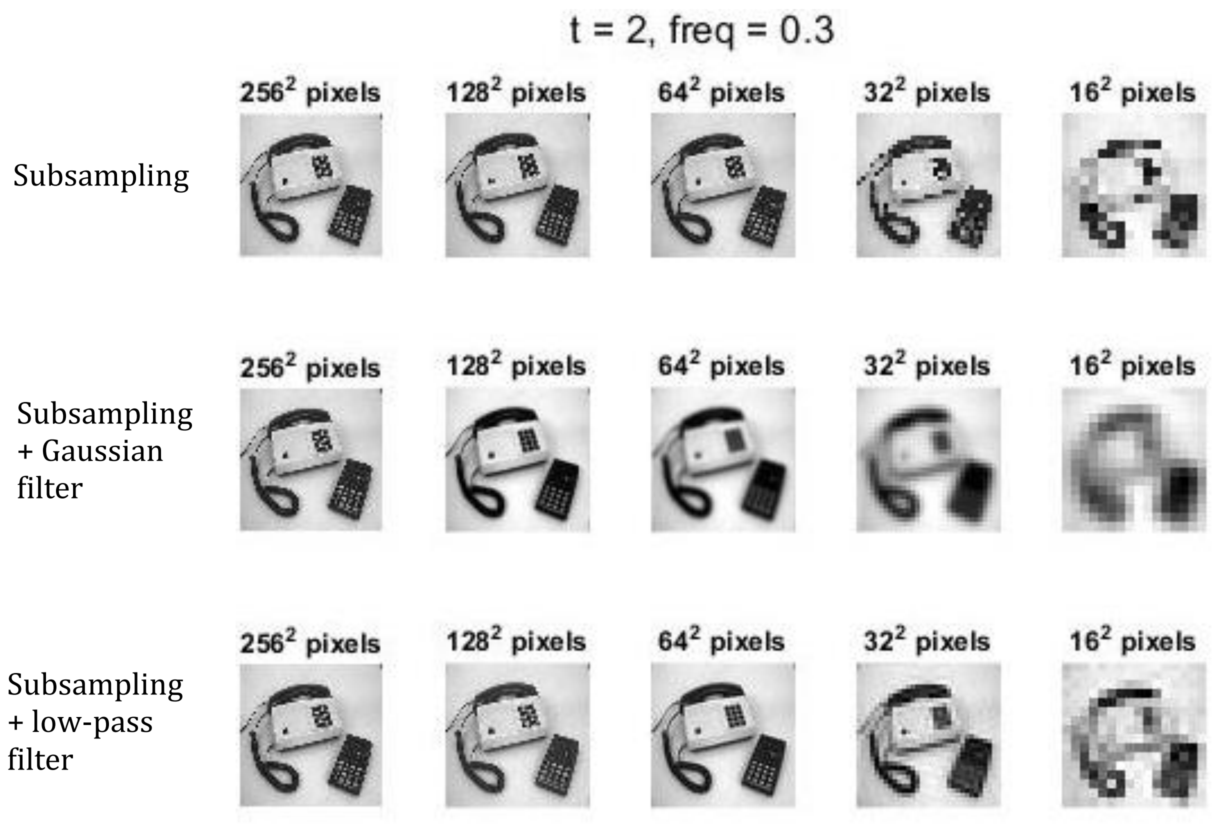
Answers:
Gaussian filter
- Works very well at removing the Gaussian blur, but does not remove the salt-and-pepper noise as well.
  A Gaussian filter removes high-frequencies (i.e. removes details) which means it works like a low-pass filter.

Median filter
- Gives an overall smoother image than the other filters and keeps the edges. Particularly effective against salt-and-pepper noise since a few high valued pixels won't affect the median filter matrix.

Low-pass filter
- Has the worst performance of them all, barely removing the 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:



When subsampling the original image all its edges become jagged. In the last iteration the image is no longer recognizable because there is a high contrast between neighboring pixels. It gets more pixelated and thus loses its details.

By blurring the image with a Gaussian filter before sampling results in a better final image with smoother edges (larger amount of different grayscale values).

The more subsampled the image is the image will contain more high-frequencies. The low-pass filter will reduce some of these contrasts. A smoother image would have been achieved by decreasing the cut-off frequency but would induce more ringing noises.

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:
Subsampling reduces the information (in form of pixels) by half, this will remove details, and details are found in high-frequencies. By blurring before subsampling the image look more like the original because more data can be preserved, less information will be lost. Blurring images before subsampling can work as anti-aliasing.
