Answers to questions in Lab 1: Filtering operations

Name: CARON Arthur & LASSARAT Côme Program: Msc Systems, Control and Robotics

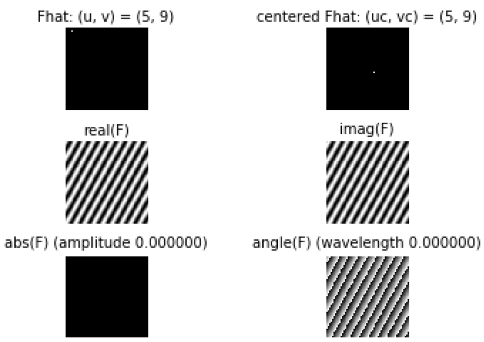
Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

Good luck!

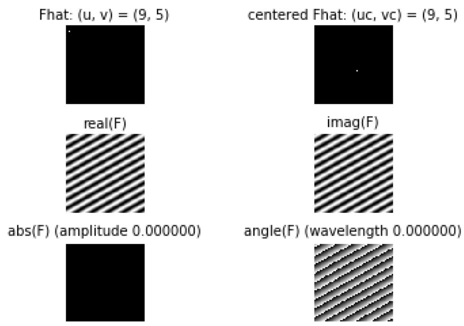
Question 1: Repeat this exercise with the coordinates p and q set to (5, 9), (9, 5), (17, 9), (17, 121), (5, 1) and (125, 1) respectively. What do you observe?

Answers:

```
In [14]: fftwave(5,9)
```

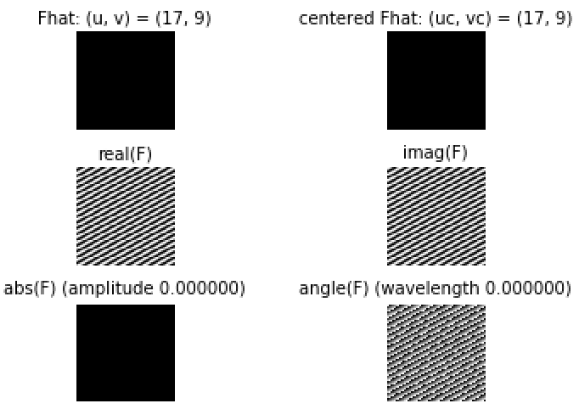



```
In [15]: fftwave(9,5)
```

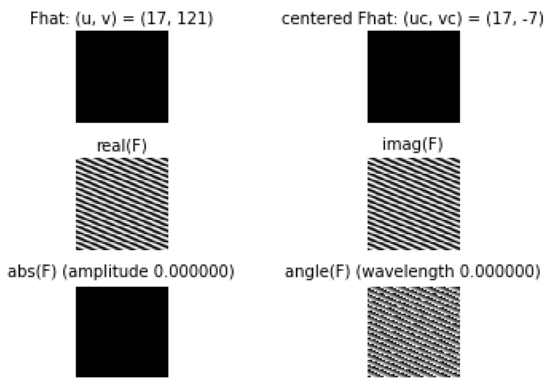


u and v are exchanged, like in real(F) and imag(F)

```
In [16]: fftwave(17,9)
```



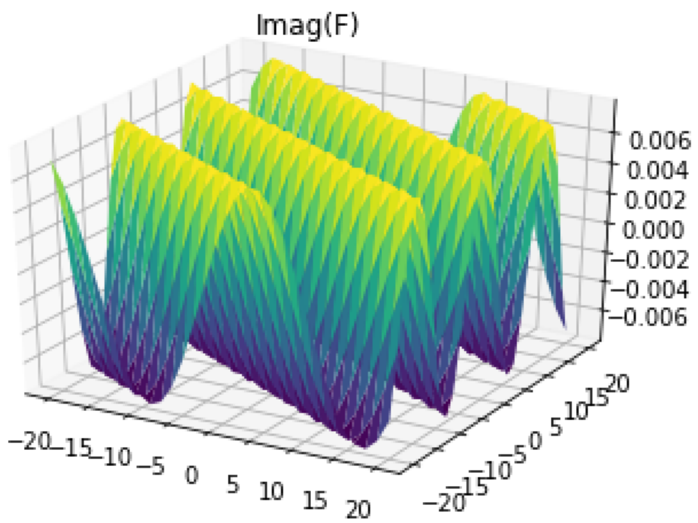

```
In [17]: fftwave(17,121)
```



→ Symmetry according to the vertical axis IF well centered (ie. need to compare (17,7) and (17,121)

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:



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 expression from equation 4 highlights an amplitude equal to

$$\frac{1}{N} * \max(\mathrm{abs}(\hat{}F(\mathrm{u}))$$

but here, $\max(\mathrm{abs}(\hat{}F(\mathrm{u})) = 1$

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 direction of the sine wave will be the direction of the vector $(p\ q)^T$

The length of the sine wave depends on the angular frequency thus it depends on p and q.

$$\lambda = \frac{2\pi}{\sqrt{\omega1^2+\omega2^2}} = \frac{2\pi * N}{\sqrt{p^2+q^2}}$$

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:

We know that the fourier transform follows the conjugate symmetry and periodicity property, thus we can calculate the fourier transform between [-N/2;N/2] thus when p or q exceeds half the image size(p>N/2) we don't have to calculate its fourier transform.

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

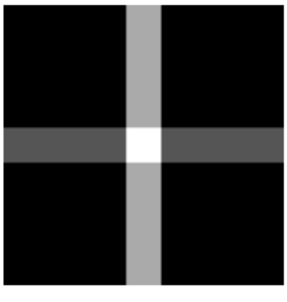
Answers:

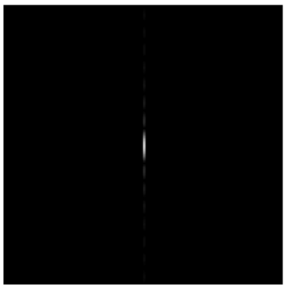
This instruction uses the property stated in Question 5, if p or q exceeds half the image size then the algorithm computes the fourier transform of p-N or q-N. By exploiting the symmetry and periodicity property, we can use N^2 instead of $2N^2$ values to represent the image in Fourier domain.

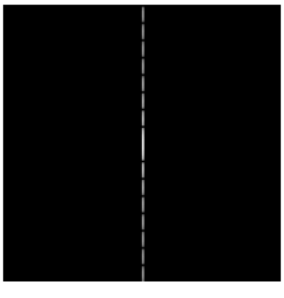
Question 7: Why are these Fourier spectra concentrated to the borders of the images? Can you give a mathematical interpretation? Hint: think of the frequencies in the source image and consider the resulting image as a Fourier transform applied to a 2D function. It might be easier to analyze each dimension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

Answers:

The frequency of those Fourier spectra are null in the x/y axis thus they are concentrated at the borders of the images. We use fftshift on the fourier transformation in order to put the frequency equal to 0 at the center of the image.







H = F+2*G Fourier of F, centered without log Fourier of F centered, with log

---

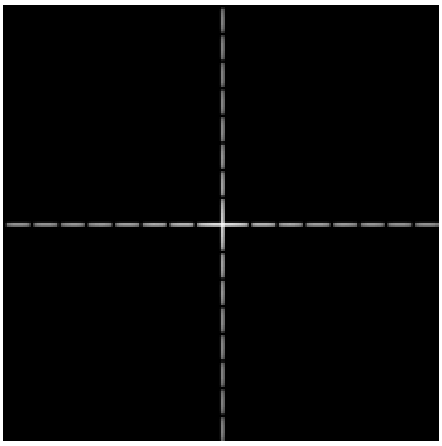
Question 8: Why is the logarithm function applied?

Answers:

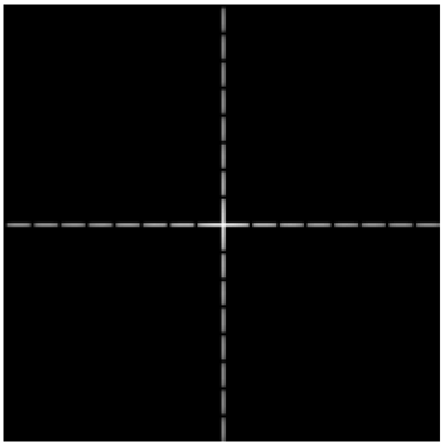
We use the logarithm function to compress the values of the fourier spectrum, if we don't compress those the relative difference between high values and low values is too high to correctly observe the low values, thus they would appear like black dots without the logarithm.

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

Answers:



Fourier of (F+2*G)

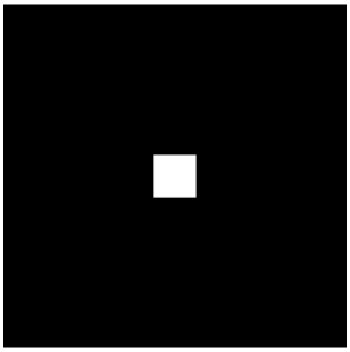


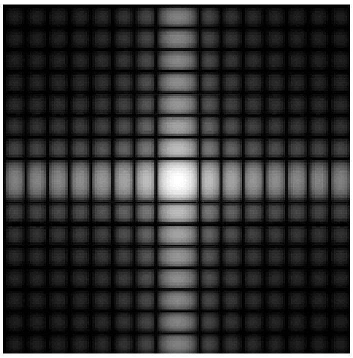
Fourier of F + 2*Fourier of G

We can see that the Fourier transformation is linear because we have fft2(F+2*G)=fft2(F)+2*fft2(G)

F(a*f1+b*f2)=a*F(f1)+b*F(f2)

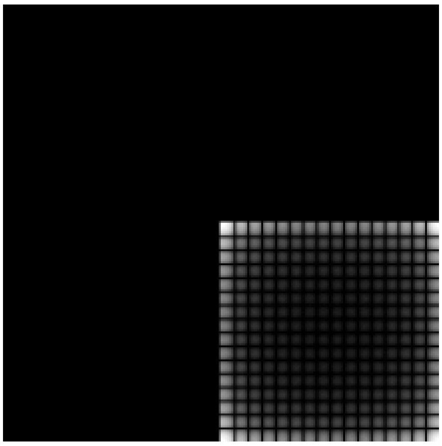
Question 10: Are there any other ways to compute the last image? Remember what multiplication in Fourier domain equals to in the spatial domain! Perform these alternative computations in practice.

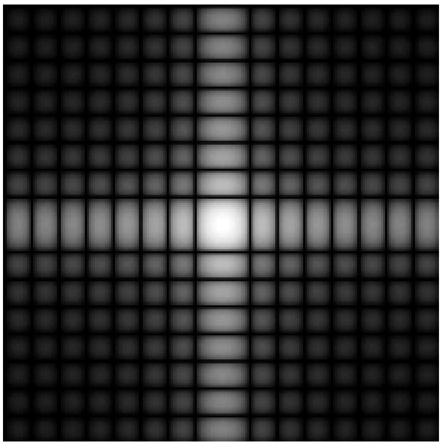




F*G

Fourier of (F*G)





normalized convolution of F(F),F(G)

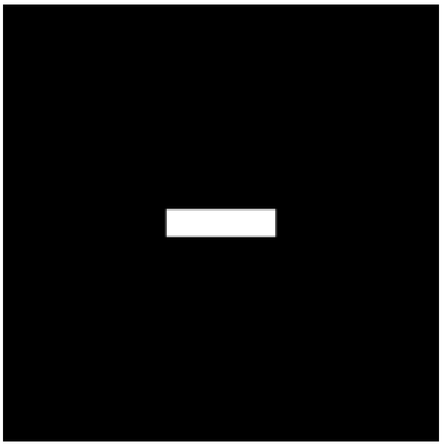
Final convolution of F(F),F(G)

Answers:

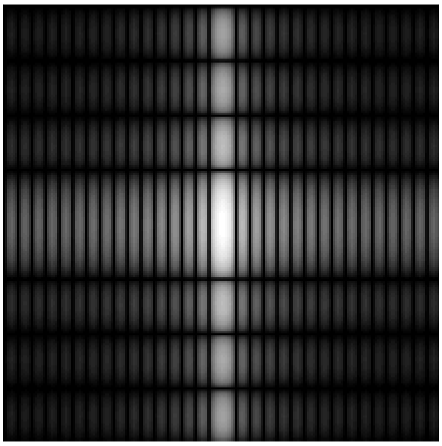
A multiplication in the spatial domain is a convolution in Fourier domain.

Thus we can compute this image thanks to a convolution of the fourier transformation of F and G, then we need to take the right pixel and normalize and we find an image equivalent to the first image.

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.



Picture of F

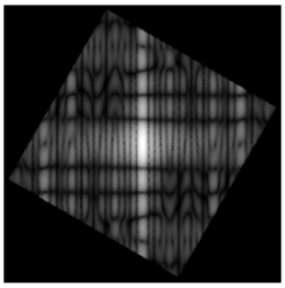


Fourier transform of F

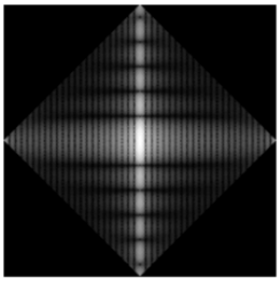
Answers:

By comparing this rectangle with the cube from the last question we can illustrate the scaling property. Indeed we see that when the image is compressed following one axis in the spatial domain then the image in the fourier domain will be expanded by the same value on the same axis.

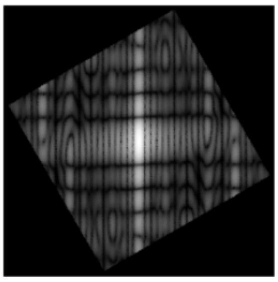
Question 12: What can be said about possible similarities and differences? Hint: think of the frequencies and how they are affected by the rotation.



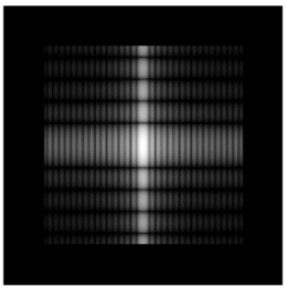
alpha=30°



alpha=45°



alpha=60°



alpha=90°

Answers:

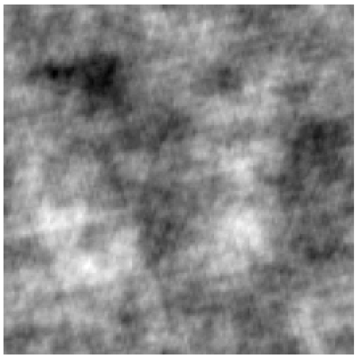
We can see that the Fourier transformation of the rotated image corresponds to the rotated

Fourier transformation of the original image. Thus a rotation in the spatial domain is equivalent to a rotation in the Fourier domain.

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

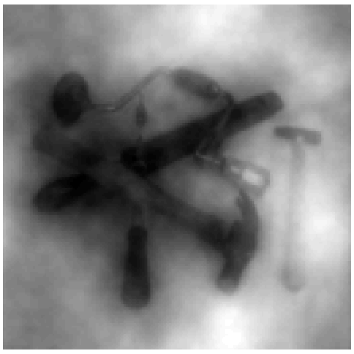
phonecalc128 image

pow2image() applied

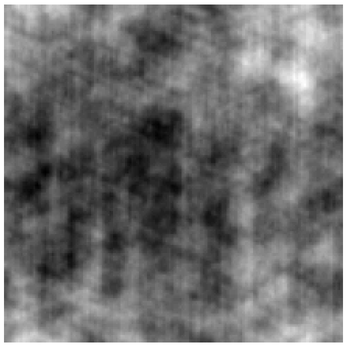


randphaseimage() applied

few128 image

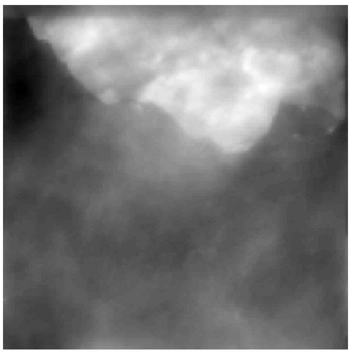


pow2image() applied

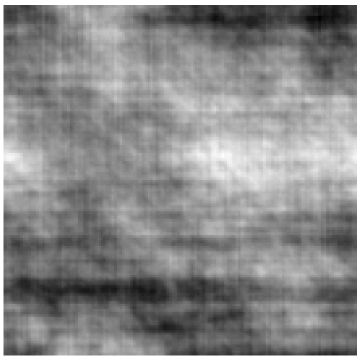


randphaseimage() applied

nallo128 image



pow2image() applied



randphaseimage() applied

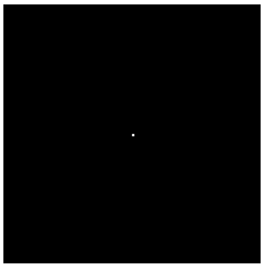
Answers:

When we change the power spectrum of the image thanks to the function pow2image; We can still guess what the image represents because we can still see the shape of the different objects in the image.

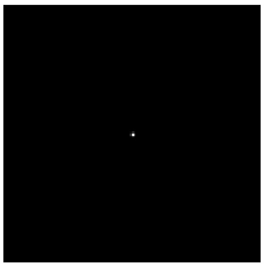
However when we use the function randphaseimage() we can't discern the objects in the image anymore and we can only observe the magnitude of the image, the intensity of the pixel color. However those pixels aren't at the same position anymore so it becomes impossible to guess what is in the picture.
We can conclude that the information contained by the magnitude is the pixels' colour intensity and the information contained by the phase is the position of those pixels.

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?

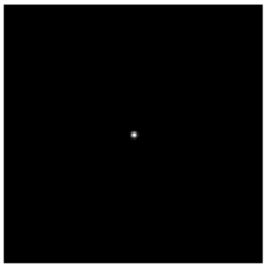
Answers:



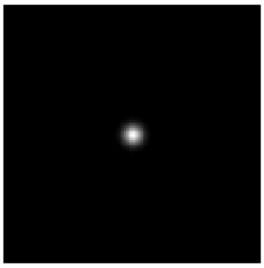
t = 0.1



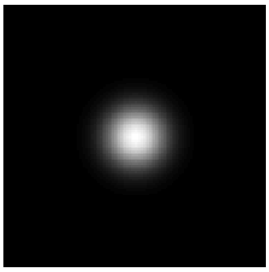
t = 0.3



t = 1



t = 10



t = 100

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:

<table>
<tr><th>t</th><th>Covariance matrix of the Gaussian</th></tr>
<tr><td>0.1</td><td><table><tr><td>0.013</td><td>0</td></tr><tr><td>0</td><td>0.013</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td></tr>
<tr><td>0.3</td><td><table><tr><td>0.28</td><td>0</td></tr><tr><td>0</td><td>0.28</td></tr></table></td></tr>
<tr><td>1</td><td><table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table></td></tr>
<tr><td>10</td><td><table><tr><td>10</td><td>0</td></tr><tr><td>0</td><td>10</td></tr></table></td></tr>
<tr><td>100</td><td><table><tr><td>100</td><td>0</td></tr><tr><td>0</td><td>100</td></tr></table></td></tr>
</table>

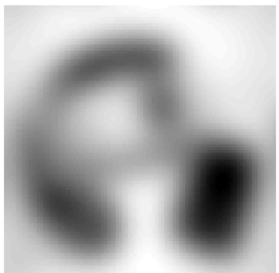
For t with a high value, the covariance matrix is well approximated, unlike when t is very low.

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:

t = 0.1

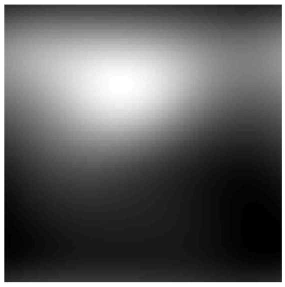
t = 4



t = 64

t = 1

t = 16



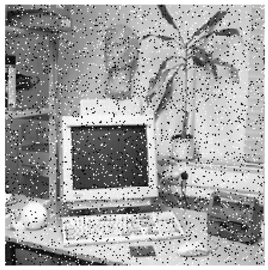
t = 256

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:

Original image

Gauss noise



Sap noise

<u>Gauss noise</u>: adds noise according to a normal distribution centered on 0, with a given standard deviation

<u>Sap noise</u>: set a defined frac/2 of pixels to the maximum value of the image and frac/2 of the pixels to the minimum value of the image.

Gaussian smoothing - Gauss noise

Original image with Gauss noise

Gauss smoothing t = 0.1

Gauss smoothing t = 0.5

Gauss smoothing t = 1

Gauss smoothing t = 5

Gauss smoothing t = 15

Gaussian smoothing - Sap noise

Gauss smoothing t = 0.1

Gauss smoothing t = 1

Gauss smoothing t = 5

Gauss smoothing t = 20

Median smoothing - Gauss noise

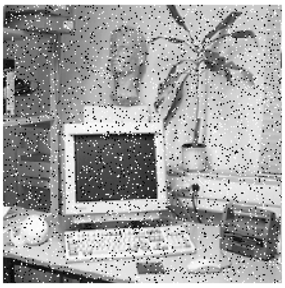
Median smoothing w = 1

Median smoothing w = 4

Median smoothing w = 10

Median smoothing w = 30

Median smoothing - Sap noise



w =1

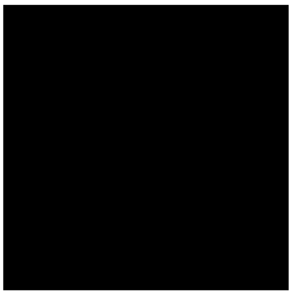
w = 2

w = 3

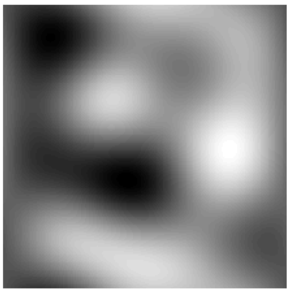
w = 10

w = 20

Low-pass filtering - Gauss noise



Filtering f = 0.001



Filtering f = 0.01

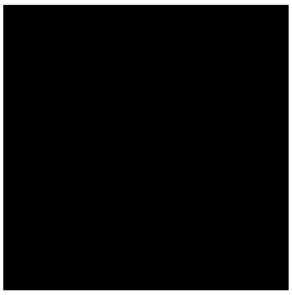
Filtering f = 0.05

Filtering f = 0.1

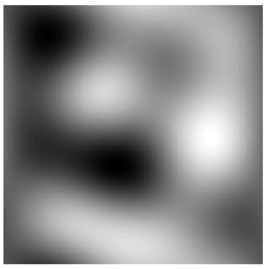
Filtering f = 0.2

Filtering f = 0.5

Low-pass filtering - Sap noise



f = 0.001



f = 0.01

f = 0.1

f = 0.23

f = 0.4

f = 0.6

Gaussian filter		Median filter		Ideal Low-pass filter	
Pros	Cons	Pros	Cons	Pros	Cons
Efficient against Gauss noise	Edges blurred	Efficient against Gauss noise and sap noise	Little leeway to filter (only integers starting from 1)	Remove white noise when there is sap noise	Not efficient against Gauss noise and sap noise
Smoothing		Edges remain			Need to handle very small frequency cutoff
					Edges blurred

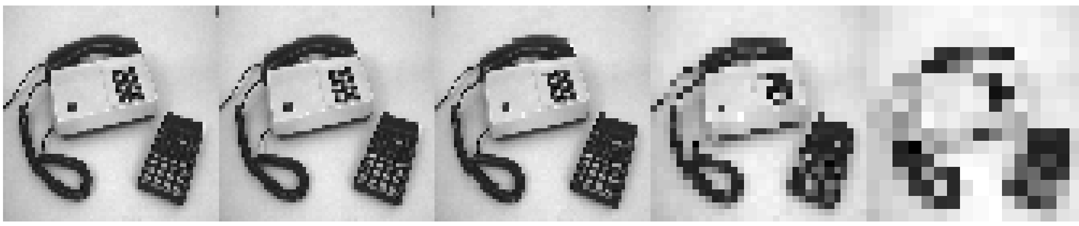
Question 18: What conclusions can you draw from comparing the results of the respective methods?

Answers:

The median filter seems to be the best trade-off for any noise studied. The Gaussian smoothing is however more efficient when the noise is known to be gaussian. The low-pass filter can also be used when there is only white 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:



Original image

Low-pass filter (f = 0.2)

Gaussian filter (t = 1)

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:

- When we compare the original image being subsampled and the others images being smoothen and subsampled, it is possible to observe that the latter remain better: less information is lost.
- Using a low-pass filter (blurring) allows to prevent aliasing: the maximum frequency of the image decreases, so the Nyquist rate also decreases

---