Assignment 1

- *Auteur* -
Hamza Benjelloun

12 novembre 2020

Table des matières

Chapitre 1

Properties of the discrete Fourier transform

1.1 Basis functions

1.1.1 Question 1 :

We observe sine waves with different lengths and different directions (depending on p and q).

1.1.2 Question 2 :

We have :

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

Then it is clear that a (p,q) value in Fourier domain will be transformed to a sine wave in spatial domain.

Here is an illustrative example from matlab :

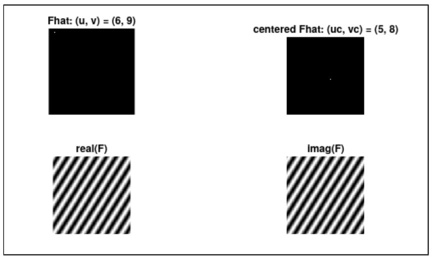

FIGURE 1.1 – **Real and imaginary part and phase of inverse discrete Fourier transform with p, q = 6, 9.**

1.1.3 Question 3 :

From the question 2 we have :

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

Then :

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

1.1.4 Question 4 :

We have from question 2 :

$$F(x) = \frac{\hat{F}(p,q)}{N} [\cos(\frac{2\pi(p,q)^T x}{N}) + i\sin(\frac{2\pi(p,q)^T x}{N})]$$

Since real and imaginary parts are similar (real part is shifted by $\frac{\pi}{2}$, we will study just the real part. And we write $x = (x_1, x_2)$

We have

$$|\cos(\frac{2\pi(p,q)^T x}{N})| = 1 \iff \frac{\pi(p,q)^T(x_1,x_2)}{N} = k\pi \iff px_1 + qx_2 = kN$$

Lines $(D_k) : px_1 + qx_2 = kN$ are parallels and represent the peak of sine wave, and they are directed by the orthogonal line $(\Delta) : x_2 = \frac{q}{p}x_1$ Therefore the direction of this sine wave depend on both p and q. And we have the length wave is the distance between two successive peaks. For example between D_0 and D_1 that is

$$\lambda = \frac{|N-0|}{\sqrt{p^2+q^2}} = \frac{N}{\sqrt{p^2+q^2}}$$

1.1.5 Question 5 :

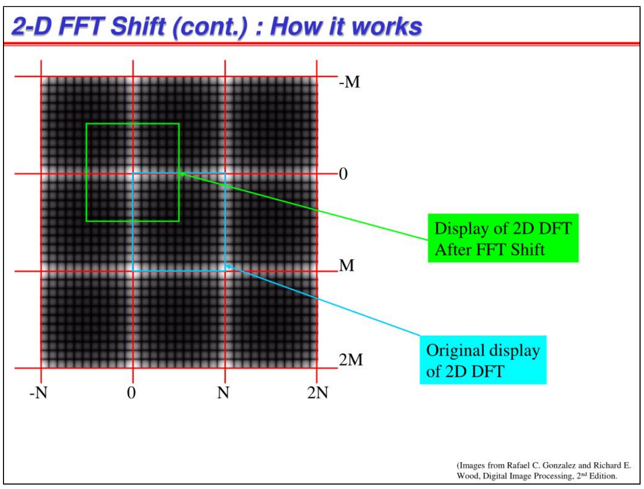

FIGURE 1.2 – **How fftshift work in matlab**

Fftshift work as stated in the image below. We move from the blue image to the green one .If we have u or v greater than $\frac{N}{2}$ we just replace it by their value $-N$ because we have $\cos(X) = \cos(X - 2\pi)$ and $\sin(X) = \sin(X - 2\pi)$.

1.1.6 Question 6 :

In this part of code we translate frequencies $w_1 = \frac{2\pi u}{N}$ and $w_2 = \frac{2\pi v}{N}$ to be in $[-\pi, \pi[$ and not $[-2\pi, 2\pi[$. If both u and v are lesser than $\frac{N}{2}$ then we remove just 1 because in our code we store frequencies in a matrix, and matrix in matlab start with indice 1, 1 and not 0, 0. Otherwise if one of u or v are greater than $\frac{N}{2}$ we should translate them by $-1-N$ to be in $[-\pi, \pi[$ (the result remain the same because of $\cos(X) = \cos(X - 2\pi)$ and $\sin(X) = \sin(X - 2\pi)$)

1.2 Linearity

1.2.1 Question 7 :

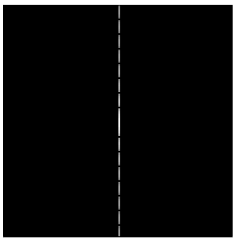

FIGURE 1.3 – **F Fourier spectra**

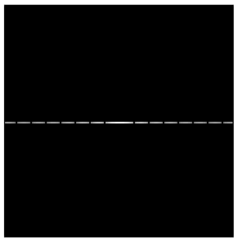

FIGURE 1.4 – **G Fourier spectra**

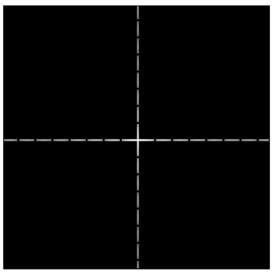

FIGURE 1.5 – **H Fourier spectra**

We have

$$\hat{F}(u,v) = \frac{1}{N} \sum_{(x_1,x_2)\in[0,\ldots,N-1]^2} F(x_1,x_2) e^{-\frac{2i\pi(u,v)^T(x_1,x_2)}{N}} = \frac{1}{N} \sum_{x_1=56}^{71} \sum_{x_2=0}^{N-1} e^{-\frac{2i\pi(x_1u+x_2v))}{N}}$$

$$= \frac{1}{N} \sum_{x_1=56}^{71} e^{-\frac{2i\pi(x_1u))}{N}} \sum_{x_2=0}^{N-1} e^{-\frac{2i\pi(x_2v))}{N}} = \frac{1}{N} \begin{cases} \sum_{x_1=56}^{71} e^{-\frac{2i\pi(x_1u))}{N}} \sum_{x_2=0}^{N-1} 1, & \text{if } v = 0 \\ \sum_{x_1=56}^{71} e^{-\frac{2i\pi(x_1u))}{N}} \sum_{x_2=0}^{N-1} e^{-\frac{2i\pi(x_2v))}{N}}, & \text{else} \end{cases}$$

$$= \begin{cases} \sum_{x_1=56}^{71} e^{-\frac{2i\pi(x_1u))}{N}}, & \text{if } v = 0 \\ \frac{1}{N} \sum_{x_1=56}^{71} e^{-\frac{2i\pi(x_1u))}{N}} \sum_{x_2=0}^{N-1} e^{-\frac{2i\pi(x_2v))}{N}}, & \text{else} \end{cases}$$

$$= \begin{cases} \sum_{x_1=56}^{71} e^{-\frac{2i\pi(x_1u))}{N}}, & \text{if } v = 0 \\ 0, & \text{else} \end{cases}$$

because we have if $v \neq 0$

$$\sum_{i=0}^{N-1} e^{-\frac{2i\pi(x_2v))}{N}} = \frac{1 - e^{2i\pi x_2}}{1 - e^{\frac{2i\pi x_2}{N}}} = 0$$

Therefore Fourier spectra of F function is concentrated in the border. And we have G is the transpose of F Then We will have the same result as previously, we should just switch x_1 and x_2. Then Fourier spectrum of G will be concentrated in the other border. And by separating the sums on F and G we find that H will be concentrated in both borders.

1.2.2 question 8 :

We use logarithm function to amplify little values. We add 1 because log is not defined in 0.

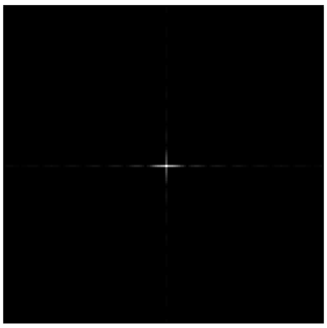

FIGURE 1.6 – **F Fourier spectra without log**

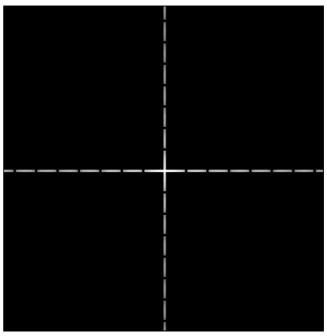

FIGURE 1.7 – **F Fourier spectra with log**

1.2.3 question 9 :

In general :

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

and

$$\mathcal{F}(F^T) = \mathcal{F}(F)^T$$

1.3 Multiplication

1.3.1 question 10 :

We know that

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

Therefore we can find the same result by computing the convolution of F and G Fourier transform :

$$\mathcal{F}(H) = \mathcal{F}(F. * G) = \mathcal{F}(F) * \mathcal{F}(G)$$

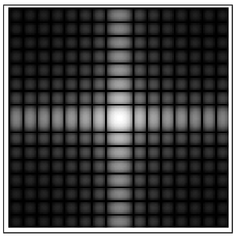

FIGURE 1.8 – **Fourier transform of F.*G with convolution.**

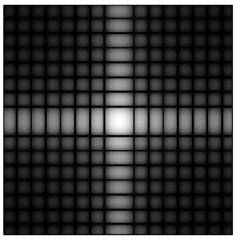

FIGURE 1.9 – **Fourier transform of F.*G without convolution**

1.3.2 Question 11 :

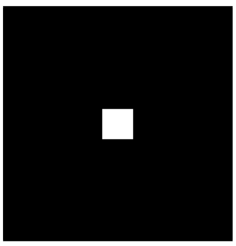

FIGURE 1.10 – **previous F.*G**

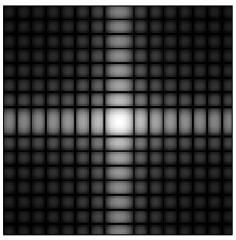

FIGURE 1.11 – **previous F.*G spectrum**

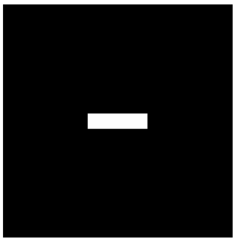

FIGURE 1.12 – **current F.*G**

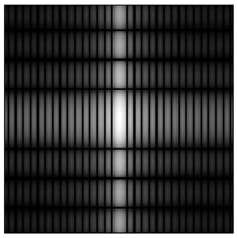

FIGURE 1.13 – **current F.*G spectrum**

We notice that when we scale both x and y component of $F. * G$ and we make them smaller the spectrum becomes bigger and wider. Therefore a scaling in spatial domain results in a scaling in Fourier domain. However compression in spatial domain is same as expansion in Fourier domain and vise versa (as stated in lecture).

1.4 Rotation

1.4.1 Question 12 :

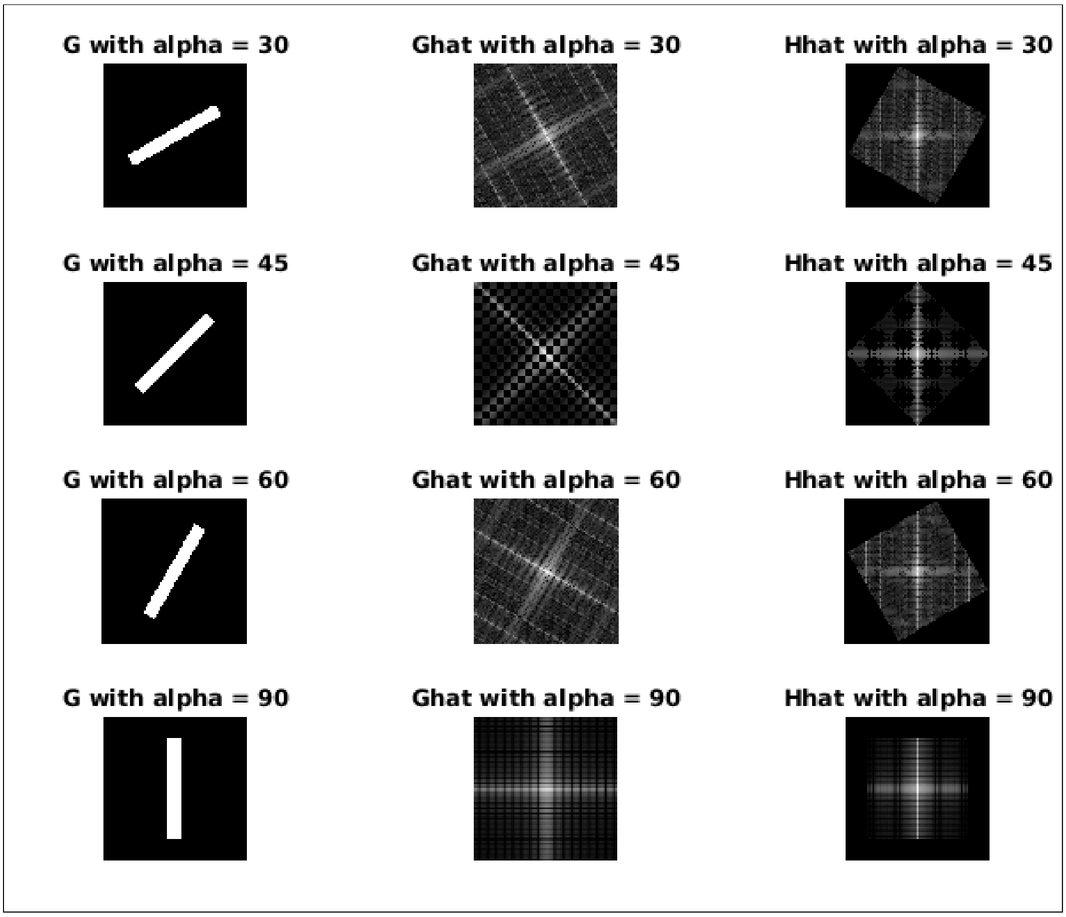

FIGURE 1.14 –

We notice that a rotation in spatial domain result in a rotation in Fourier domain. Let demonstrate that. We note F_{rot} the function F after rotation with angle α and $R_\alpha = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$ the matrix of rotation.

We have :

$$\hat{F}_{rot}(u) = \frac{1}{N} \sum_{x\in[0,\ldots,N-1]^2} F_{rot}(Rx)e^{-\frac{2i\pi u^T Rx}{N}} = \frac{1}{N} \sum_{x\in[0,\ldots,N-1]^2} F(x)e^{-\frac{2i\pi (R^T u)^T x}{N}} = \hat{F}(R^T u)$$

Therefore :

$$\hat{F}_{rot}(Ru) = \hat{F}(R^T Ru) = \hat{F}(U)$$

Then frequencies of F_{rot} are rotation of frequencies of F.

We notice that there are some deformation in both spacial and Fourier domain come from the imperfect rotation.

1.5 Information in Fourier phase and magnitude

1.5.1 Question 13 :

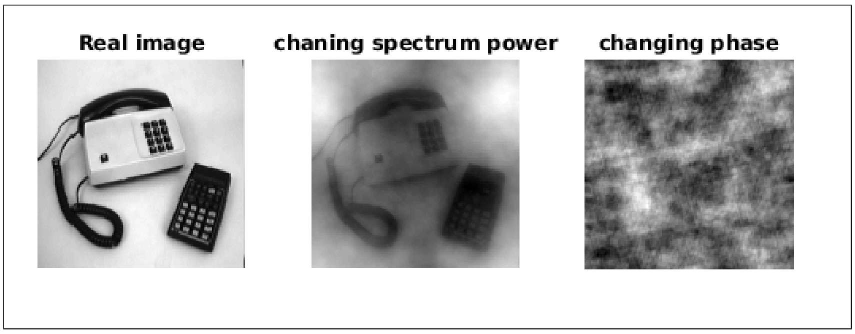

FIGURE 1.15 – **Phone**

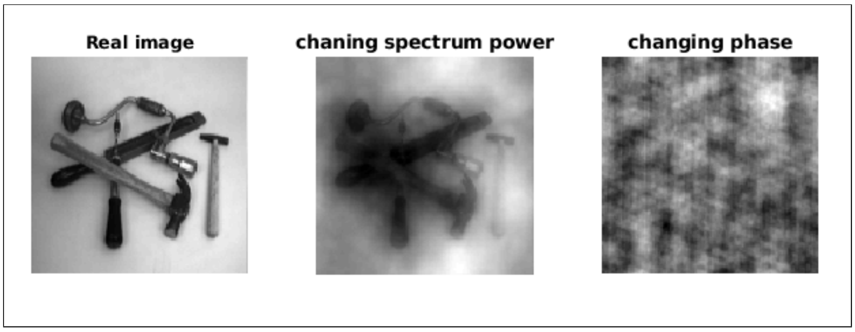

FIGURE 1.16 – **Few**

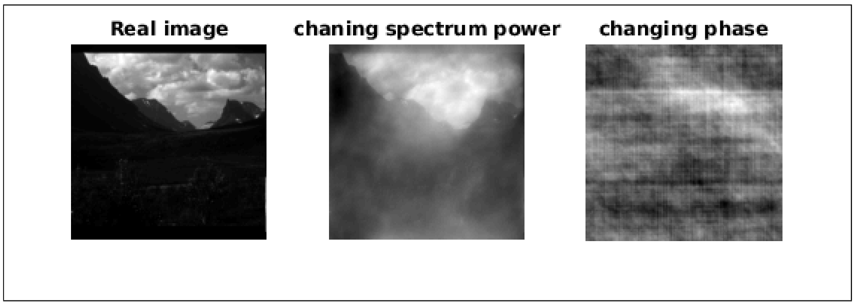

FIGURE 1.17 – **Nallo**

So, we conclude that : the phase defines where edges will end up in the image and the spectrum power what grey-levels are on either side of edge.

Chapitre 2

Gaussian convolution implemented via FFT

2.1 Question 14 :

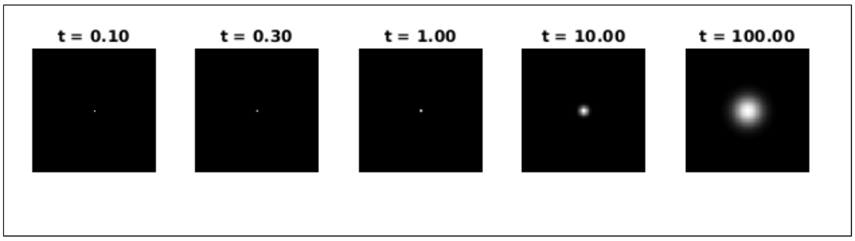

FIGURE 2.1 –

t	covariance matrix
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	$\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$

2.2 question 15 :

We notice in question 14 that for $t \geq 1.0$ we have the same result as the ideal continuous case. However for $t < 1.0$ we are close to the ideal case but we commit some errors. It may result to the bad approximation of Gaussian function when we have small variance, because the peak is compressed then we may skip it or represent it with few points.

2.3 question 16 :

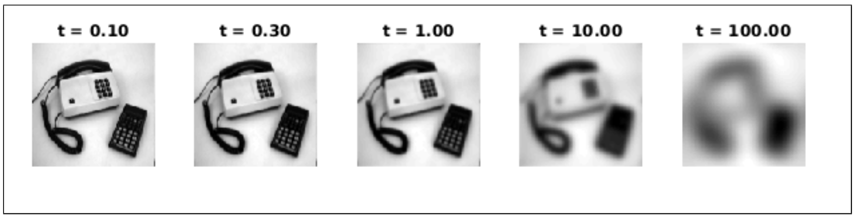

FIGURE 2.2 –

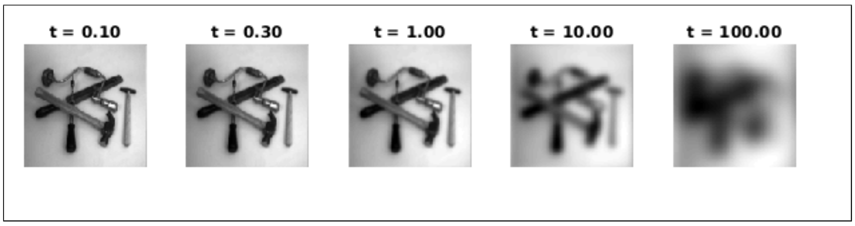

FIGURE 2.3 –

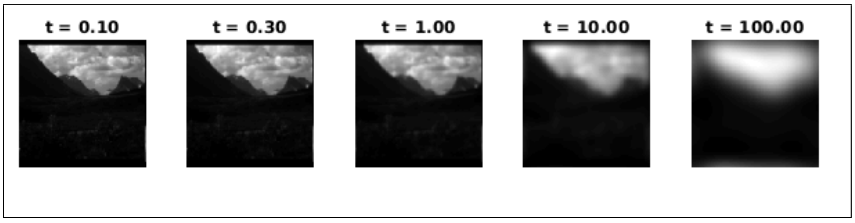

FIGURE 2.4 –

We notice that when we increase t the image becomes blurred. It is reasonable because more t is big more we remove high frequencies.

Chapitre 3

Smoothing

3.1 Question 17 :

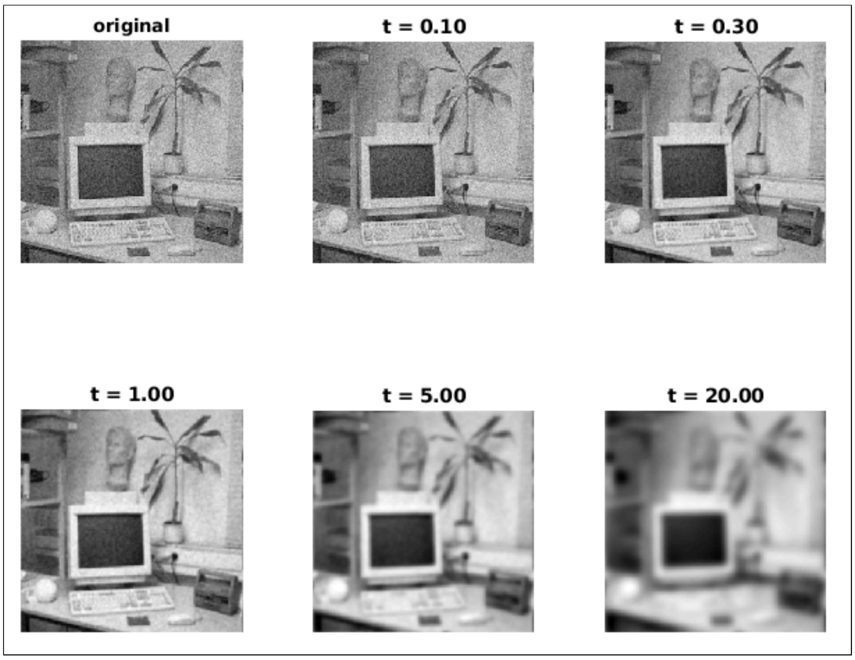

FIGURE 3.1 – **Gaussian filter with Gaussian noise**

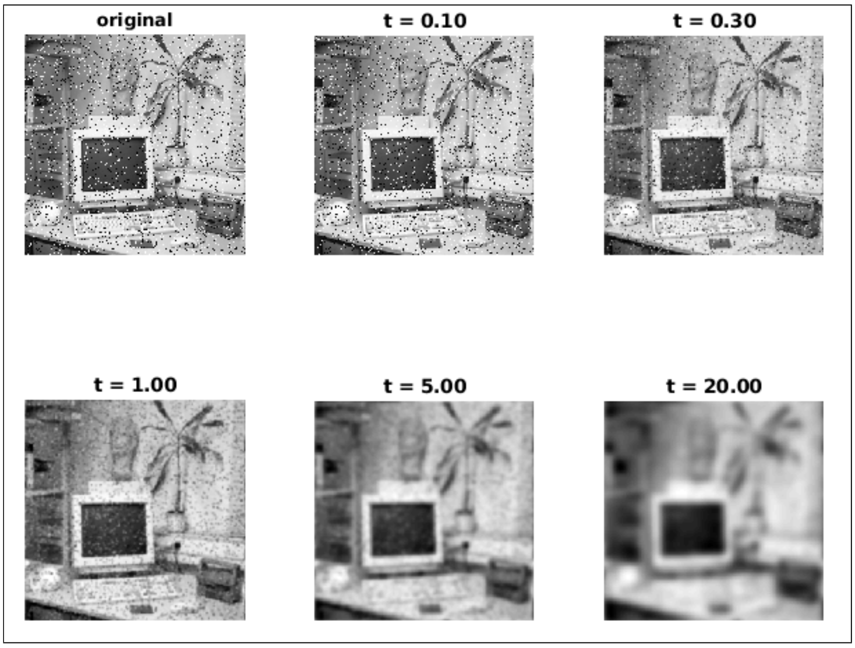

FIGURE 3.2 – **Gaussian filter with sap noise**

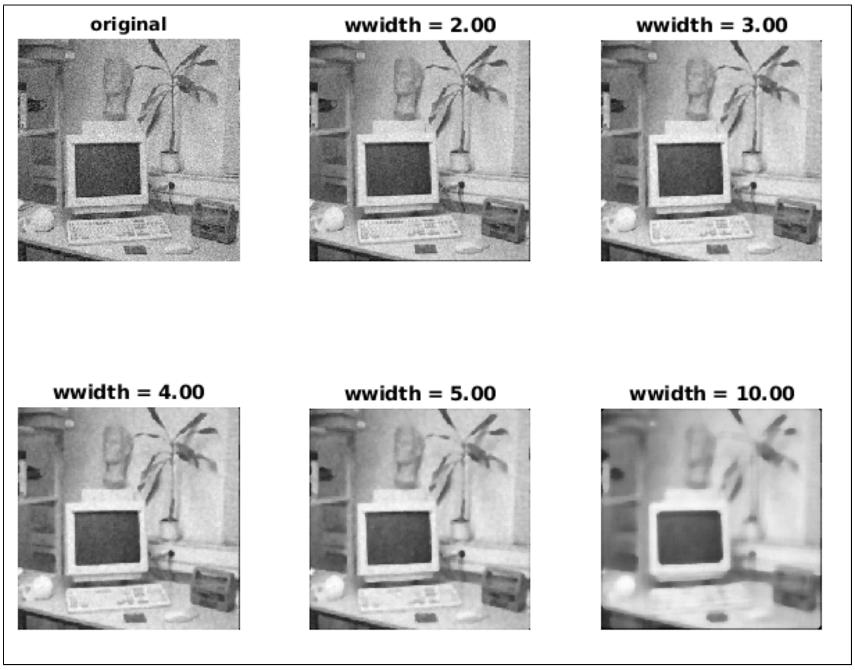

FIGURE 3.3 – **Median filter with Gaussian noise**

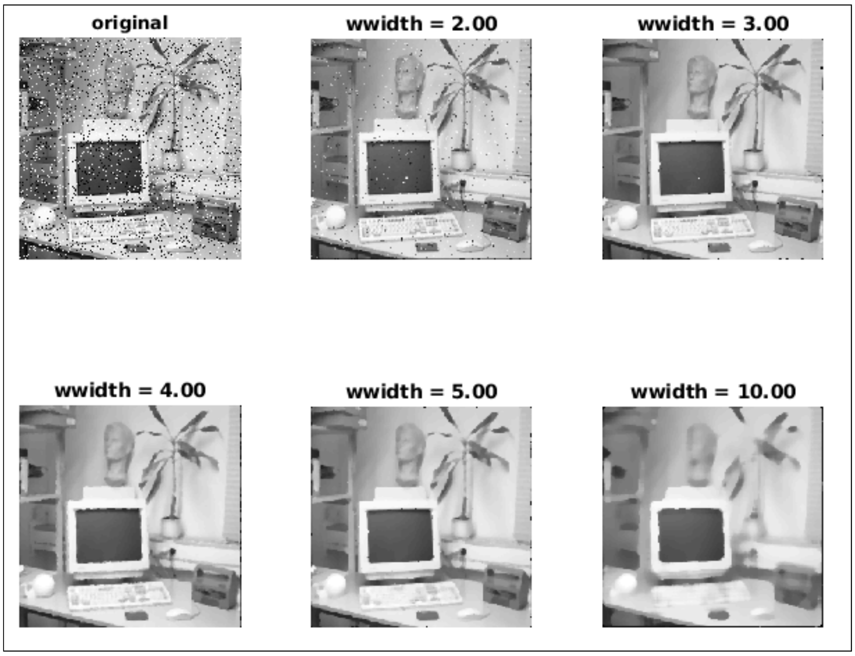

FIGURE 3.4 – **Median filter with sap noise**

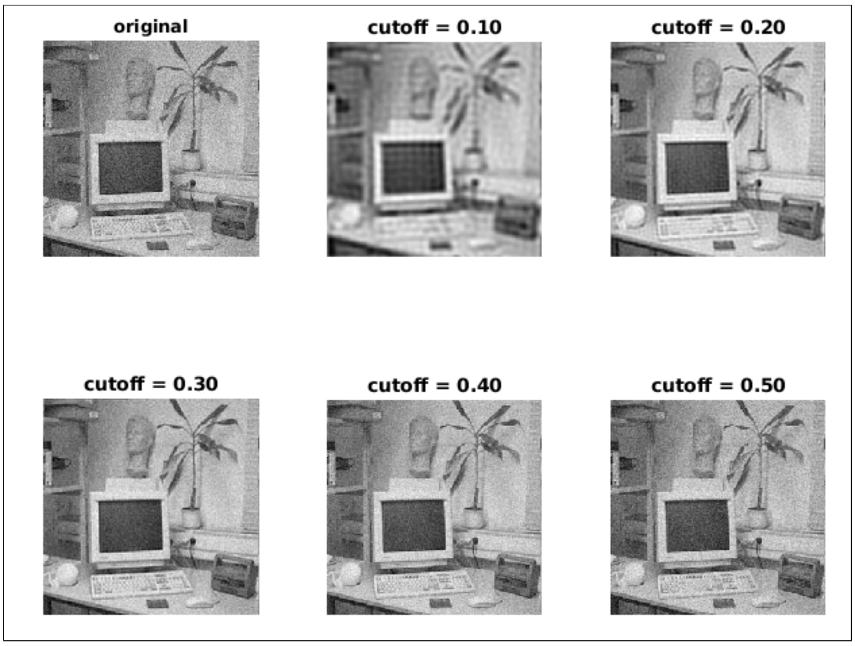

FIGURE 3.5 – **Ideal filter with Gaussian noise**

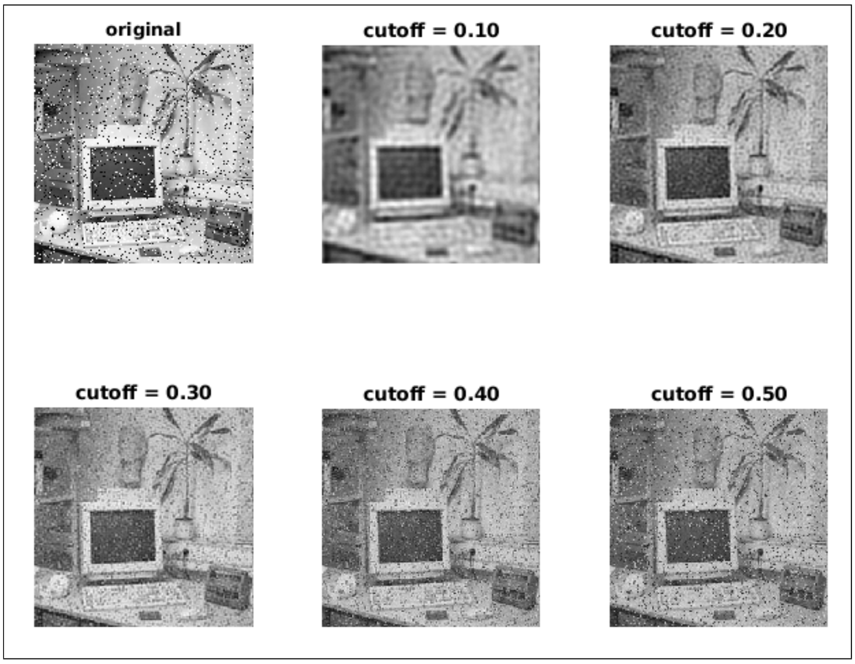

FIGURE 3.6 – **Ideal filter with sap noise**

Advantages and drawbacks of Gaussian filter :
+) Remove Gaussian noises.
+) Smooth uniforms areas
-) Blur image.
-) If the variance is high it removes edges.
-) It doesn't handle pepper and salt noises.

Advantages and drawbacks of median filter :
+) Deal with pepper and salt noises.
+) Preserve edges.
-) Creates painting-like images when the size of the window is high.

Advantages and drawbacks of ideal low-pass filter :
+) smooth image.
-) doesn't work well with pepper and salt noise.
-) Blur image.
-) remove high frequencies.

3.2 Question 18 :

Based on question 17 we conclude that Gaussian filter works well on Gaussian noise and median filter works well on pepper and salt noises.

Chapitre 4

Smoothing and subsampling

4.1 Question 19 :

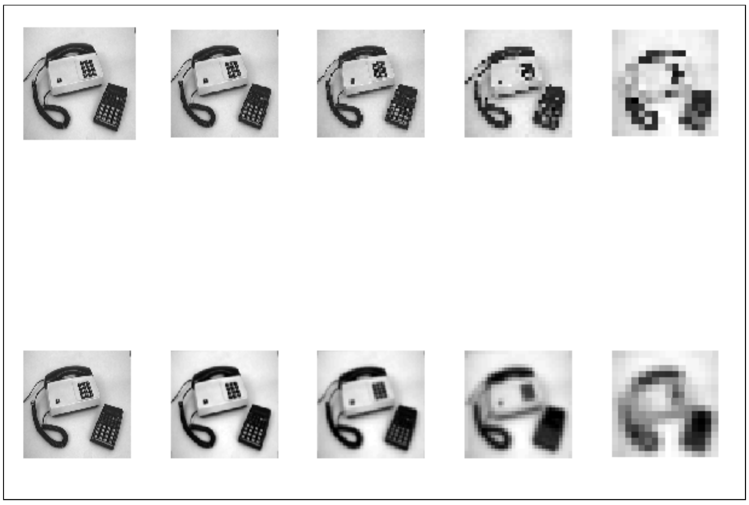

FIGURE 4.1 – **subsimpling and Gaussian filter.**

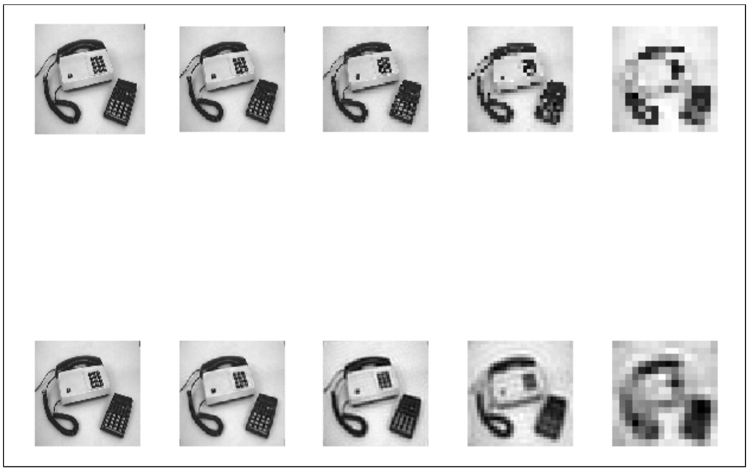

FIGURE 4.2 – **subsimpling and low pass filter.**

We notice that filtering before subsimpling yields much better results (We can recognize the phone in the iteration i = 4).

4.2 Question 20 :

Aliasing occurs when a signal is sampled at a less than twice the highest frequency present in the signal. Signals at frequencies above half the sampling rate must be filtered out to avoid the creation of signals at frequencies not present in the original one. So using Gaussian and low pass filters can improve the result of subsimpling.