Answers to questions in
Lab 1: Filtering operations

Name: Xavi de Gibert and Pablo Laso
Program: DD2423 CVHT21 HT21-1 Image Analysis and Computer Vision

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.

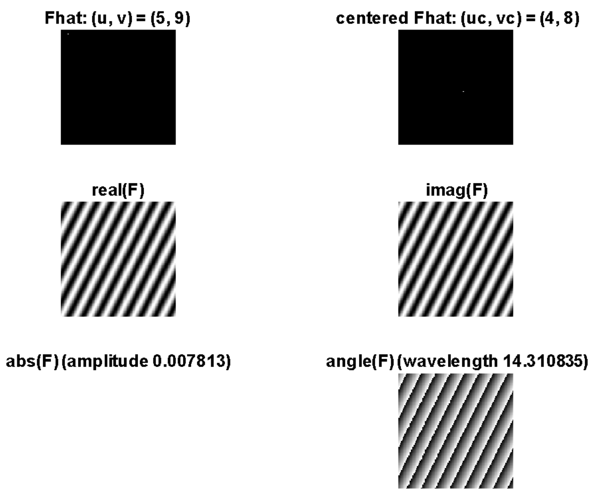
Good luck!

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:

If we define a vector from the origin (0,0) to the point where point (p,q) is, we could observe, after the applying Fourier Transform, a pattern of lines that are orthogonal to such a vector. Moreover, the further away our point is from the origin in the time domain, the higher the frequency will be, i.e. we will observe that parallel lines are now closer in the frequency domain.

NB: The magnitude stays the same. We can see that the wavelength actually changes by π/2, since cos(x - π/2) = sin(x).



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:

Essentially, points in the *frequency* domain refers to the "composition" of sinusoidal waves with specific frequencies (defined by each point). In this case, a position/point (p,q) in the Fourier domain corresponds to a cosine/sine (real/imaginary) with that frequency in the time domain. Mathematically, this is proven by describing the sine wave according to *Euler's identity*, and then applying the *Fourier Transform* formula, obtaining a *Dirac delta function*, i.e., the "point" in the Fourier space. Applying the inverse FT will give us the sinusoidal wave back.

NB: Strictly, Euler's identity defines the *sine wave* as the imaginary part of the exponential "e", i.e., we will have two Dirac functions (half magnitude each), equidistant from the origin.

As we can observe from Figure 1, a discrete sine wave will yield a (two) unit impulse or delta function, and vice versa.

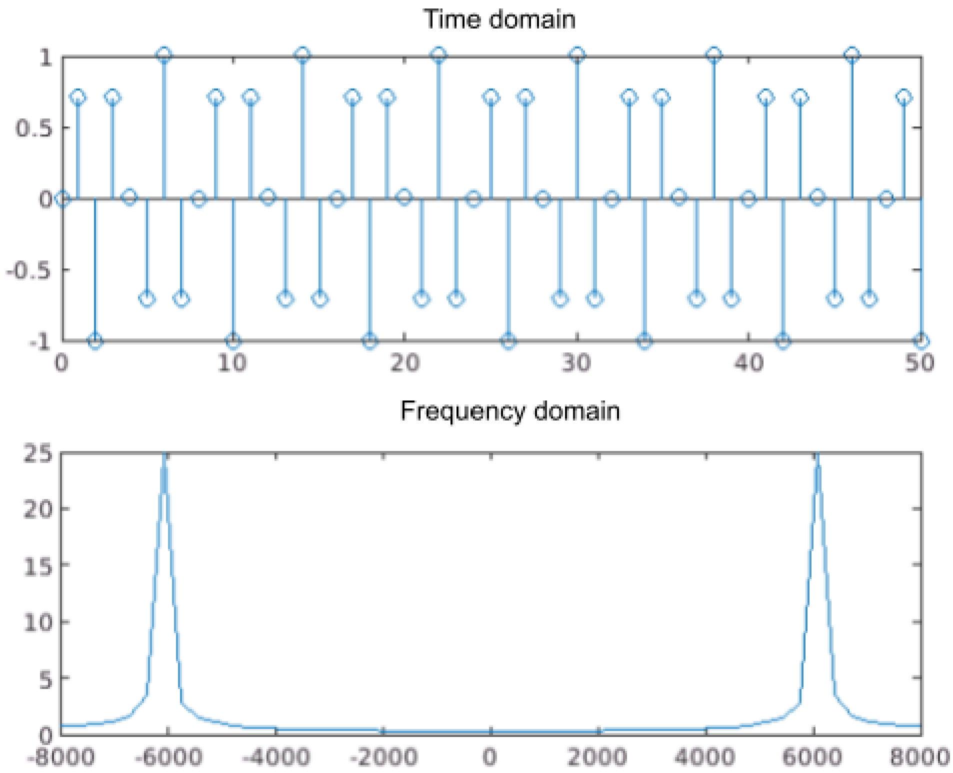


Figure 1. MATLAB plot of a discrete sine wave in time (above) and frequency (below) domains.

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:

→ starting from Equation 4,

$$F(x) = F_D^{-1}(\hat{F})(x) = \frac{1}{N}\sum_{u\in[0,N-1]^2} F(u)\cdot e^{\frac{2\pi i u^T x}{N}}$$

given $F(u) = \begin{cases} 1, & \text{if } x=p,\ y=q. \\ 0, & \text{else.} \end{cases}$,

$$F(x) = \underbrace{\frac{1}{N}}_{\text{amplitude}} \cdot \underbrace{e^{\frac{2\pi i u^T x}{N}}}_{\text{sine wave}}$$

→ $A = \frac{1}{N} = \frac{1}{n \cdot m} = \frac{1}{128} = \underline{0.0078}$

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 formula for the *wavelength* given in the lecture notes is the following:

$$\lambda = \frac{1}{\sqrt{u^2+v^2}},$$

Depending on the position we define the point (p,q), we will obtain a different sine wave. If it is closer to the center (0,0), waves will have greater wavelengths, as well as a smaller frequency (inversely proportional). Points located far from the center will have a higher frequency, and shorter wavelength. Wave direction is given by the vector taking from the origin to the point (p,q).

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:

As the Fourier transform has a period and is symmetric around the origin, the elements that go past the half of the image size have their replicas on the first half of the image.

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

Answers:

The purpose of the instructions is to locate the points properly when they exceed the half of the image and center the points p and q values. We modify the range from [0, N-1] to [-N/2,N/2-1] and also the angular frequency now goes from [-pi, pi). It does the same operation as ffshift.

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:

When applying Formula 3 for 2D FT, the summation over the y is going to be null except for the first column, i.e., when y=0. This is due to the fact that image F only spatially changes in the y-axis, as can be observed from Figure 2.

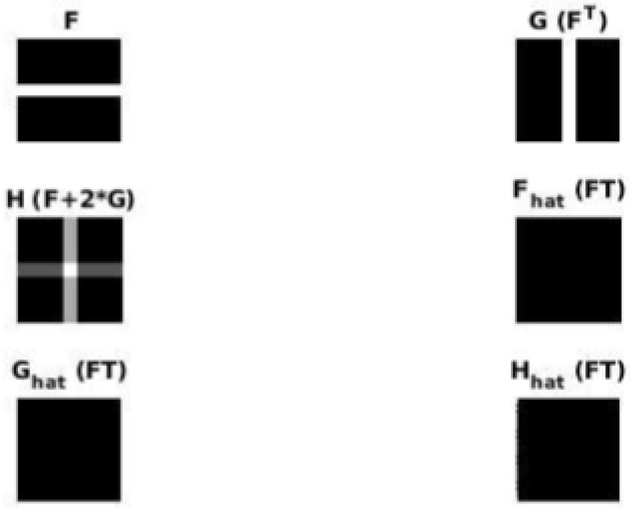


Figure 2. FT of different images.

Question 8: Why is the logarithm function applied?

Answers:

A log scale is usually applied for exponential growth curves, since they would increase too quickly and taking the log allows us to fit the curve within a smaller graph (Figure 3), for the sake of visualization.

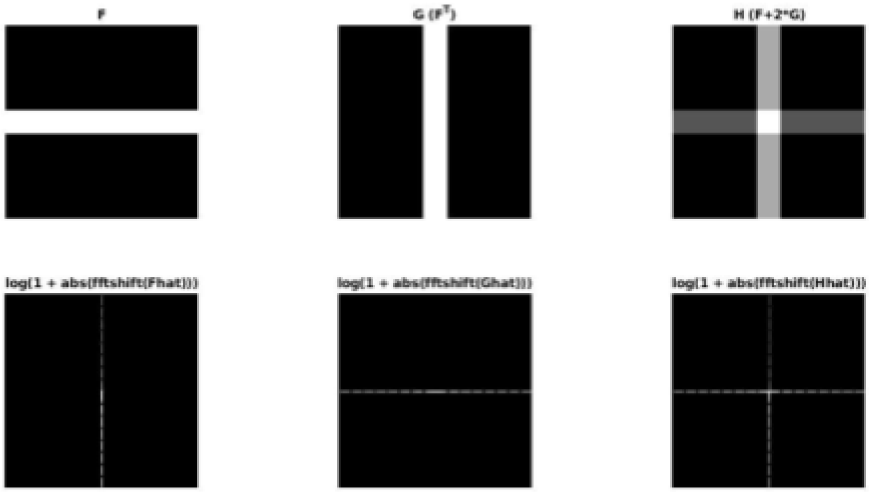


Figure 3: Log scaled, shifted FT of the images.

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

Answers:

According to the *linearity* property of the FT,
FT{H} = FT{F+2*G} = FT{F} + FT{2*G}, since H = F + 2*G
Therefore,
FT{a*f(m,n) + b*g(m,n)} = a*FT{f(m,n)} + b*FT{g(m,n)}

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:

After multiplying F*G, we obtain the intersection square and its FT (Figure 4a). Multiplying two functions in each of the two domains does not yield the same result. Instead, the multiplication in one domain equals the *convolution* on the other. In this case, the convolution (which can be used for, say, filtering) in the time domain, is reduced to a simple multiplication in frequency domain (useful for, e.g., designing a filter or reduce computational costs), as illustrated in Figure 4b.

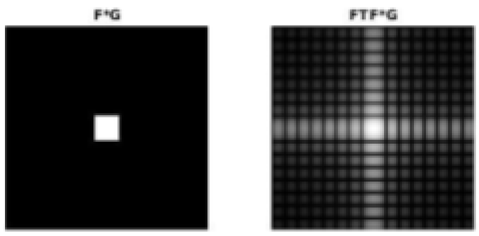


Figure 4a: Original image and FT.

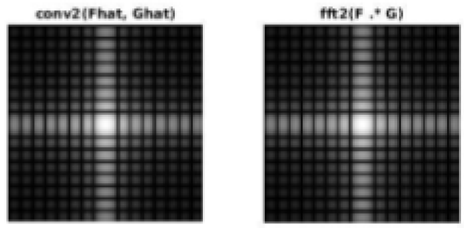


Figure 4b: Convolution and FT.

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:

When comparing the results from previous exercise (F_before) and the newly defined F (F_new), we notice that the square is now a rectangle, more precisely, it has been stretched horizontally (i.e. lower frequency in the x-axis), yet shrank vertically (i.e. higher frequency in this axis).

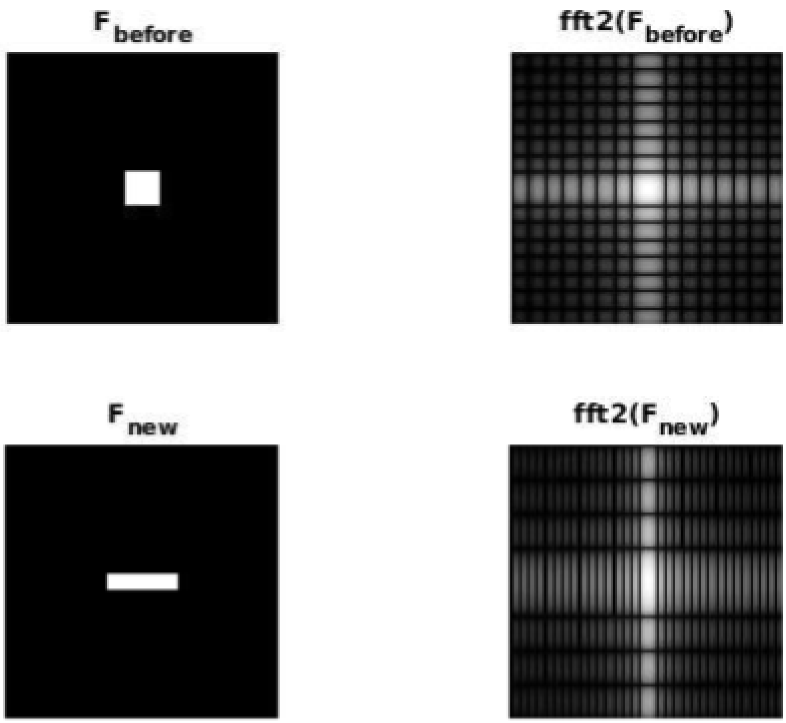


Figure 5. Difference between F_new and F_before

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:

Since the image is centered at the same point (origin 0,0) and has only been rotated, the FT will show that very same degree of rotation in the frequency spectrum. As we can observe from Figure 6, the FTs are identical, yet rotated accordingly. The only difference is the slight distortion introduced in rotations 30º and 60º in time domain, since the rectangle perimeter is irregular. This can is translated into the FT, where some distortion is highly apparent.

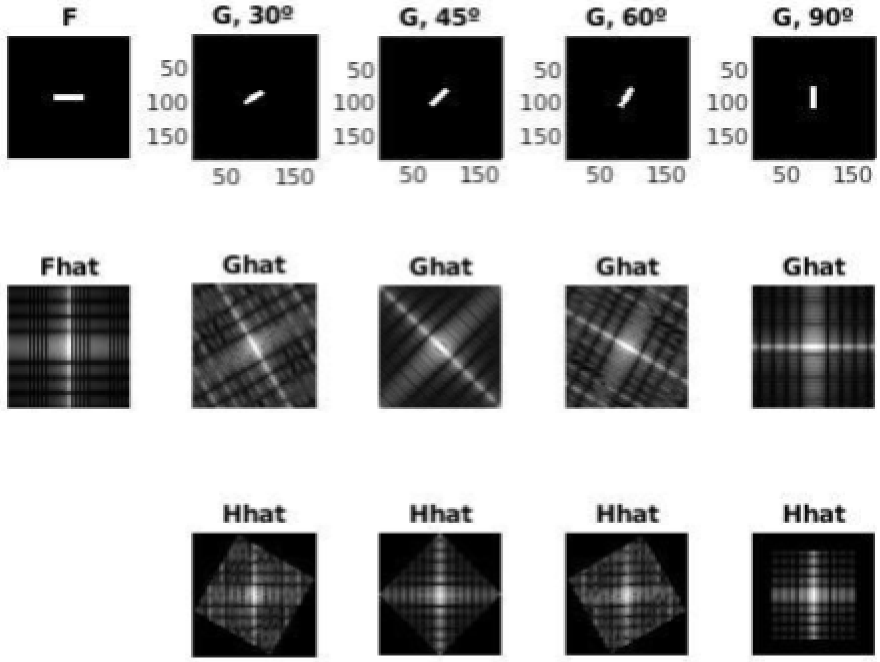


Figure 6. Rotating an image and its effect on the Fourier domain.

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

Answers:

Magnitude contains pixel intensity information. However, should we try to reconstruct an image from the Fourier spectrum by using the magnitude only, we will find white patches with probably no sense at all. By adding the information described in the phase, we can figure out the location of the pixels, which will yield the structure or shape of the original image (i.e. the edges).
As we can observe from Figure 13, the images are recognizable when we keep the phase information, since the edges are maintained. Observing the magnitude allows no distinction between images, since no contours can be detected.

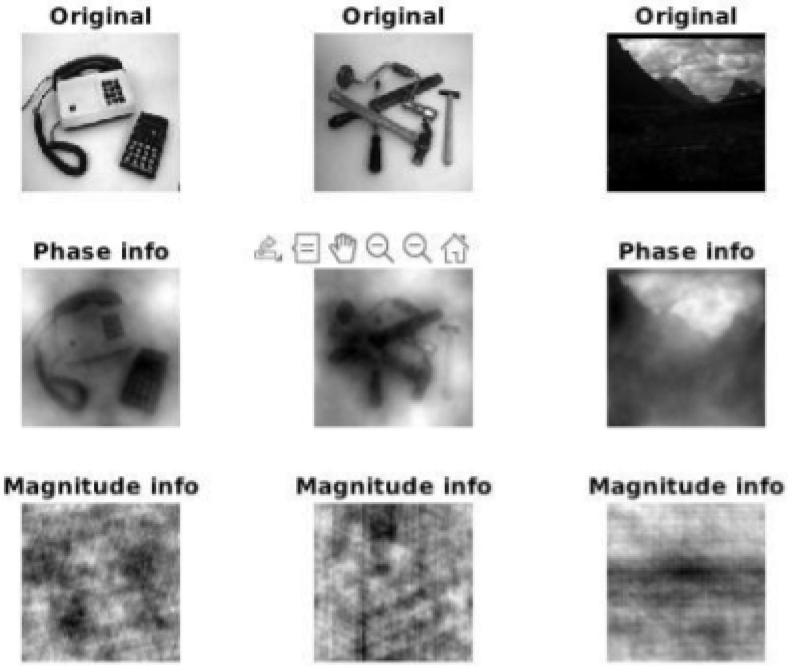


Figure 13. Images, their FT phase, and their FT magnitude.

To sum up, the magnitude is related to the intensity while the phase, to the location and contours.

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 response for each kernel is shown in Figure 14. We can observe the Gaussian expanding for increasingly higher values of the variance "t".
The variances are 2x2 diagonal matrices, where the diagonal values are, respectively for each t, as follows: 0.0133, 0.2811, 1.0, 10.0, and 100.0.
NB: The estimated variances for t<1 are close, yet not completely exact to the actual value.

Figure 14. Gaussian kernel impulse response.

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:

Values are identical for t>1 and t=1, i.e., the ideal continuous case. For t<1, however, values are close yet not exactly identical. When sampling for variance estimation, we must notice that the Gaussian functions with smaller variance "t" are thinner, i.e., the probability mass is concentrated in a smaller space and the spatial variations (i.e. frequencies) are greater (see Figure 15). Sampling higher frequencies becomes harder, and estimations are thus poorer.

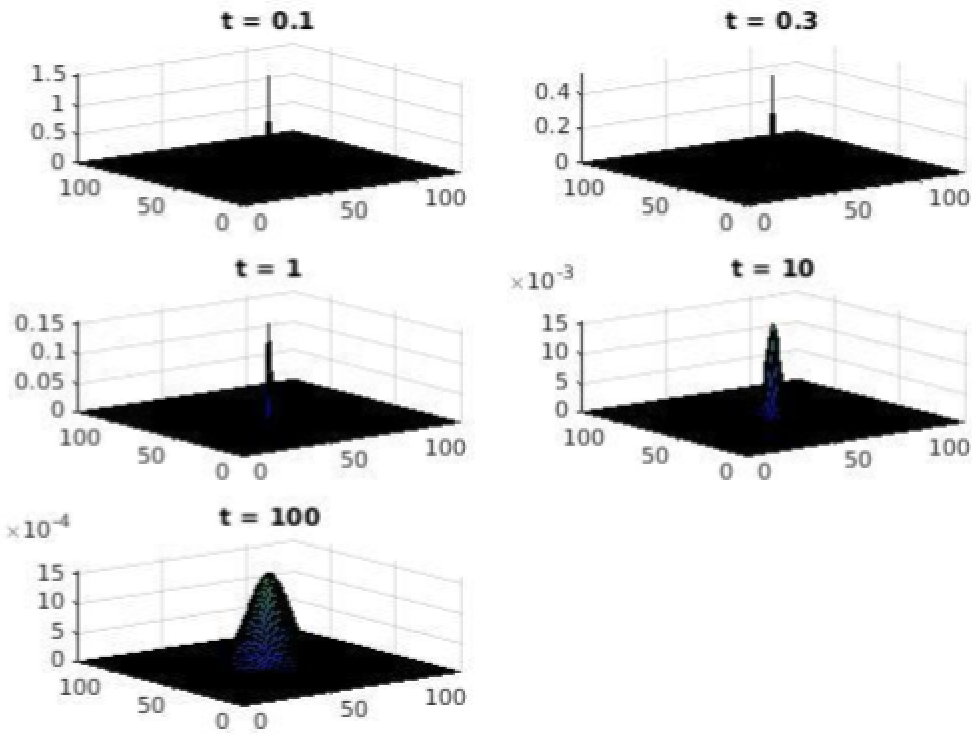


Figure 15. Gaussians functions for different values of variance "t".

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:

As depicted in Figure 16, choosing a higher variance "t" for Gaussian kernels will yield blurrier images, since higher frequencies shall be removed.
Although blurring might seem counterproductive at first, working with images having higher frequency does not necessarily have to be better. As a matter of fact, blurring or using an LPF before segmentation or edge detection is commonly a good practice.

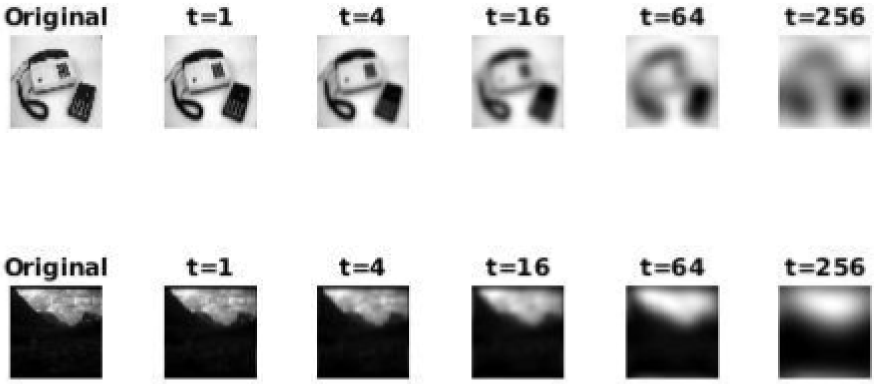


Figure 16. Applying Gaussian kernel with different variances.

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:

Figure 17 shows the two types of noise that we will be applying: *Gaussian* noise, and *salt-and-pepper* noise, which is formed by adding black and white randomly throughout the image.

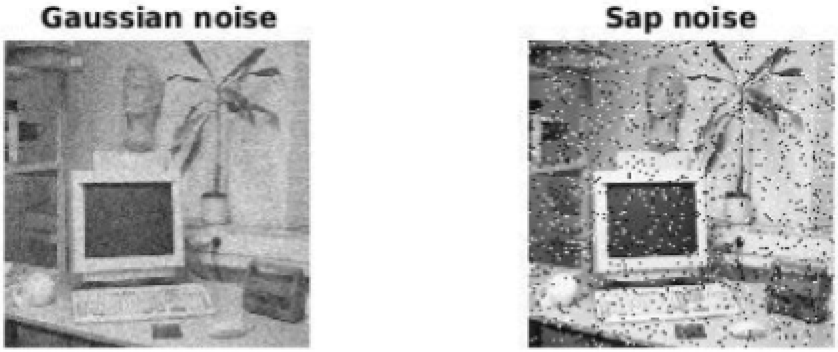


Figure 17. "Gaussian" and "sap" noise

In an attempt to correct Gaussian-noised images by applying a **Gaussian filter**, we observe (Figure 18a) optimal values for t=5 and t=10, where the image has been denoised, yet still conserved useful for, say, segmentation.
On the other hand, when using Gaussian filtering for sap-noised images, we observe (Figure 18b) a somewhat less desirable result. That is, images have been blurred (the higher the variance, the more blurring), yet the black and white noisy points have been spread to neighboring pixels and have therefore greatly influenced the rest of the picture.
To sum, Gaussian filters render blurred images with significantly less noise. Albeit very useful for reducing Gaussian noise, they are nonetheless less efficient for *salt and pepper* noise. Moreover, choosing a very high variance might lead to an image where edges are no longer recognizable.

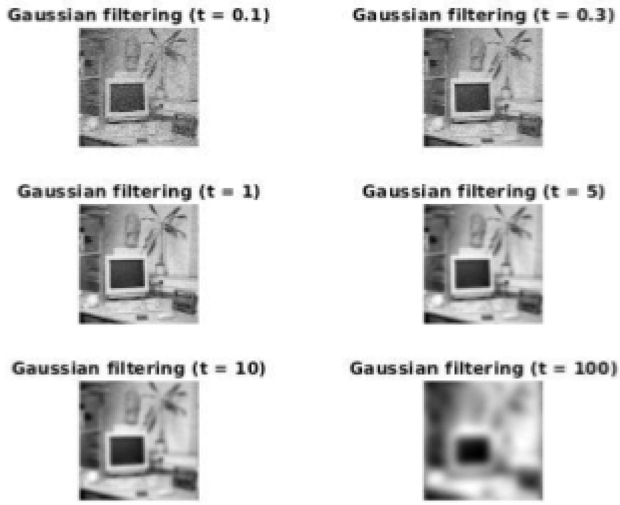


Figure 18a. Gaussian filtering with different variance parameters (Gaussian noise).

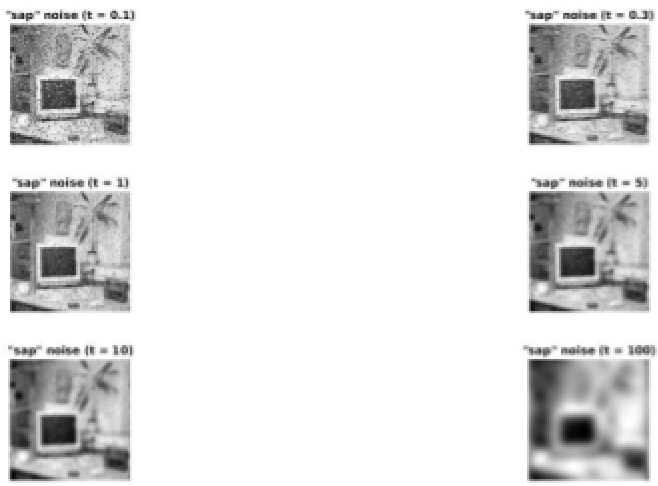


Figure 18b. Gaussian filtering with different variance parameters ("sap" noise).

Similarly, we apply a **Median filter**. We obtain Figure 19a, where we can observe a good performance upon Gaussian-noised images, yet might not as efficient as the Gaussian filtering in some aspects, since some objects are distorted. In Figure 19b, it is applied upon "sap"-noised images, and the results clearly overcome those previously obtained with a Gaussian filter.
To sum, median filtering has very good overall results on both noisy images- The higher the window, the more pixels are considered, and more noise is removed (although the image might also be oversimplified, and some details lost). Furthermore, it is able to preserve contours. However, median filter effects are known to be difficult to treat analytically. There is no error propagation.

Figure 19a. Median filtering with different window parameters (gaussian noise).

Figure 19b. Median filtering with different window parameters ("sap" noise).

Finally, we used the **ideal filter** with different cutoff frequencies. Results for Gaussian-noised images are shown in Figure 20a. The outcome is rather poor when compared to other filters, and quite huge distortions are observed. Likewise, when applied to "sap"-noised images, some distortions are also introduced into the image (Figure20b).

To sum, this filter might be easy to apply and can remove noise, yet it also renders highly distorted images. The lower the cutoff frequency, the more noise that is removed, but also more distortions will ensue.

Figure 20a. Ideal filtering with different cutoff frequencies (gaussian noise).

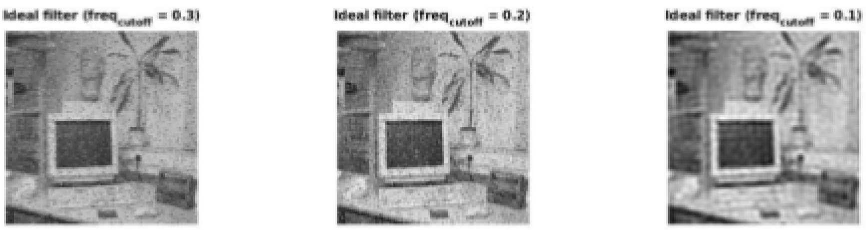


Figure 20b. Ideal filtering with cutoff frequencies ("sap" noise).

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

Answers:

Gaussian filtering has good results, especially for Gaussian noise. Higher variances will yield blurrier images, and hopefully more denoised.

Median filters have very good overall results, especially for *salt-and-pepper* noise. Choosing greater windows will decrease noise, but might also oversimplify our image.
Ideal filters have overall bad results in our examples. Choosing lower frequencies will remove more noise (which has high frequency), but also distorts our image to a quite great extent.

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 keeping one pixel out of two, we obtain a very pixelated image. When smoothing (applying a filter) before, we observe a more detailed image.

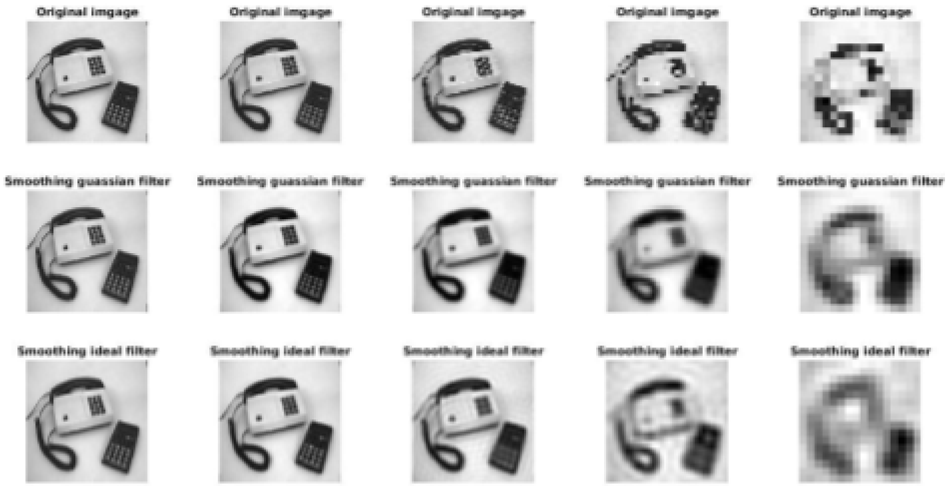


Figure 21. Smoothing with a *Gaussian* and an *ideal* filter.

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 an image with higher frequencies will be harder, and the outcome will be worse. However, if we smooth before subsampling, that is, removing high frequencies, the sampling method will prove more optimal and render better-looking images as a result. In other words, if we aim to *subsample*, we should choose a low sampling frequency (i.e. larger steps), thus keeping a lower number of samples. Using a low sampling frequency might lead us to disregard the *Nyquist Sampling Theorem*, leading to *aliasing*. For this reason, smoothing or LP filtering is a commonly good practice before subsampling.
