# Answers to questions in Lab 1: Filtering operations

Noé Samaille

November 2019

**Program** DD2423

**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!

## 1 Properties of the discrete Fourier transform

### 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** We observe different wavelength and angle depending on p and q. The magnitude remains the same for all cases. The wavelength of the real and imaginary parts is the same, but with a shift in phase of $\frac{\pi}{2}$ ($\sin(x + \frac{\pi}{2}) = \cos(x)$).

### 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.

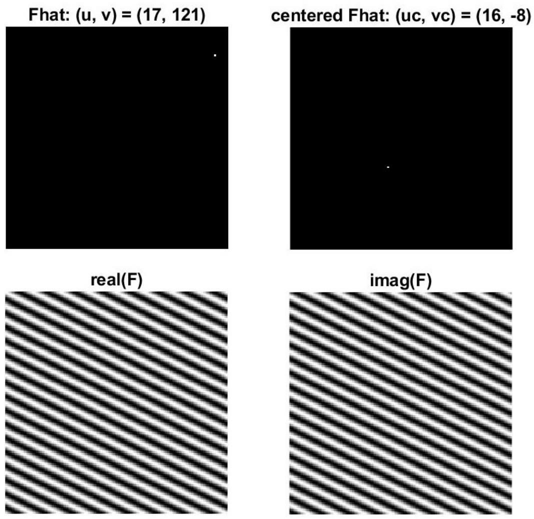


Figure 1: Example of a point in fourier domain and corresponding Fourier inverse.

**Answers** Using Equation (4), the inverse Fourier transform of $\hat{F}$ is

$$F(x) = \frac{1}{N} \sum_{u \in [0..N-1]^2} \hat{F}(u) e^{\frac{2\pi i u^T x}{N}}$$
$$F(x) = \frac{1}{N} e^{\frac{2\pi i (p,q)^T x}{N}}$$
$$F(x) = \frac{1}{N} \cos(\frac{2\pi (p,q)^T x}{N}) + i \frac{1}{N} \sin(\frac{2\pi (p,q)^T x}{N})$$

As shown in Figure.1, a position (p, q) in the Fourier domain is projected in the spatial domain as a complex. Its real and imaginary parts are sine waves with a angular frequency $\omega = \frac{2\pi}{N}(p,q)$. Phase is between $[-\pi; \pi]$ and equal to $\frac{2\pi i u^T x}{N}$.

## 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** Inverse fourier transform:

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

Only the pixel at (p, q) is equal to 1.

$$F(x) = \frac{1}{N} e^{\frac{2\pi i (p,q)^T x}{N}}$$

The amplitude is the modulus of $F(x)$ so $A = \frac{1}{N} = \frac{1}{sz} = \frac{1}{128}$.

## 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, the length of the sine wave $\lambda$ is related to $(p,q)$ centered and shifted to $(u_c, v_c)$ as following:

$$\lambda = \frac{2\pi}{||\omega||} = \frac{2\pi}{\sqrt{\omega_1^2 + \omega_2^2}} \tag{1}$$
$$\omega_1 = \frac{2\pi u_c}{N} = \frac{2\pi u_c}{sz} \tag{2}$$
$$\omega_2 = \frac{2\pi v_c}{N} = \frac{2\pi v_c}{sz} \tag{3}$$
$$\implies \lambda = \frac{2\pi}{\sqrt{(\frac{2\pi u_c}{sz})^2 + (\frac{2\pi v_c}{sz})^2}} \tag{4}$$
$$\lambda = \frac{sz}{\sqrt{u_c^2 + v_c^2}} \tag{5}$$

The direction of propagation of the sine wave is given by the vector $(u_c, v_c)$.

## 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** We can see the image in the Fourier domain as a pattern that is repeated infinitely over the plan as illustrated in Figure.2. The `fftshift` function places the smallest freqency at the origin, illustrated in the figure by moving from green coordinates (yellow image) to the red coordinates (blue image).

So when we pass the point in the center and either p, q or both exceed half the image size, we observe the following shifts and inverse transforms:

- $(64, 65)$ or $(65, 64)$: if either $p$ or $q$ exceeds half the image size then the corresponding $(u_c, v_c)$ changes region in the newly centered image (cf. Figure.2 and Figure.3).
- $(65, 65)$: the real and imaginary parts are perfect checkerboard. This point corresponds to a maximum frequency (and the minimum wavelength): after `fftshift`, it the furthest away from the origin at (-64, -64).

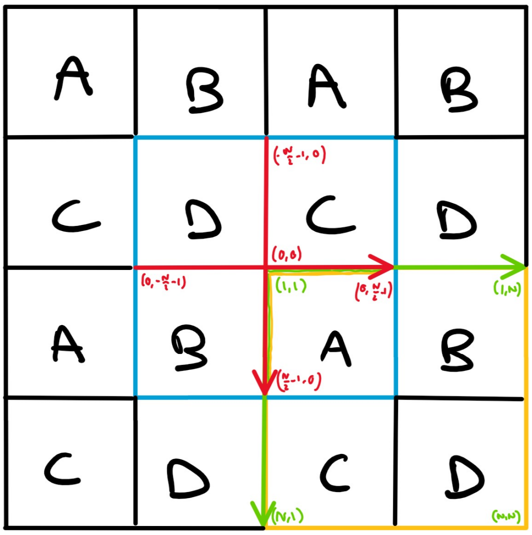


Figure 2: Effect of `fftshift`

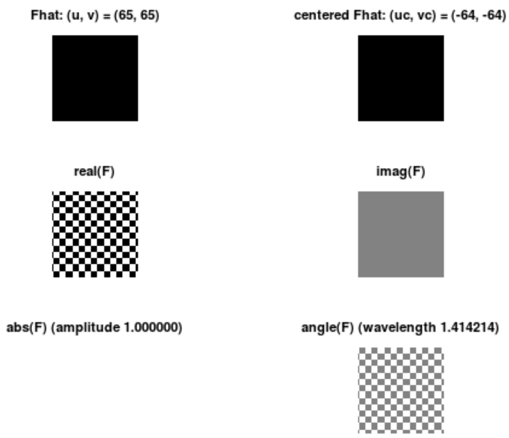


Figure 3: (p, q) = (65, 65)

## 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 theses instructions is to compute new coordinates $(u_c, v_c), u_c, v_c \in \{-\frac{N}{2} - 1, ..., \frac{N}{2} - 1\}$ of $(u, v), u, v \in \{1, ..., N\}$ after the `fftshift` function where minimum frequency is centered to the origin, e.g. in Figure.2 $(u_c, v_c)$ would be the red coordinates in the blue image of the point represented by green coordinates $(u, v)$ in the yellow image.

The displays of the phase and the real/im parts, with centered coordinates between $\frac{-sz}{2}$ and $\frac{sz}{2}$ so the values are shifted by $-sz$. (because of phase interval).

## 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** A mathematical interpretation to why the Fourier spectra are concentrated to the borders of the images can given by the derivation the 2D discrete Fourier transform (3).

$$\hat{F}(u, v) = \frac{1}{N} \sum_{x,y=0}^{128} F(x, y) e^{\frac{-2\pi i(ux+vy)}{N}}$$

Since only the rows from 56 to 71 contain ones, and we can separate the $x_s$ from the $y_s$, the formula is rewritten as:

$$\hat{F}(u, v) = \frac{1}{N} \sum_{x=56}^{71} e^{\frac{-2\pi i u x}{N}} \sum_{y=0}^{128} e^{\frac{-2\pi i v y}{N}}$$

$\sum_{y=0}^{128} e^{\frac{-2\pi i v y}{N}}$ is equal to 0 (sum of unity roots) for all columns except the first one.

Therefore, there is only one line of the first column.

More intuitively, the 'direction' of the line in fourier spectra is explained by the actual direction of variations in the spatial domain. The reason why the fourier spectra is concentrated at the border is that there is no variation through the $x$ axis in the spatial domain (black lines followed white ones then again black ones).

## Question 8

Why is the logarithm function applied?

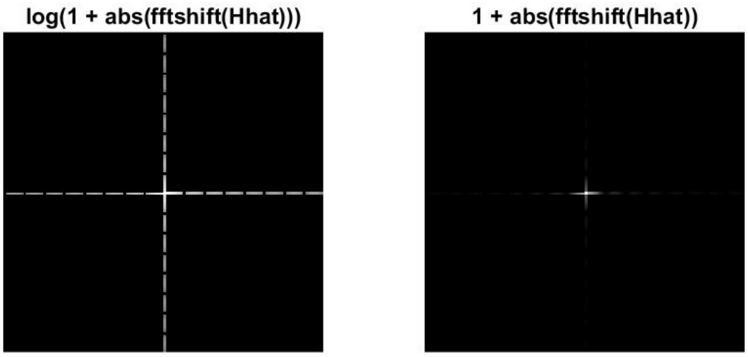


Figure 4: Example of a fourier spectra with (left) and without (right) use of logarithm function

**Answers** The logarithm function is applied because it helps compressing large dynamic range and make details visible, i.e. it exponentially reduces the Fourier spectrum, as shown in Figure.4.

## Question 9

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

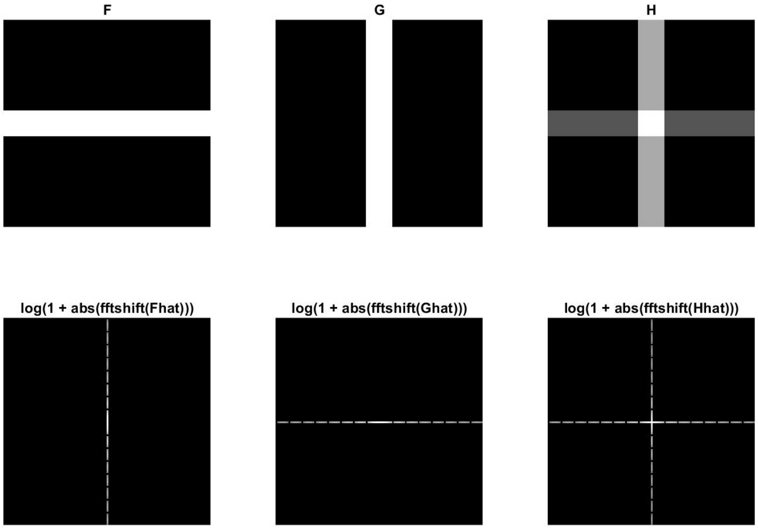


Figure 5: Illustration of the linearity property (question 9)

**Answers** Regarding linearity, we can graphically see (Figure.5) that the fourier transform of image H follows the property $\mathcal{F}(H) = \mathcal{F}(F + 2G) = \mathcal{F}(F) + 2\mathcal{F}(G)$. That is to say, $\hat{H}$ is the superposition of the lines obtained with $\hat{F}$ and $\hat{G}$. From these observations we can derive the expression in the general case: $\mathcal{F}(aF + bG) = a\mathcal{F}(F) + b\mathcal{F}(G)$.

## Question 10

Explain the results. 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.

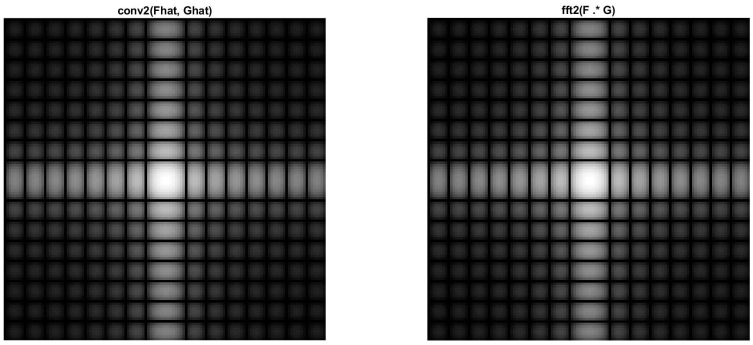


Figure 6: Application of the fact that multiplication in spatial domain is the same as convolution in Fourier domain

**Answers** The image $F * G$ can be seen as a bidimensional step function. The waves propagating along $x$ and $y$ correspond to the absolute value of the sinc function in both direction (the Fourier transform of a step function is a sinc). Multiplication in spatial domain is the same as convolution in Fourier domain, and vice-versa: $\mathcal{F}(fg) = \mathcal{F}(f) * \mathcal{F})(g)$. Figure.6 is the result of computations for question 10, it shows that this relation holds in this case.

## 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.

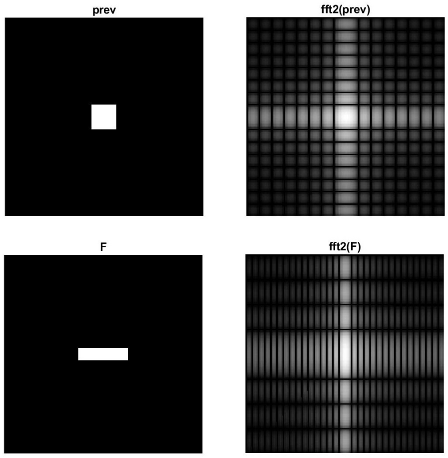


Figure 7: Illustration of the effect of scaling (question 11)

**Answers** By comparing the results with those in the previous question (see figure.7), we can conclude that scaling in the spatial domain results in scaling the Fourier domain, and expansion of an axis in the spatial domain results in compression in the Fourier domain, 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.

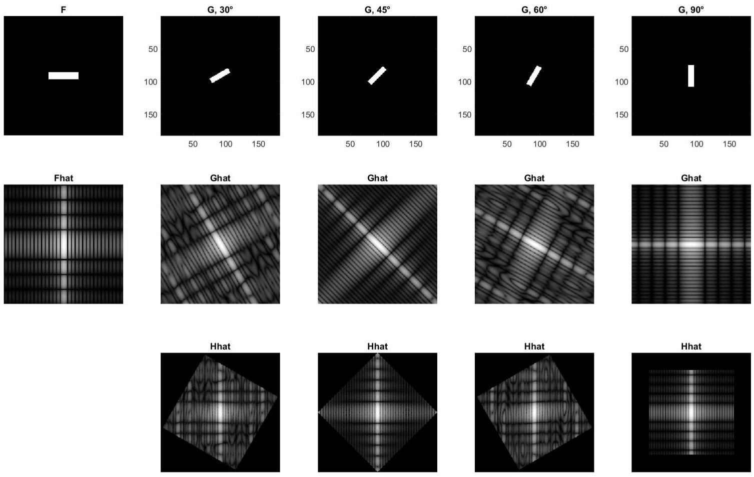


Figure 8: Effect of rotation (question 12)

**Answers** Figure.8 shows that the rotation of the image in the spatial domain induces a rotation by the same angle in the Fourier domain. It seems pretty intuitive since the frequencies are the same, in a different direction defined by the angle of the rotation. We can notice some distortions for $\alpha = 30°$ and $\alpha = 60°$, probably because these angles give irregular edges to the rotated rectangle.

## Question 13

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

**Answers** When the magnitude of the image is changed using `pow2image`, the image becomes "cloudy" but the edges remain (see Figure.9). We can assume phase is responsible for these edges. This is confirmed with the randomization of the phase: the images become complete noise. Our assumption is that the magnitude gives the average color intensity of the image.

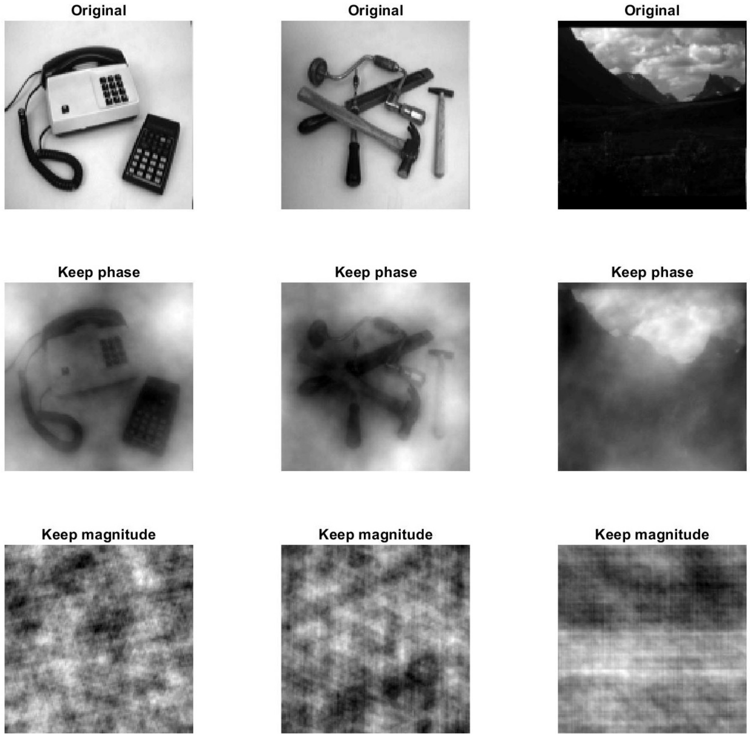


Figure 9: Information in Fourier phase and magnitude

## 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 $100.0$ ?

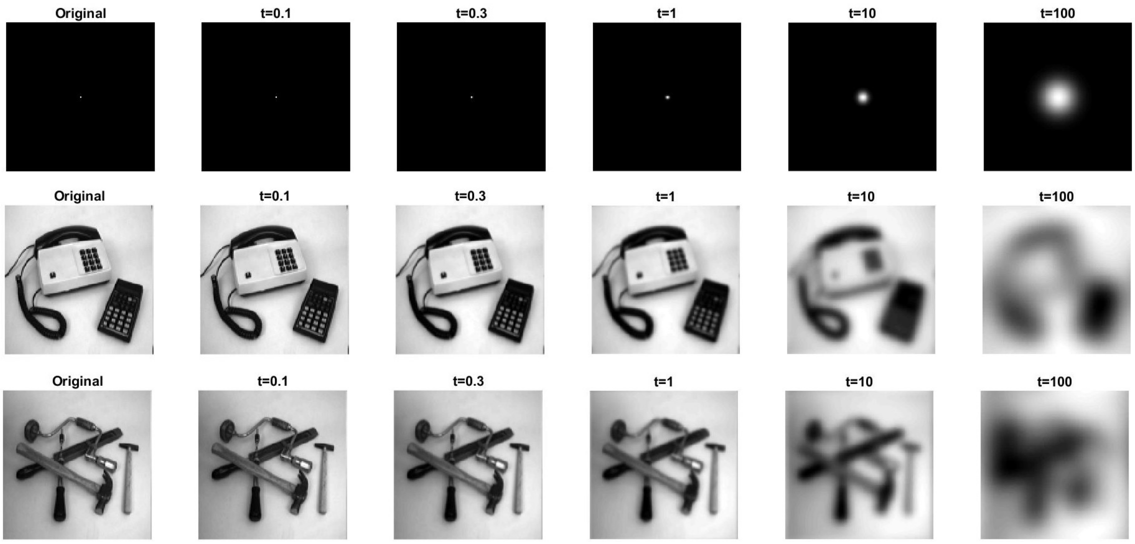


Figure 10: Impulse response of gaussian filter for different t-values.

**Answers** The first row of Figure.10 shows the impulse response of our discretized Gaussian kernel for the requested t-values. We can see that the point in the center seem to spread as $t$ increases, which seems to fit a gaussian behaviour with increasing width. Corresponding variances are shown in Table.1.

| t | Variance |
|---|---|
| 0.1 | $\begin{pmatrix} 0.0133 & 0 \\ 0 & 0.0133 \end{pmatrix}$ |
| 0.3 | $\begin{pmatrix} 0.2811 & 0 \\ 0 & 0.2811 \end{pmatrix}$ |
| 1 | $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ |
| 10 | $\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$ |
| 100 | $\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$ |

Table 1: Variances of discretized gaussian kernel for different t-values (question 14)

## 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 results (shown in Table.1) are quite similar to the estimated variance: for $t \geq 1.0$, the result correspond exactly to the ideal continuous case for which the variance is $t$ multiplied by the identity matrix. For $t < 1.0$, our variances are close but not equal to the ideal ones, which we think is due to the fact that we discretized the gaussian convolution by getting a sampled version of the Gaussian function, so in low variance of the gaussian (i.e low $t$), we can see the 2D gaussian as something very steep with a pic so sampling such gaussian gives a very bad approximation.

## 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 results are shown in the 2 last rows of Figure.10. We can observe that the result is more blurred as $t$ increases, which seems reasonable since $t$ represents the variance of the gaussian function (i.e. the bigger $t$ is, the more high frequencies will be removed in the result).

## 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** Our results with the different filters to reduce Gaussian noise reduction are shown in Figure.11 for different values of the filter parameters. Those on salt and pepper noise are shown in Figure.12. We can observe that the 3 filters behaves differently:

- **Gaussian smoothing**:
  - Positive effects: has a much more blurring effect than the 2 others. Surfaces of uniform colors are smoothed, reducing the gaussian noise in these area.
  - Negative effects: it just blurs the noise with neighbour pixels when dealing with salt and pepper noise, it does not remove it.
  - The result is more blurred as $t$ increases, and we lose sharp edges.
- **Median filtering**:
  - Positive effects: performs well on salt and pepper noise (removes salt/pepper pixels by computing the median of the window). Edges are also preserved.

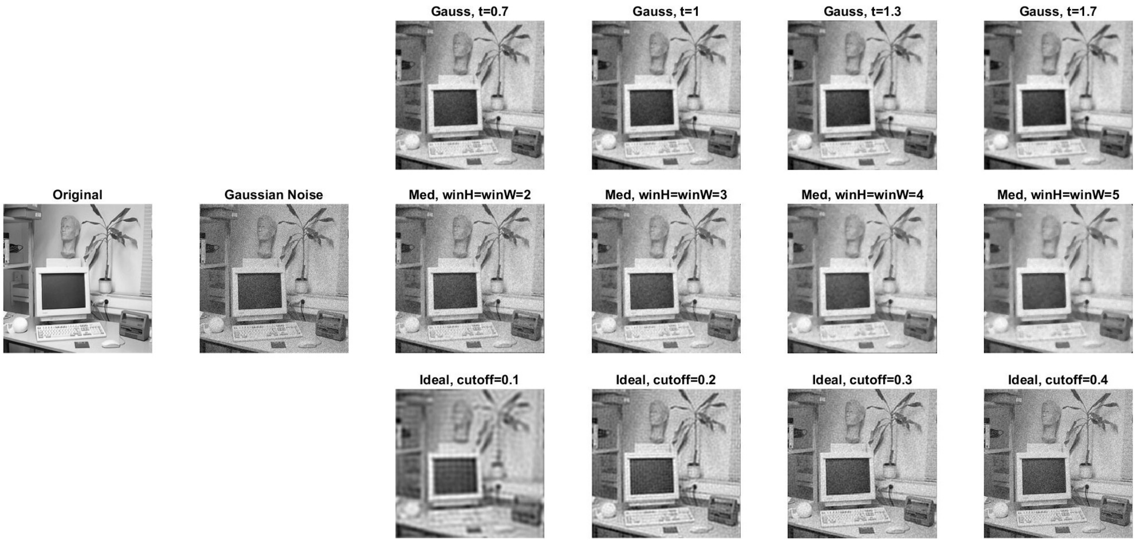


Figure 11: Results of Gaussian smoothing ($1^{st}$ row), median filtering ($2^{nd}$ row) and ideal low-pass filtering ($3^{rd}$ row) to reduce Gaussian noise.


Figure 12: Results of Gaussian smoothing ($1^{st}$ row), median filtering ($2^{nd}$ row) and ideal low-pass filtering ($3^{rd}$ row) to reduce Salt and Pepper noise.

  - Negative effects: lose information on colors as the window size increases (the result end up looking like a painting as we lose visible details).
  - The result looks more like a painting as the window size increases.

- **Ideal low-pass filter**:
  - Positive effects: remove frequencies above the cutoff frequency, smoothing to a limited extend gaussian noise
  - Negative effects: Smooth noise but does not remove it (keeps salt and pepper noise).
  - As the cut-off frequency decreases, the number of removed frequencies increases (as more frequencies are above the cut-off) therefore the image goes smoother.

## Question 18

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

**Answers** Based on the results shown in Figures 11 and 12 as well as the previous question we can conclude that the Gaussian smoothing method performs better on Gaussian and median filtering performs better when dealing with 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 $i = 4$.

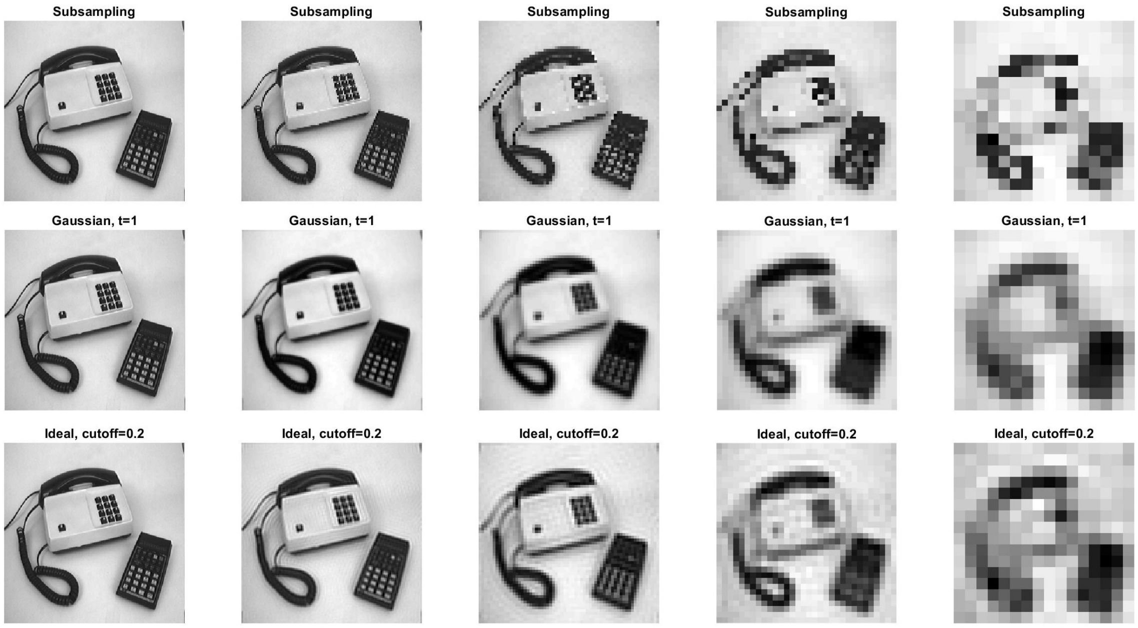


Figure 13: Subsampling ($1^{st}$ row) vs smoothing-subsampling ($2^{nd}$ and $3^{rd}$ rows) processes

**Answers** Figure.13 compares the subsampling process with and without prior smoothing using Gaussian smoothing with $t = 1$ and ideal low-pass filtering with a cut-off frequency $c = 0.2$ (our best results). We can observe that applying smoothing before subsampling reduces the aliasing artifacts on the smaller image, giving much better results after a few iteration, e.g. for iteration $i = 4$ (fourth column) we can recognize the phone and the shape of the calculator on smoothed versions while we cannot with just subsampling.

## 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** We can conclude that smoothing combined with subsampling reduces information loss: we know from the sampling theorem that the minimum sampling rate should be greater or equal than twice the highest frequency to retrieve completely the signal from its samples. As smoothing reduces the highest frequencies in the images, combining it with subsampling reduces the minimum sampling rate required to retrieve the original signal from its samples.