Answers to questions in Lab 1: Filtering operations

November 17, 2023

Name: Leonardo De Clara

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?

Answer:
Coordinates p and q determine the frequency (therefore the wavelength) and the direction of the sinusoid. Coordinates closer to the zero frequency (0,0) will be characterized by small frequency and bigger wavelength, while points further from (0,0) will have higher frequency, therefore smaller wavelength. This property can be noticed when comparing the results obtained, for example, with coordinates (5,1) and (17,9): since (5,1) is much closer to the original than (17,9) it results in a lower frequency wave with much wider wavelength.

The magnitude is not affected by the placement of coordinates, since it does not depend on the specific sinusoid, while the phase does.

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.

Answer:

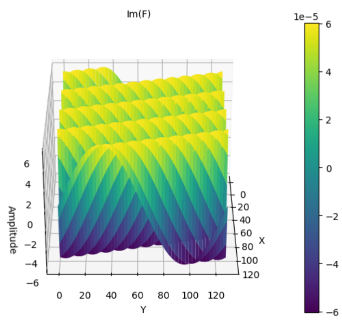
A specific position (p,q) in the discrete Fourier domain can be projected back as a sine wave in the spatial domain applying the inversion theorem:

$$f(x, y) = \frac{1}{N^2}\hat{f}(u, v)e^{(2\pi i(\frac{xu}{N}+\frac{yv}{N}))}$$

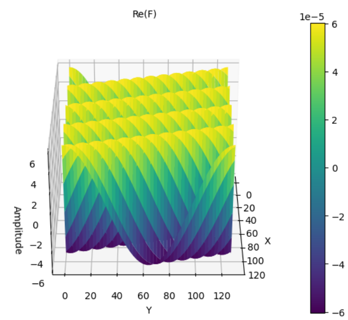
Once Euler's formula is applied the result obtained is a complex sinusoidal component:

$$f(x,y) = \frac{1}{N^2}\hat{f}(u,v)(cos(\frac{2\pi(px+qy)}{N}) + isin(\frac{2\pi(px+qy)}{N}))$$

Taking, for example, (p,q)=(4,1) the result obtained when projecting it back in the spatial domain is showed in the figures below.



(a) Imaginary part for (p,q)=(4,1)



(b) Real part for (p,q)=(4,1)

Question 3.

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

Answer:

From lecture notes (3)
$f(x,y) = \frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\hat{f}(u,v)e^{(2\pi i(\frac{xu}{N}+\frac{yv}{N}))}$ (Inversion theorem),
$e^{(2\pi i(\frac{xu}{N}+\frac{yv}{N}))} = cos(\frac{2\pi(xu+yv)}{N}) + isin(\frac{2\pi(xu+yv)}{N})$ (Euler's formula),
$|\hat{f}(u,v)| = |\sqrt{Re^2[\hat{f}(u,v)] + Im^2[\hat{f}(u,v)]}|$ (Complex module),
The amplitude A of f is given by

$$A = \frac{1}{N^2}\max_{u,v}(|\hat{f}(u,v)|)$$

In the case of

$$\hat{f}\,(u,v) = \begin{cases} 1, & (u,v) = (p,q) \\ 0, & otherwise \end{cases}$$

and N = 128 then $A = 0,000061$.

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.

Answer:
From lecture notes (3) and the lab description

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

where

$$\omega_1 = \frac{2\pi p}{N}\ ,\ \omega_2 = \frac{2\pi q}{N}$$

After centering $(p, q) = (u_c, v_c)$, thus

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

The sine wave direction can be computed from previous equations and the angle will be $atan(\frac{u_c}{v_c})$

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!

Answer:

By using numpy's function fft.fftshift() the zero frequency component is shifted to the center of the spectrum by swapping the 1st quadrant with the 3rd and the 2nd with the 4th, therefore the new position of point (p,q) must be computed with respect to the new origin. Given an image of size (N,N) point (p,q) is shifted if it is placed over the center, thus if either p > N/2 or q > N/2.

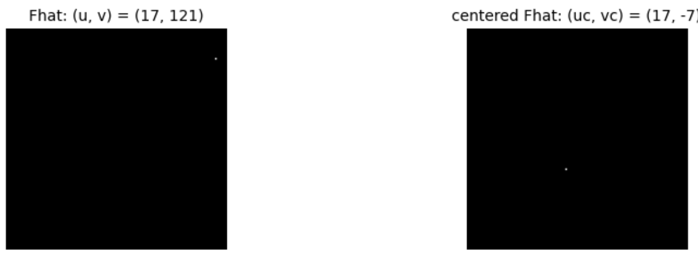


Figure 2: (p,q)=(17,121)

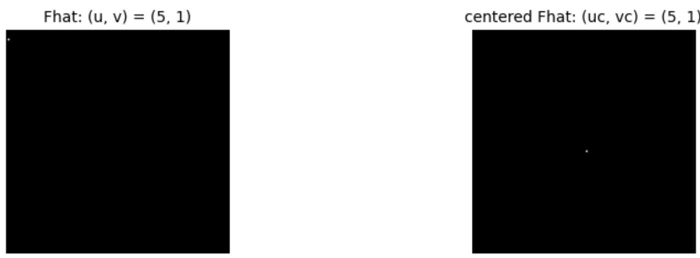


Figure 3: (p,q)=(5,1)

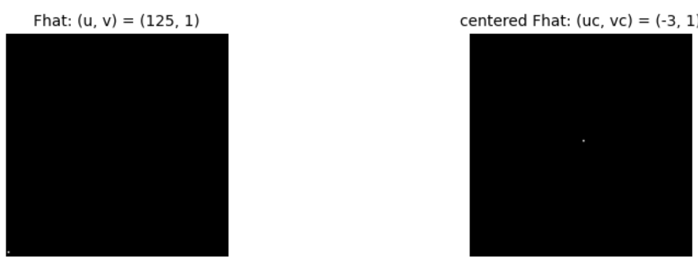


Figure 4: (p,q)=125,1

Question 6.

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

Answer:
The purpose of these instructions is to compute the new position of the point associated with coordinates (p,q) in order to obtain the correct relative

position to the origin after moving Fhat's zero component to the center of the image by using numpy's fft.fftshift().

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!

Answer:
Focusing on the first case (horizontal white band) and given that

$$\hat{f}\,(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \hat{f}\,(x,y)\, e^{\left(-2\pi i\left(\frac{xu}{N}+\frac{yv}{N}\right)\right)}$$

we can simplify it given modularity's property and knowing that

$$f(x,y) = \begin{cases} 1, & 55 < x \leq 71 \\ 0, & otherwise \end{cases}$$

thus obtaining

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

Since the image's spectrum assumes value equal to 1 when v = 0 and 0 otherwise we can simplify the expression even further using

$$\delta(v) = \begin{cases} 1, & v = 0 \\ 0, & v \neq 0 \end{cases}$$

and setting

$$\sum_{y=0}^{N-1} e^{-2\pi i \frac{yv}{N}} = \delta(v)$$

Thus

$$\hat{f}\,(u,v) = \sum_{x=56}^{71} e^{-2\pi i \frac{xu}{N}}\,\delta(v)$$

We can see that the spectrum will not be zero only when $v = 0$, therefore all non-negative values will be placed on the left border. The same reasoning can be applied to G to show why all non-negative values will be placed on the top border. As for H, being a linear combination of F and G, its spectrum will be a linear combination of F and G's spectra.

Question 8.

Why is the logarithm function applied?

Answer:

Logarithm function is applied to enhance the lowest intensity frequencies and reduce the dynamic range. In this way the the absolute values of gray values and difference between the highest and lowest values become smaller since the logarithm function will even out the distribution of the pixels, distributing them in a larger range, thus increasing the contrast of the image.

Question 9.

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

Answers: H is built as a linear combination of images F and G in the spatial domain $H = F + 2G$. By visually observing the plots we can also say that the linear combination $DFT(F) + DFT(2G)$ results in the DFT of H obtained above. As a matter of fact, the Fourier transform is a linear operator:

$$\mathcal{F}(aF + bG) = a\mathcal{F}(F) + b\mathcal{F}(G)$$

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.

Answer:
According to the convolution theorem a convolution in the spatial domain is the same as multiplication in the Fourier (frequency) domain.

$$\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g)$$

The viceversa is also true: A convolution in the Fourier domain, is the same as multiplication in the spatial domain.

$$\mathcal{F}(fg) = \mathcal{F}(f) * \mathcal{F}(g)$$

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.

Answer:
The resulting image can be seen as a scaled version of the previous one, where a compression ($Sx = 1/2$) has been applied to the x-dimension and an expansion ($Sy = 2$) to the y one. That is $f_{new} = f_{old}(\frac{x}{2}, 2y)$. Compression in the spatial domain represents expansion in the Fourier one and vice versa, therefore $\hat{f}_{new}(u, v) = \hat{f}_{old}(2u, \frac{u}{2})$.

In the direction where the bar is longer we have a slow change, therefore the spectrum will be characterised by low frequencies. In the direction where the bar is shorter, instead, we have a more rapid change, therefore the spectrum will be characterised by higher frequencies.

Question 12.

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

Answer: A rotation of the image results in a rotation of its Fourier representation by the same angle. Given a rotation defined in the cartesian domain by matrix

$$M(\alpha) = \begin{pmatrix} cos(a) & -sin(a) \\ sin(a) & cos(a) \end{pmatrix}$$

such that $g(x, y) = f(xcos(a) - ysin(a), xsin(a) + y(cos(a)))$, its Fourier transform $\hat{g}(u, v)$ will be

$$\hat{g}(u, v) = \hat{f}(ucos(a) - vsin(a), usin(a) + v(cos(a)))$$

This result is clear when observing the plots of the rotated version of the image's spectrum. However small distortions in the spatial domain due to the fact that the rotated image cannot be represented perfectly appear to be clearer in the Fourier domain, leading to more high frequencies.

Question 13.

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

Answer: The magnitude spectrum represents the distribution of different frequencies present in the image, therefore what grey-levels will end up in the image, while the phase spectrum contains information about the spatial relationships between pixels in the image, thus defining where edges will end up in the image. In the output of pow2image() we still see the edges and contours of the image, but the magnitude is changed. In the randphaseimage() case, on the other hand, we can no longer see any contours of the original image, while the magnitude is not affected. The ratio between bright and dark pixels is unchanged but with totally different phase values there is no information about where they should be located.

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?

Answer:

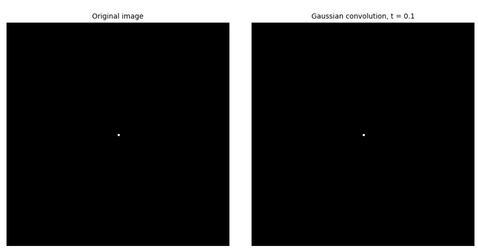


Figure 5: Gaussian kernel, t=0.1

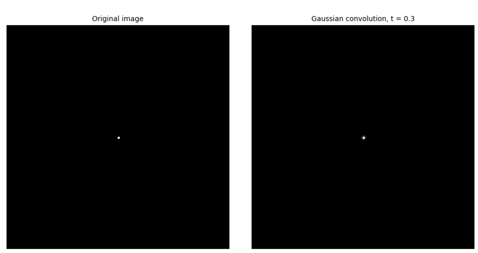


Figure 6: Gaussian kernel, t=0.3

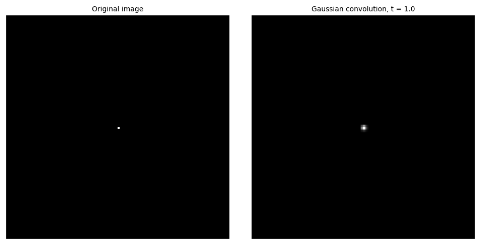


Figure 7: Gaussian kernel, t=1.0

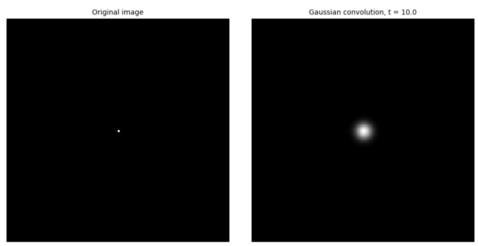


Figure 8: Gaussian kernel, t=10.0

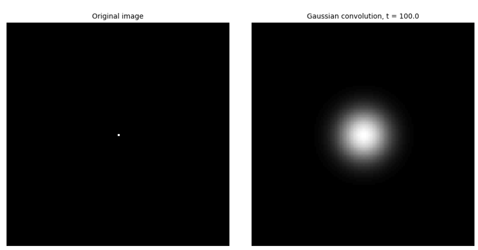


Figure 9: Gaussian kernel, t=100.0

Variance	Covariance matrix
0.1	$\begin{bmatrix} 0.013 & 0.0 \\ 0.0 & 0.013 \end{bmatrix}$
0.3	$\begin{bmatrix} 0.281 & 0.0 \\ 0.0 & 0.281 \end{bmatrix}$
1.0	$\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$
10.0	$\begin{bmatrix} 10.0 & 0.0 \\ 0.0 & 10.0 \end{bmatrix}$
100.0	$\begin{bmatrix} 100.0 & 0.0 \\ 0.0 & 100.0 \end{bmatrix}$

Table 1: Approximated covariance matrices

High t-values lead to a wider impulse response, resulting in an higher variance result, therefore a blurrier picture. The Gaussian kernel in the frequency domain acts as a low-pass filter, allowing lower frequencies to pass through while attenuating higher frequencies: high-frequency details in the image, which represent rapid changes, are effectively averaged out by the filter.

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.

Answer:

Variance	Absolute different between matrix
0.1	$\begin{bmatrix} 8.67032748e-02 & 1.95676808e-14 \\ 1.95676808e-14 & 8.67032748e-02 \end{bmatrix}$
0.3	$\begin{bmatrix} 1.89461699e-02 & 2.00117700e-14 \\ 2.00117700e-14 & 1.89461699e-02 \end{bmatrix}$
1.0	$\begin{bmatrix} 2.11227226e-07 & 5.60662627e-15 \\ 5.60662627e-15 & 2.11227189e-07 \end{bmatrix}$
10.0	$\begin{bmatrix} 1.92201810e-12 & 2.88657986e-15 \\ 2.88657986e-15 & 2.00905959e-12 \end{bmatrix}$
100.0	$\begin{bmatrix} 6.71768944e-07 & 6.35047570e-14 \\ 6.35047570e-14 & 6.71768717e-07 \end{bmatrix}$

Table 2: Approximated covariance matrices

For small values of t such as 0.1 and 0.3 the difference between estimated covariance and the ideal continuous case is significant: this result could be due to the fact that for t < 1 the filter is a spike in the spatial domain and its Fourier transform is a wide spectrum, which may lead to having some values being cut out, thus not being able to fully represent the signal and leading to suboptimal estimation of the variance. For larger t values it becomes negligible, thus converging to the true Gaussian distribution.

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?

Answer:

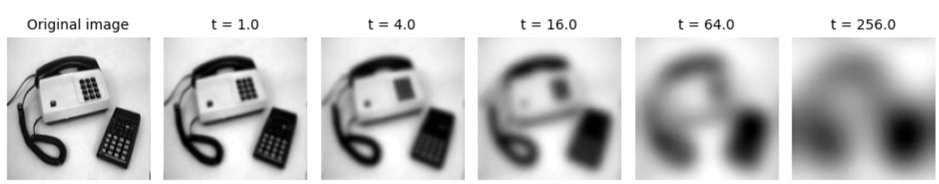


Figure 10: Gaussian filter applied to image phonecalc128

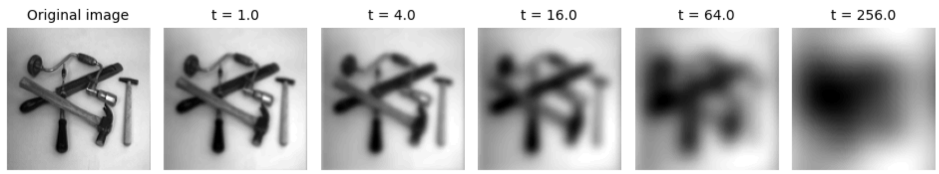


Figure 11: Gaussian filter applied to image few128

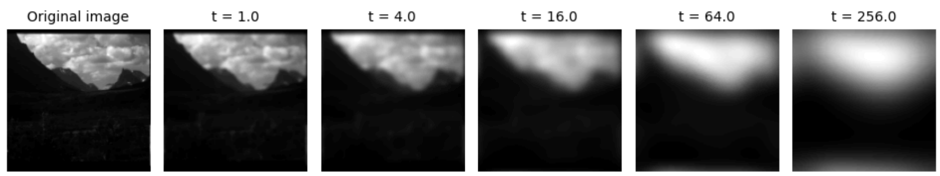


Figure 12: Gaussian filter applied to image nallo128

As the results show, when the t-value increases the image becomes more and more blurred. The main reason is that larger t-value are responsible for filtering more high frequencies by reducing the cut-off frequency. Since higher frequencies are responsible for most of the information regarding edges, the application of a Gaussian filter results in smoother and therefore images. While smaller variance values lead to localized smoothing, higher variance values are responsible for deeper smoothing, leading to loss of details.

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).

Answer:

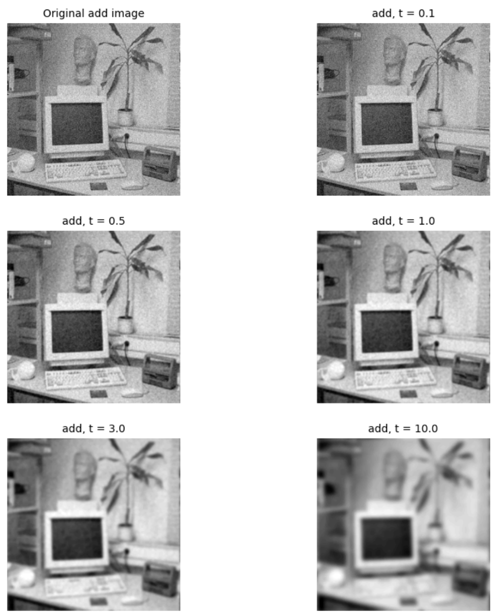


Figure 13: Gaussian smoothing, additive noise

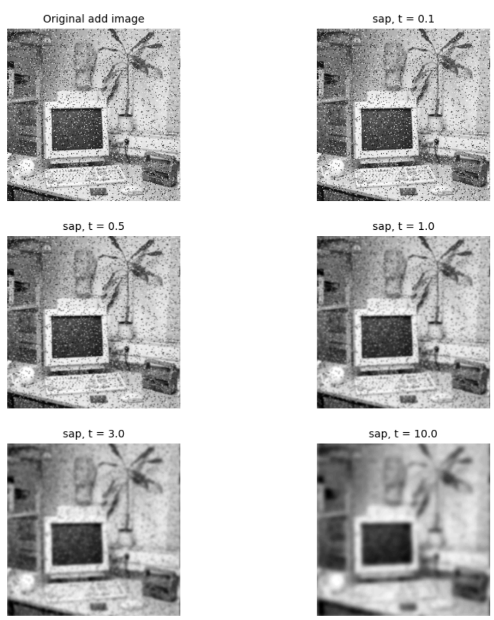


Figure 14: Gaussian smoothing, salt and pepper noise

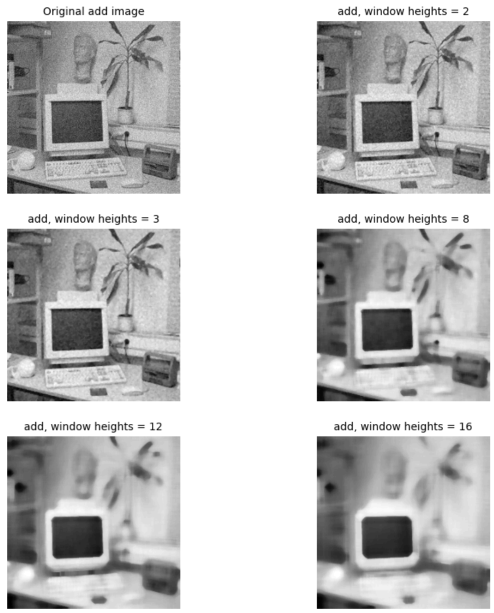


Figure 15: Median filtering, additive noise

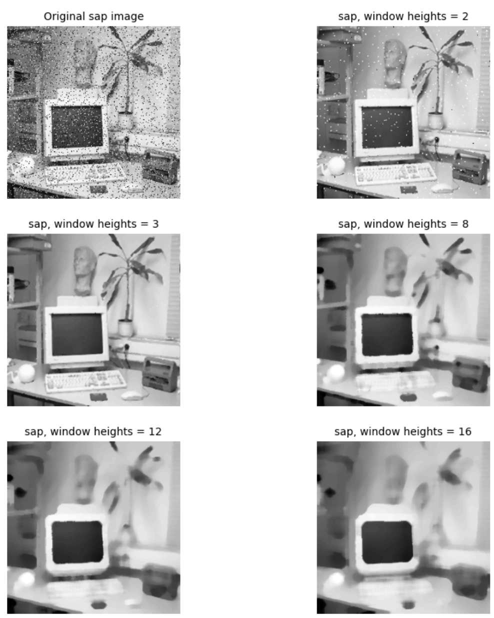


Figure 16: Median filtering, salt and pepper noise

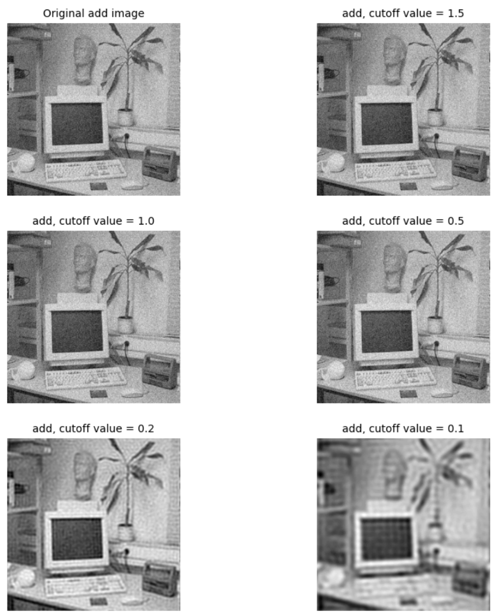


Figure 17: Ideal low pass filtering, additive noise

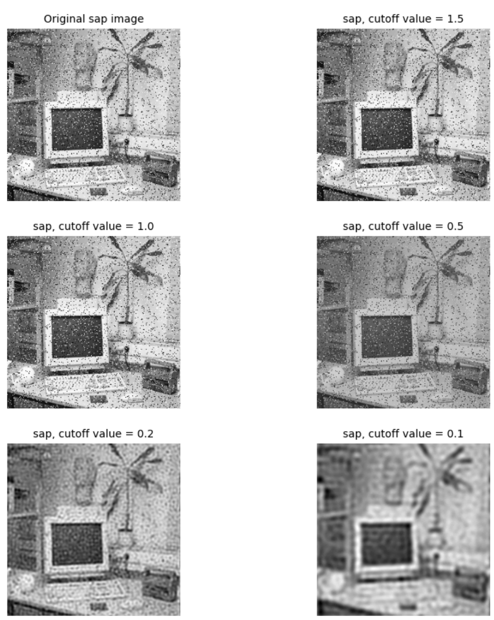


Figure 18: Ideal low pass filtering, salt and pepper noise

Guassian smoothing:

- It is able to properly smooth the image;
- For tuned values of t it handles Gaussian noise well;
- Edges tend to be blurred out easily even with low t-values;
- It's not able to reduce sap noise for any value of t.

Median filtering:

- It provides similar performances to guassian filter when applied to additive noise image;
- It's able to significantly reduce sap noise (as it eliminates extreme values);
- Non distint features are easily filtered out ;
- Image looks "painting-like" when the window size is increased;

Ideal low-pass filtering

- Application is computationally easy
- It's not able to properly handle neither additive noise nor salt and pepper;
- Image is distorted image when the cutoff value is too low (ringing effect)

Question 18.

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

Answer :
The conclusions that can be drawn from these tests are that gaussian smoothing is able to provide good performance on Gaussian noise, but it can't properly manage sap noise. Median filtering, on the other hand, delivers the best performance overall by preserving edges in both cases, but the window size needs to be tuned in order to avoid paining-like images. At last, the ideal

low pass is not able to reduce neither kind of noise and its results significant differe when changing cutoff values (even by a small margin).

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.

Answer:

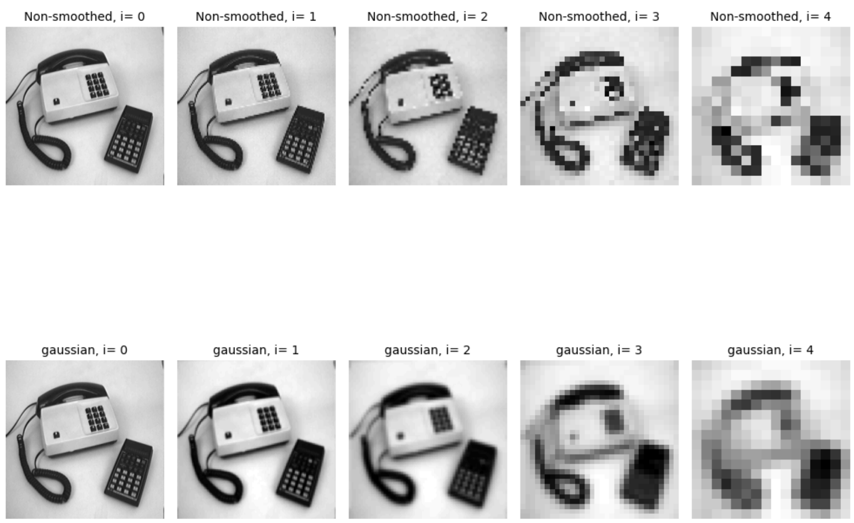


Figure 19: Gaussian smoothing, t = 1.1

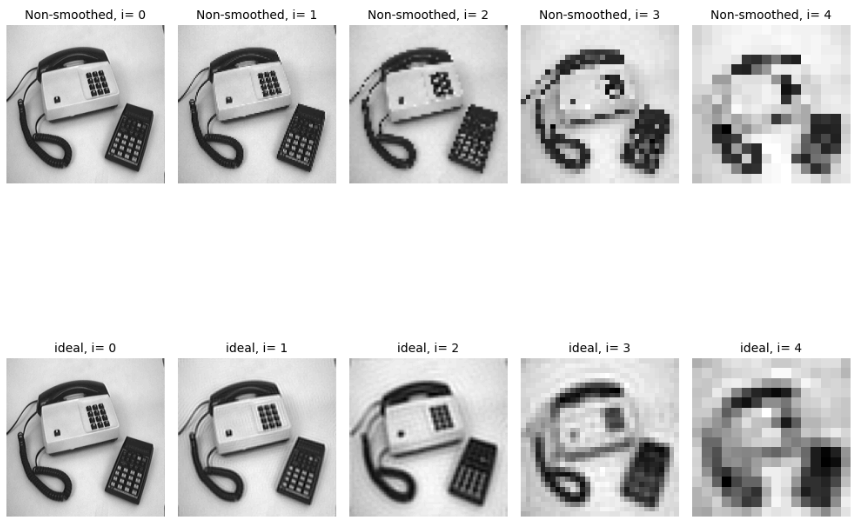


Figure 20: Ideal low-pass filter, cutoff frequency = 0.2

Raw subsampling results in a more pixelated appearance, especially in regions with high-frequency details, while smoothed and then sampled versions tends to produce visually smoother images with reduced noise. However the filter's parameters need to be properly tuned, otherwise smoothing could lead to worse results (eg. too high variance with gaussian filter or too low cutoff frequency with ideal low pass filter). When using the ideal low pass filter we risk to obtain a ringing effect that negatively affects the sampling process, leading to poor quality results, while the gaussian filter is able to smooth uniformly the image thus helping when 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.

Answer:

By subsampling with rawsubsample() we reduce the size of the image by a factor of two in each dimension, i.e. by picking out every second pixel along each dimension. By smoothing the image before to the subsampling we can prevent losing information from the image and encountering aliasing. As a matter of fact, by blurring the image we can effectively reduce reduce the maximum frequency and decrease the Nyquist rate to better match the new subsampled image. The required frequency will need to be $f_N > 2f_{max}$