School of Computer Science and Communication

DD2423 Image Analysis and Computer Vision:

Lab 1: Filtering operations

1. Basis Functions

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?**

There are 3 groups of values of p,q that have a similar representation of the real and imaginary parts and the angle, just differing on the orientation of the lines. These groups are shown in the following figures:

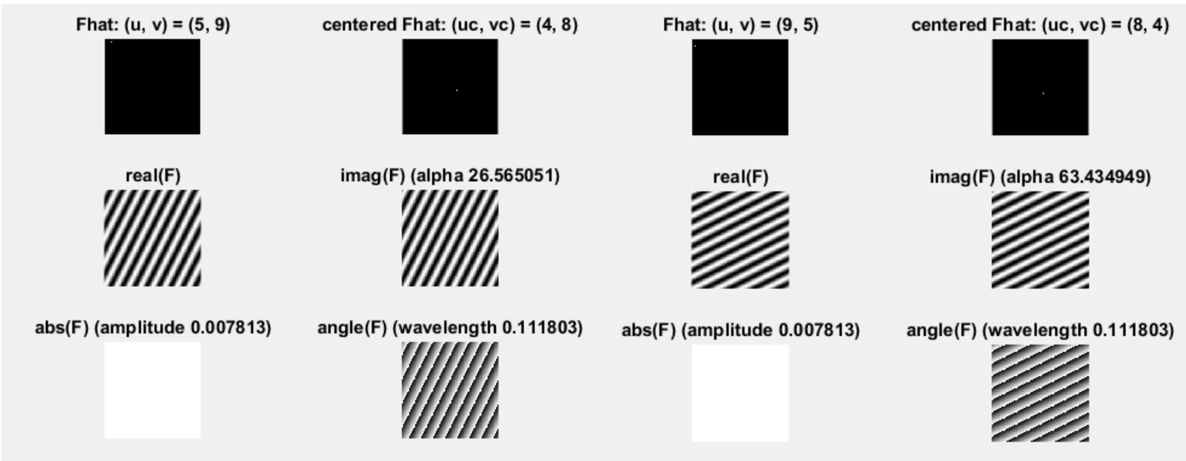

Figure 1: Real, Imaginary, Absolut and Angle representation for `(p,q)` = (5,9) and (9,5)

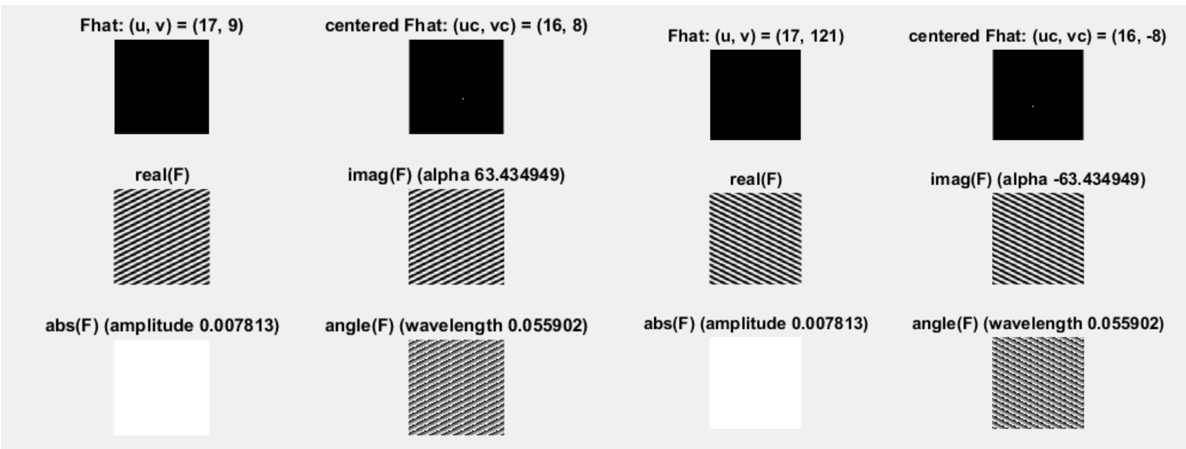

Figure 2: Real, Imaginary, Absolut and Angle representation for (p,q) = (17,9) and (17,121)

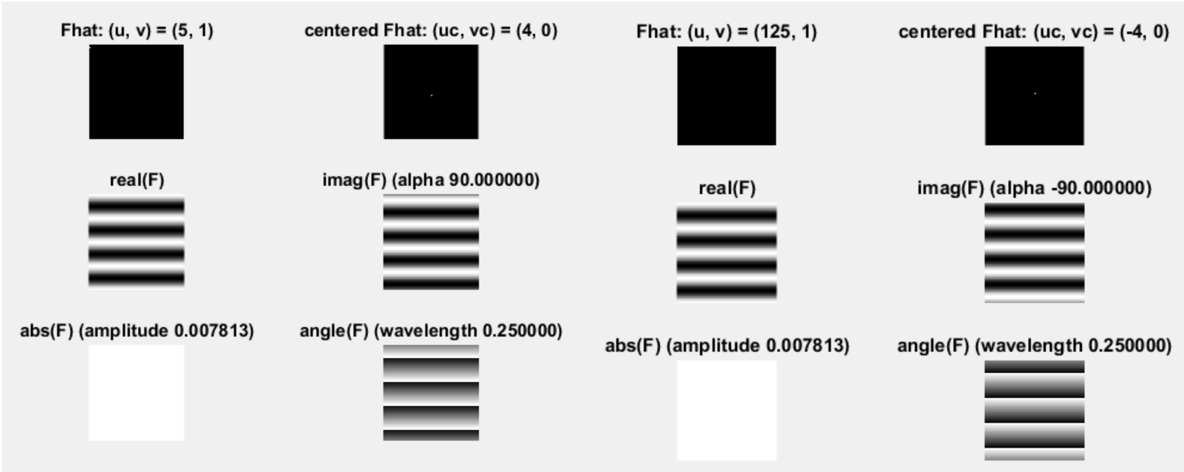

Figure 3: Real, Imaginary, Absolut and Angle representation for (p,q) = (5,1) and (125,1)

Observing the 3 figures we can conclude, first, that the bigger the value of p, the smaller is the period in the y axis, and the same with q and the x axis. This is because, p, and q represents the frequency in this axis, as it was indicated in the lecture notes. In the figure 2 and 3 this frequencies don't change but the images are different. This is because, after we transform `(p,q)` coordinates to the new centered coordinates `(uc,vc)` the absolute value of `(uc,vc)` remains the same in both images, but the sign changes. This is why the orientation of the lines change in a symmetrical way, as the Fourier transform can be represented as a concentric wave with increasing frequency. Also, due to this increasing frequency, the further the chosen point is from the center of the image, the higher is the frequency. This is shown by the value of the wavelength that decreases when the distance to the origin is greater.

On the other side, we can see that the amplitude is constant, as it does not depend on p or q. The amplitude is the maximum value of the absolute value of the frequency. As only one pixel is changed this value is always (being N = size of the image):

$$max\left|\frac{1}{N}f(p,q)\right| = \frac{1}{N}$$

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.**

As it is explained in the lectures slides the expression of the inverse Fourier transformation is:

$$\mathrm{f}(m,n) = \frac{1}{\sqrt{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\hat{\mathrm{f}}(u,v)e^{+2\pi i\left(\frac{mu}{M}+\frac{nv}{N}\right)}$$

And we also now the Euler´s identity:

$$e^{iwx} = cos(wx) + isin(wx)$$

Then we have that

$$e^{i2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)} = cos\left(2\pi\left(\frac{mu}{M}+\frac{nu}{N}\right)\right) + i * sin\left(2\pi\left(\frac{mu}{M}+\frac{nu}{N}\right)\right)\ with\ freq = 2\pi\left(\frac{mu}{M}+\frac{nu}{N}\right)$$

And then the sine wave in spatial domain is:

$$\|f(m,n)\| = \frac{1}{N}\|\hat{f}(u,v)\| * \left(cos\left(2\pi\left(\frac{mu}{M}+\frac{nu}{N}\right)\right) + i * sin\left(2\pi\left(\frac{mu}{M}+\frac{nu}{N}\right)\right)\right) = \frac{1}{\mathrm{N}} * (cos(freq) + i * sin(freq))$$

With N = size of the image in pixels, and the freq defined by:

$$freq = 2\pi\left(\frac{mu}{M}+\frac{nu}{N}\right)$$

Implementing this equation in MATLAB, for the first given point (p,q) = (5,9) we get:

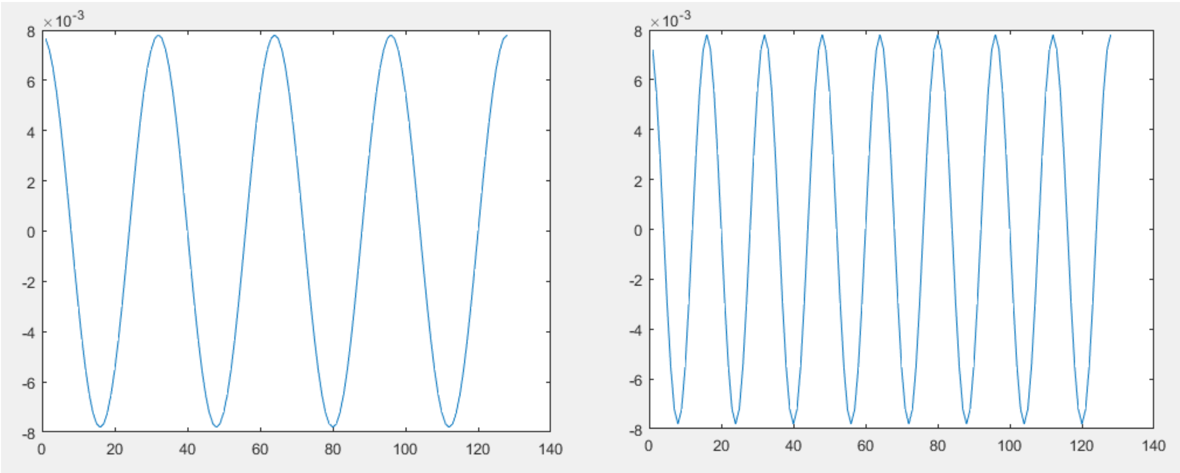

Figure 4: Output for p = 5 and q = 9. On the left, propagation on the x axis, y axis on the right

Question 3: **How large is the amplitude? Write down the expression derived from Equation (4) in these notes. Complement the code (variable amplitude) accordingly.**

The equation (4) in the labs notes is:

$$F(x) = \mathcal{F}_D^{-1}(\hat{F})(x) = \frac{1}{N}\sum_{u\in[0..N-1]^2}\hat{F}(u)\,e^{\frac{+2\pi i u^T x}{N}}. \tag{4}$$

Then, the amplitude of this function is obtained doing `|F(x)|`, which can be done using `abs()` in MATLAB. But we also have to take into account, that MATLAB´s function FFT uses a factor $1/N^2$ in the inverse, this is why we have to multiply the value given by the function `abs()` by N. Having then:

amplitude = Fabsmax*sz;

Question 4: **How does the direction and length of the sine wave depend on p and q? Draw an illustrative figure on paper. Write down the explicit expression that can be found in the lecture notes. Complement the code (variable wavelength) accordingly.**

As it is explained in the lecture notes, the wavelength of the sine wave is:

$$Length = \frac{1}{\|(uc, vc)\|}$$

Where `(uc,vc)` are the transformed coordinates of `(p,q)` so the center of coordinates is centered in the image and adapted to the periodical character of the transformation. This would be explained in the question number 6.

Using the same notation, uc and vc, we can calculate α. I will use figure 5 in order to obtain the mathematical expression of α.

As we can see, and using trigonometrical identities, we have:

$$L = \frac{1}{vc}cos(a) = \frac{1}{uc}sin(a) \rightarrow \frac{uc}{vc} = tg(a) \rightarrow a = tg^{-1}\left(\frac{uc}{vc}\right)$$

We can see in figure 6 the difference in the angles when we choose (5,9) or (9,5) as (p,q).

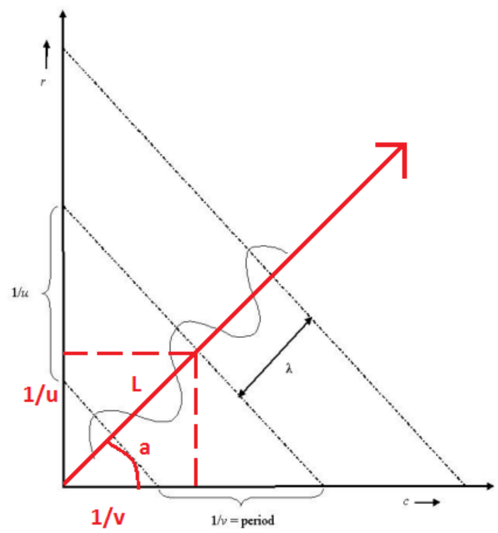

Figure 5: sin wave direction, period and length representation.

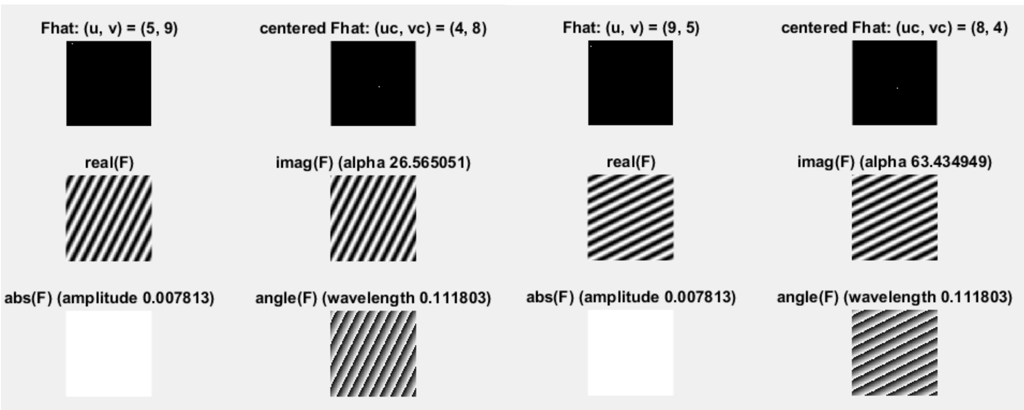

Figure 6: angles (in degrees) for (p,q) = (5,9) and (p,q) = (5,9)

<u>Question 5:</u> **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!**

As it was explained in class, the center of the Fourier transform is the center of the image, and there is symmetry on both the x and y axis due to the periodic character of this transformation. We can show this by choosing a point on the left upper side of the image, and the symmetrical point around y axis on another image. If the transformation is periodic as it was said in class, we should get a symmetrical solution around the y axis. This can be shown by the following figures:

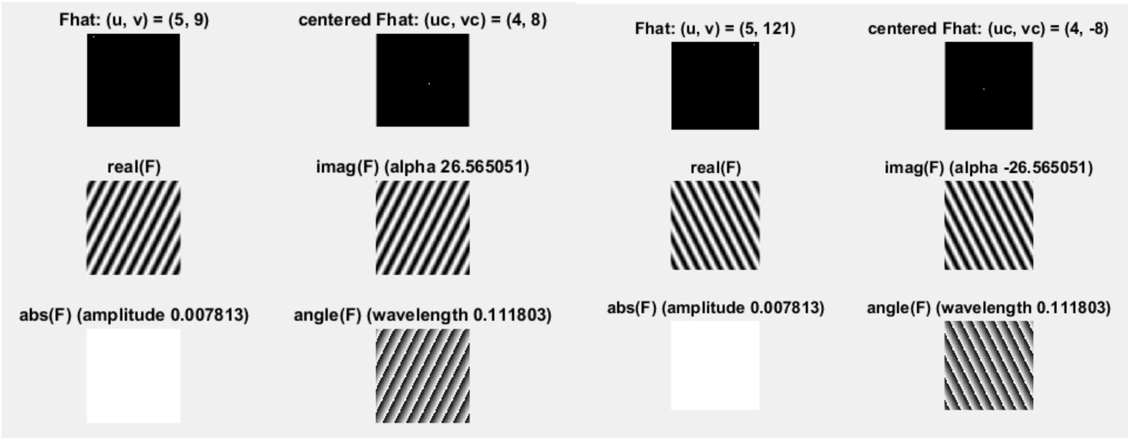

Figure 7: on the left, coordinates (5,9) . On the right, the y—axis simmetrycal point, (5,121)

Repeating the process, but this time, exceeding the value in the y axis, we get that the symmetry is now around the x axis, as expected.

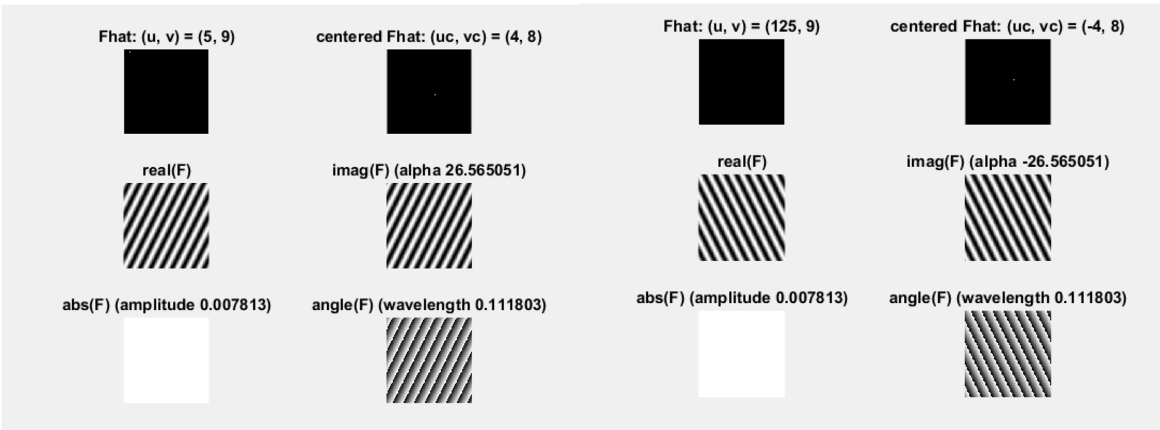

Figure 8: on the left, coordinates (5,9) . On the right, the x—axis simmetrycal point, (125,9)

The last experiment is then to exceed the value both in the x and y axis, and now, the symmetry should be around a point, the center of the image.

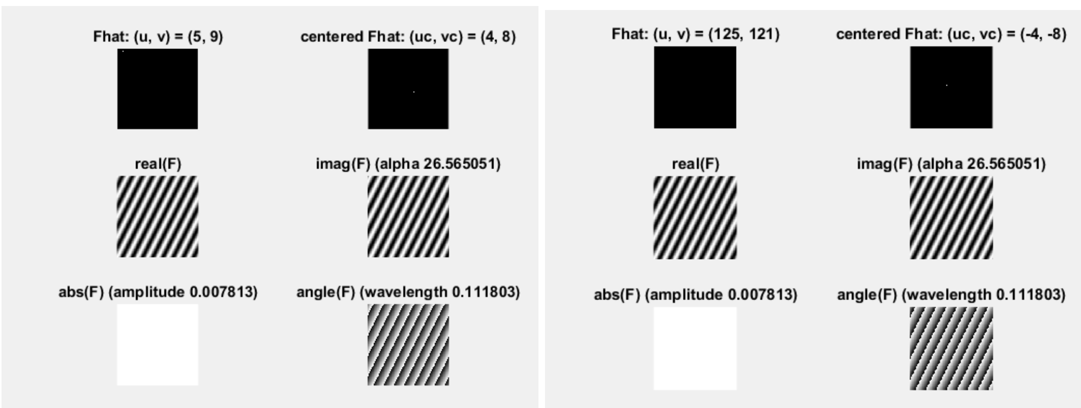

Figure 9: on the left, coordinates (5,9) . On the right, the center simmetrycal point, (125,121)

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

As we concluded before, there is a periodical character on the transformation; this is why is the same to represent the frequencies from 0 to N than to –N/2 to N/2. Using this notation makes it easier to understand what is happening, this is why, in the function `fftwave()`, the system of coordinates is changed so that we can use this new notation. In the original image, the origin was in the point `(1, 1)`, situated in the upper left corner. Now, we want the origin of our new system to be `(0, 0)`, and centered in the image. And also, by this change, now instead of using the 0 to N system of frequencies, using the –N/2 to N/2 one. This is done by the part of the code referred as 'What is done by these instructions?'.

2. Linearity

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!**

After implementing the code that was expressed in the lab guide, we get the following output:

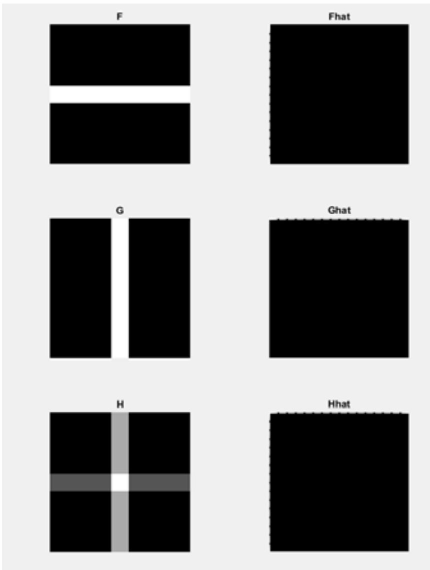

Figure 10: Original image (left) and the Fourier transformation(right)

To show the transformation output, it was need to use `log(1 + abs(Fhat))` instead of Fhat, and the same for Ghat and Hhat. We use the `abs()` function, as we need the magnitude of the transformation, we cannot represent an imaginary number in an image otherwise. The one is added just to help us see the line represented as this way the line represented is not a continuous line and then it can be differentiated better.

The spectra is concentrated in the borders of the images as the coordinates haven´t been shifted so that the origin is centered in the middle of the image, and then our system moves from 0 to N not from –N/2 to N/2. If we reorder the quadrants, using the function `fftshift()`, we have:

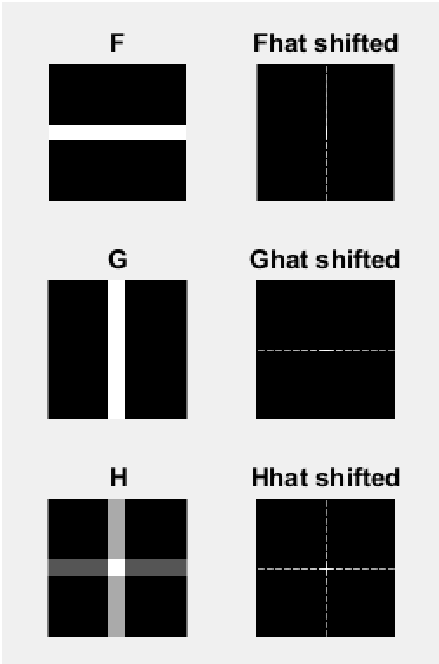

Figure 11: Original image (left) and the shifted Fourier transformation (right)

We observe that in the first 2 images we only have variation in one of the axis. In the first image, the value of the pixels remains constant if we move along the x axis; they only change if we move along the y axis, and this is why, in the Fourier transformation, the output only shows changes in the y axis.
This also happens in the G image, where now, in the original image we only have changes along the x axis, and then, in the transformed image, there are only changes in the x axis.

The Fourier transformation transforms a signal in the summatory of the frequencies that form this signal. Due to this, if there is no variation, the frequency is zero, and then, no frequencies can be represented along the invariant axis.

The last one, the H image, has a variation of the pixels both in the x and y axis, then as there is a variation there is a frequency that defines this variation, and this frequency is shown by the Fourier transformation image Hhat, that shows a variation of the frequency in both axis x and y.

Question 8: Why is the logarithm function applied?

The logarithm function is used so that the output is clearer.

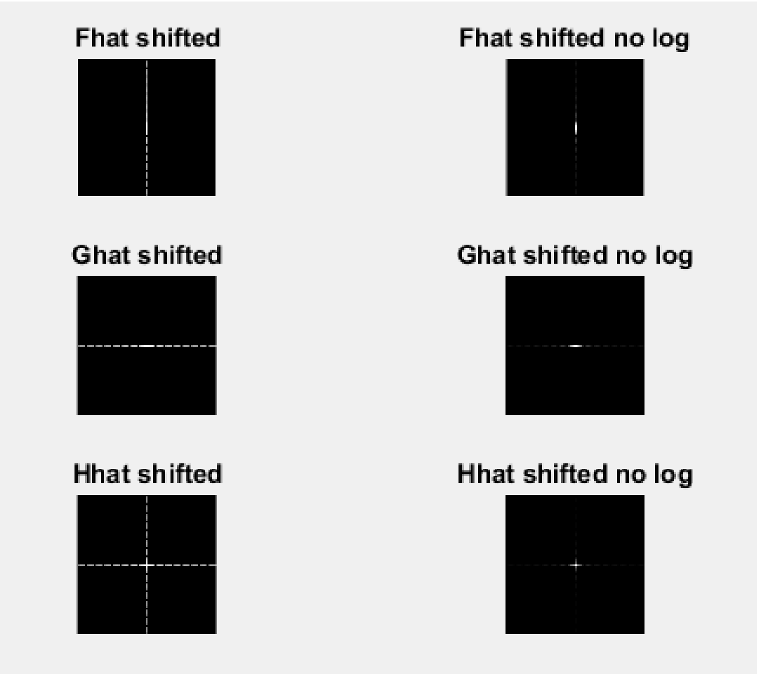

Figure 12: Log function used (left) and no use of the log function (right)

I implemented some code just to output the numerical variation given by the use of the log function and I got the following output.

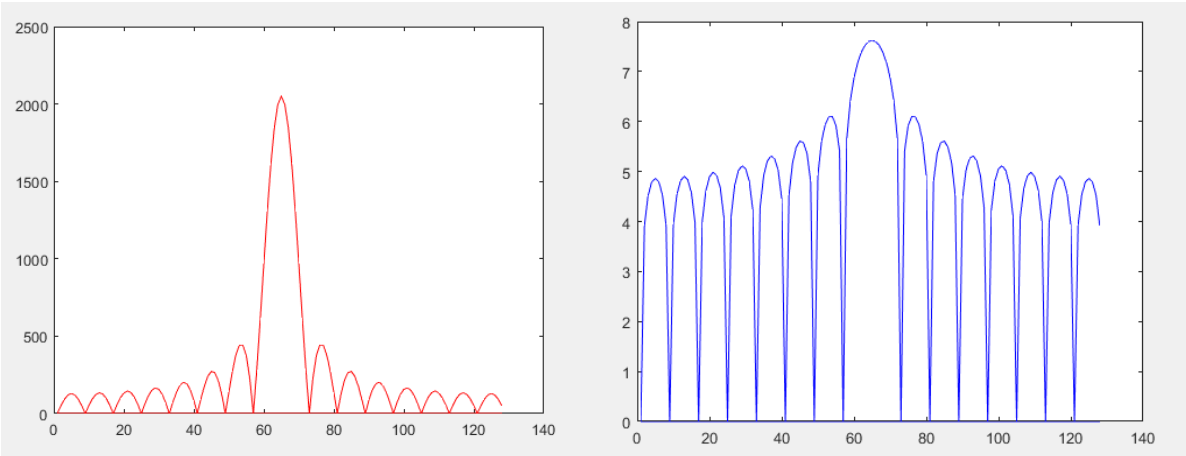

Figure 13: Output of the values after the Fourier transformation using log() (right) or not (left)

Now, we can conclude that using the logarithm function we smooth the dynamics of the transformation so the different between the central value and the rest of the crests is not as big and the visualization is clearer.

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

As it can be clearer observed in the figure 14, if we have an image that is a linear summatory of images, then the Fourier transformation of this image can be expressed as the summatory of the Fourier transformation of the added images. This is mathematically expressed as:

$$H = a * F + b * G \rightarrow F(H) = F(a * F + b * G) = F(a * F) + F(b * G)$$
$$= a * F(F) + b * F(G)$$

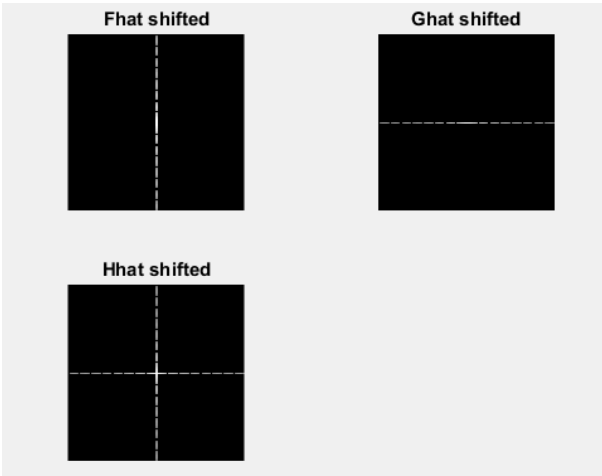

Figure 14: Output of shifted Fourier transformation for F,G and H

Using the already implemented function `showfs()`, that is equivalent to use `showgrey(log(1 +abs(fftshif(Hhat)))`, with both Hhat , and the linear combination Fhat + 2*Ghat, we can get the following output:

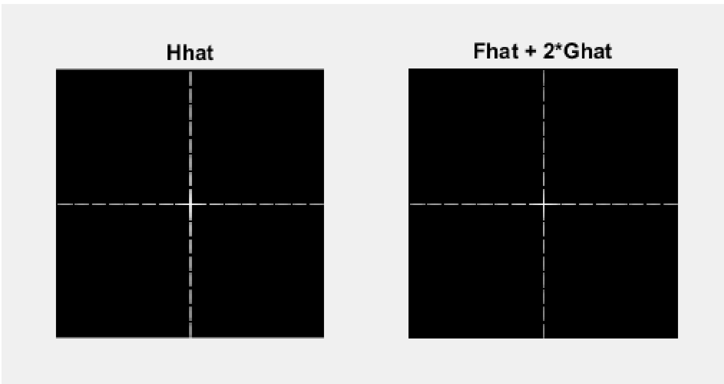

Figure 15: Fourier transformation of H on the left, output of Fhat + 2*Ghat on the right

3. Multiplication

<u>*Question* 10*:*</u> **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.**

As indicated in the lecture notes, we know that the Fourier transformation of the convolution of two images in the spatial domain is the same as the multiplication of the images in the Fourier domain.

$$\mathcal{F}(\mathrm{hf}) = \mathcal{F}(\mathrm{h}) * \mathcal{F}(\mathrm{f})$$

Then if we implement the convolution of the images in the Fourier domain, we can get the next output:

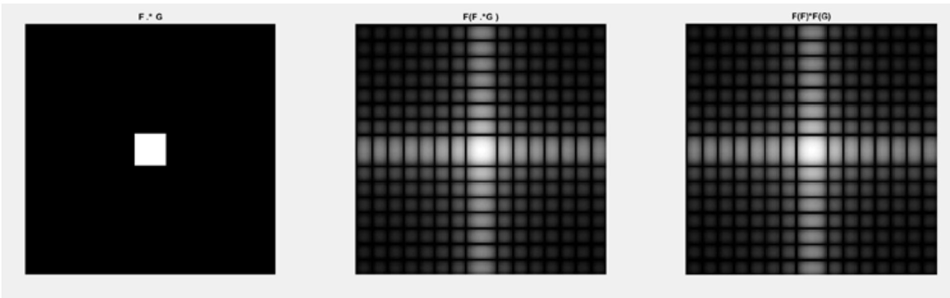

Figure 16: Output of F.*G, F(F.*G) and F(F)*F(G)

We can easily see that the output when doing F(F.*G) and F(F)*F(G) is the same.

4. Scaling

<u>*Question* 11*:*</u> **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.**

If we compare the output of this exercise with the one obtained if the previous one (figure 17), we can conclude that an expansion in one axis of the the spatial domain means a compression in the same axis in the Fourier domain and vice versa.
Using the lectures notes, we have the next mathematical expression:

$$A = \begin{pmatrix} S_1 & & \\ & \ddots & \\ & & S_n \end{pmatrix} \text{ (diagonal)}$$

$$g(x) = f(S_1 x_1, \ldots, S_n x_n)$$

$$\hat{g}(\omega) = \frac{1}{|S_1 \ldots S_n|} \hat{f}\left(\frac{\omega_1}{S_1}, \ldots, \frac{\omega_n}{S_n}\right)$$

This means that is $S_i > 1$, it will expand in the x_i direction in the spatial domain and then it will be compressed in the Fourier domain in the w_i correspondent direction. If $S_i < 0$ then we will have the opposite situation.

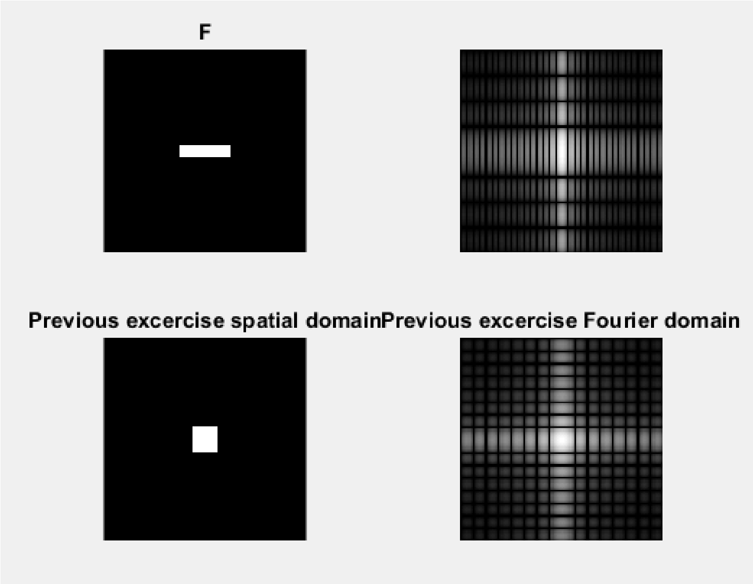

Figure 17: Fourier transformation of two test images

5. Rotation

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

After rotating the image with different angles, we had the output shown in the Figure 18. We can observe that the frequency pattern rotates the same angle as the original image, taking into account the scaling factor explained in the previous question. There is, however, some distortions in the frequency representation when the angle is different from 0,45,90. This is because the image is made of pixels, and therefore is not continue. This is why in the 30 and 60 degrees rotation, an increase in the x direction doesn't mean the same increase in the y direction and this is why it cannot be perfectly transformed to the Fourier domain, as we are missing some information in the transformation.

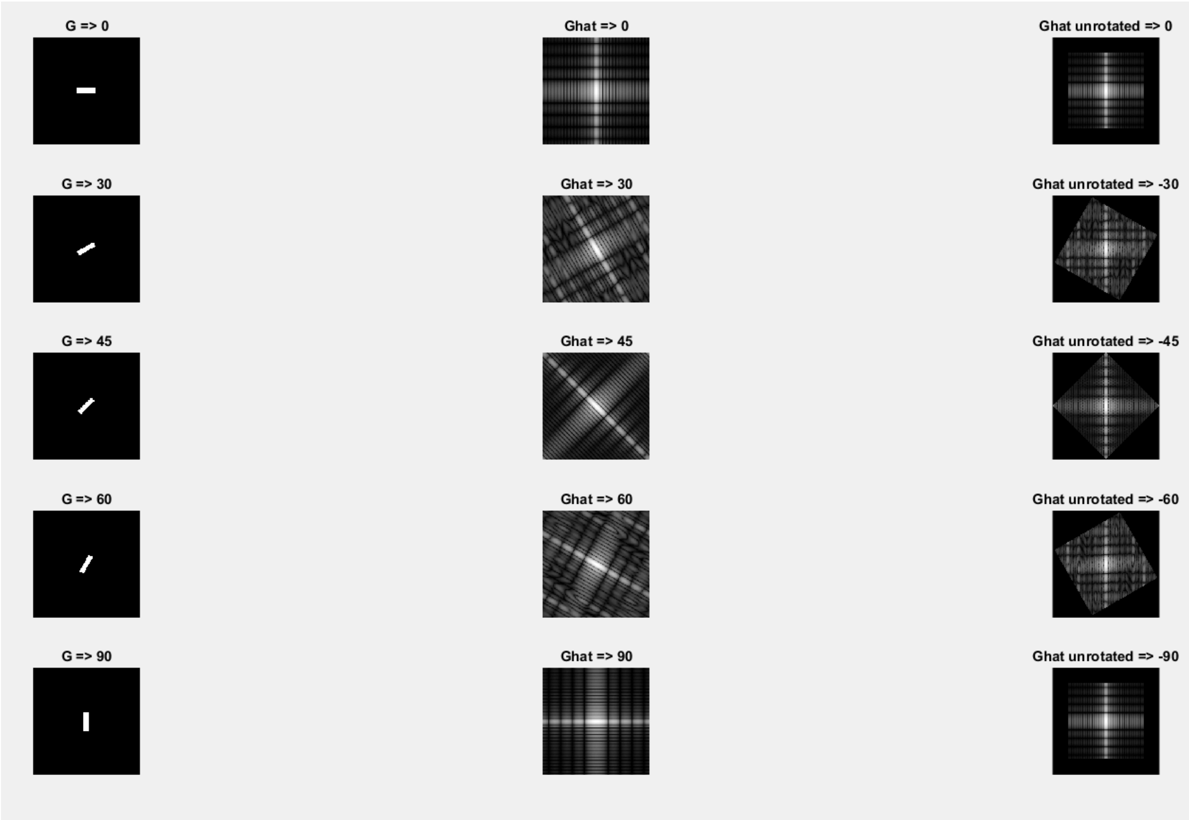

Figure 18: Different rotations and the Fourier transformation of each rotation

6. Information in Fourier phase and magnitude

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

In the figure 19 we can see how an image is transformed when we apply the functions `pow2image()` and `randphaseimage()`. We first have to understand what this functions do. In the case of `randphaseimage()`, is easy, it changes the phase of the image frequency by a random distribution.

In the case of the funtion `pow2image()`, the spectrum of the image frequency is changed, by the following expression:

$$|\hat{f}(\omega)|^2 = \frac{1}{a + |\omega|^2}$$

Now that we understand this functions, we can make some conclussions after observing how this functions change the output images.

When the funtion `pow2image()` is applied, the output image is blurry and the grey distribution changes, but we can still see how the original picture was, eventhough that the shapes are less clear now.

If we apply `randphaseimage()` funtion, the image is completely distorted. This makes sense, as the distribution of the phase is now random, and the angles are not in order now. The frequency distribution is then not ordered neither, then, if we

apply the inverse transformation, the pixels will be messy, and we will not be able to recognise the image as we can observe in the image 19.
We can conlude then that the amplitude contains information about the intensity of the pixels and the phase about the position of these pixels.

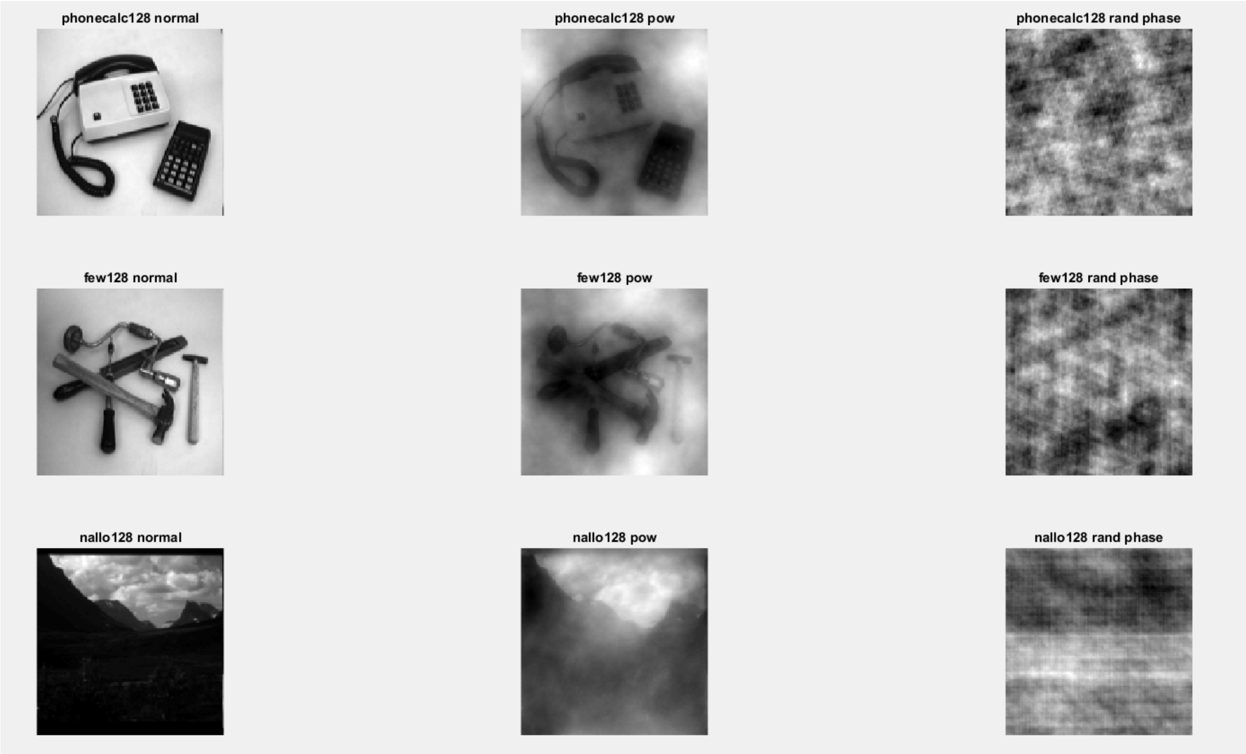

Figure 19: magnitude and phase transformation of three images

7. Filtering procedure

<u>*Question* 14:</u> **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?**

In the figure 21 the values of the variance and the impulse response for our Gaussian Kernel are shown:

<u>*Question* 15:</u> **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.**

As shown in the figure 21, the results don't differ from the estimated variance for the highest values of t. However, for low values of t we have that the lower the variance t, the greater is the difference between the results and the estimation. For lower t variance, the number of pixels inside in the Gaussian distribution is too small and the discretization of the continuous spatial domain is not perfectly obtained. This can be shown in the figure 20. For t = 100, the Gauss distribution is wide, and in the distribution is represented by a big amount of pixels, so the changes in the continuous spatial domain can be represented well enough by the discretization. For t = 0.1, the variation in the distribution is much greater and is represented by a few number of pixels, then the discretization of the continuous spatial domain of the Gaussian distribution is not completed.

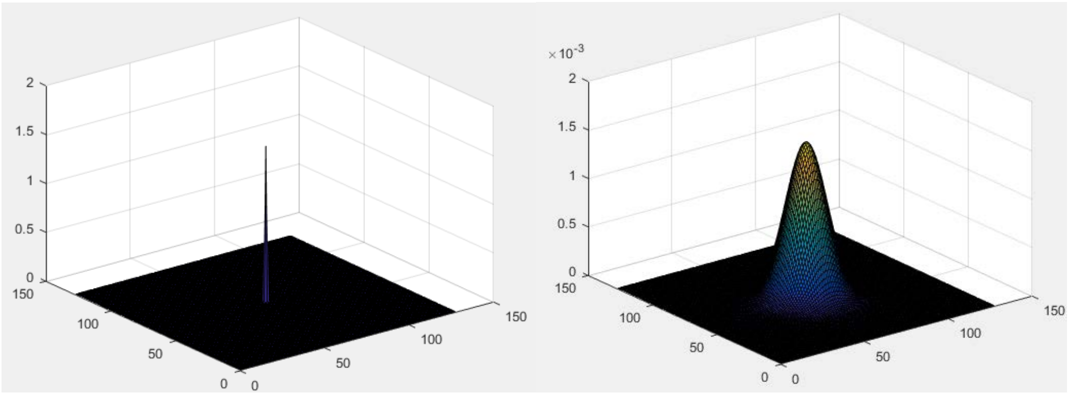

Figure 20: Gaussian spatial continuous distribution for t = 0.1 and t = 100 respectively

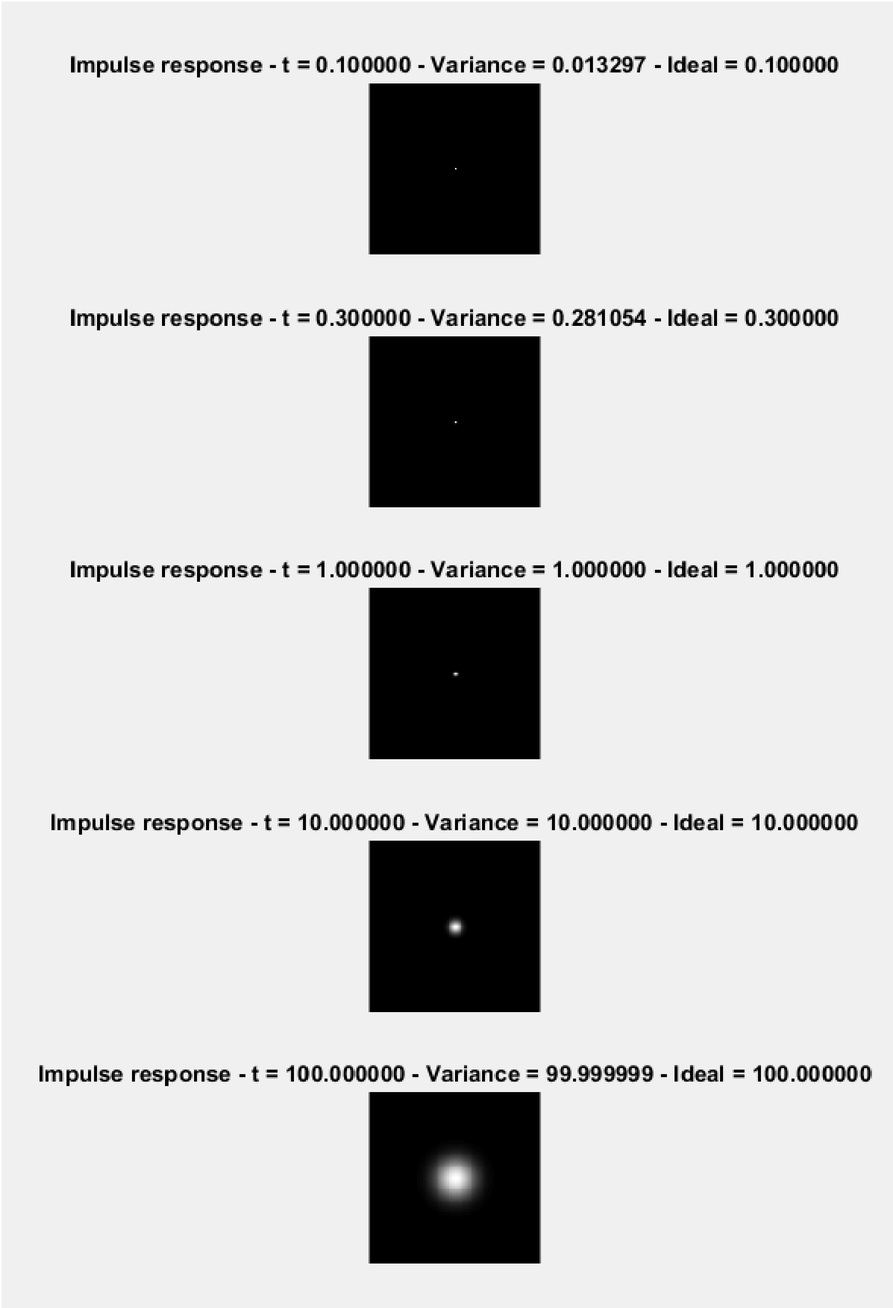

Figure 21: output of the impulse response of a Gaussian filter for different values of t.

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?**

If we use the same code that we used for the previous exercises for different images and values of t we can get an output similar to the one below:

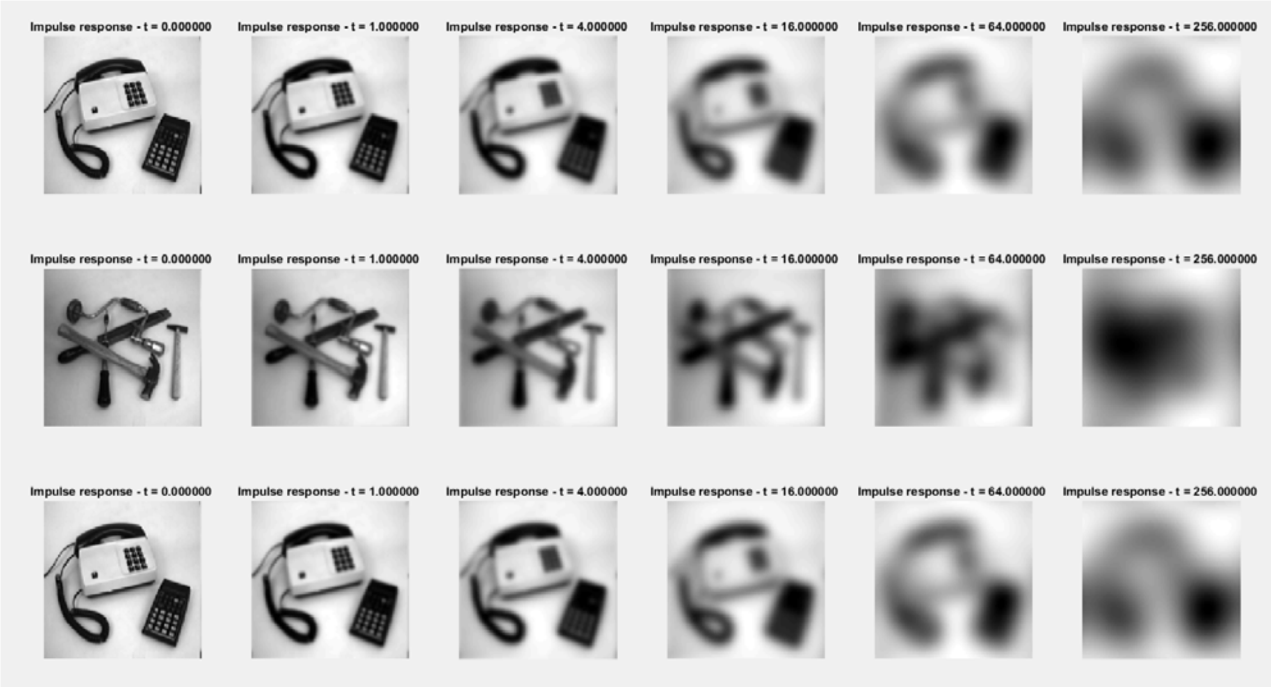

Figure 22: 3 images filtered with Gaussian filters with different values of the variance t

We can easily see that if we increase the value of t, the image gets more blurry. This is because the Gaussian filter is a low pass filter. If we represent the Gaussian filter in the Fourier domain, for the different values of t we get:

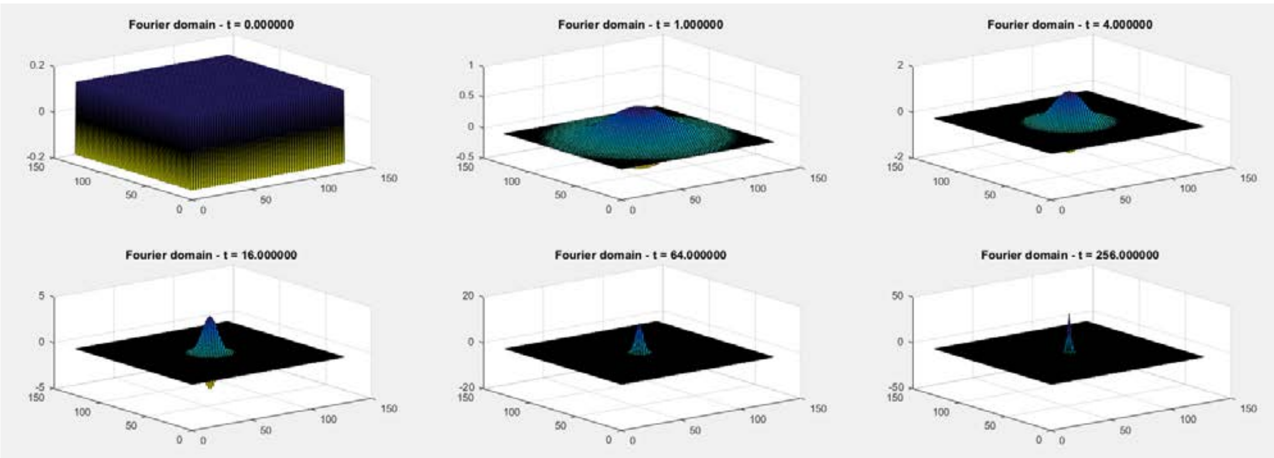

Figure 23: Gaussian distribution in the Fourier domain for different values of t.

We can observe that, the bigger the t the narrower the distribution, and then the higher frequencies are filtered, as only the frequencies inside the distribution are not filtered. This is why, the image definition get worse when more frequencies are filtered, as we are 'loosing' information.

8. Smoothing

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).**

For the Gaussian noise image, both the Median and the Gaussian filter work pretty good. They both manage to decrease the effect of the noise and still leave the edges of the objects of the image detectable. The median, however, works a little worse. This is because, when computing the median of the neighbor pixels, it is affected by the noisy pixels. On the other side, the Ideal filter doesn't work well as the noise only disappears when the image is too blurry and the edges are undetected. (Figure 24).

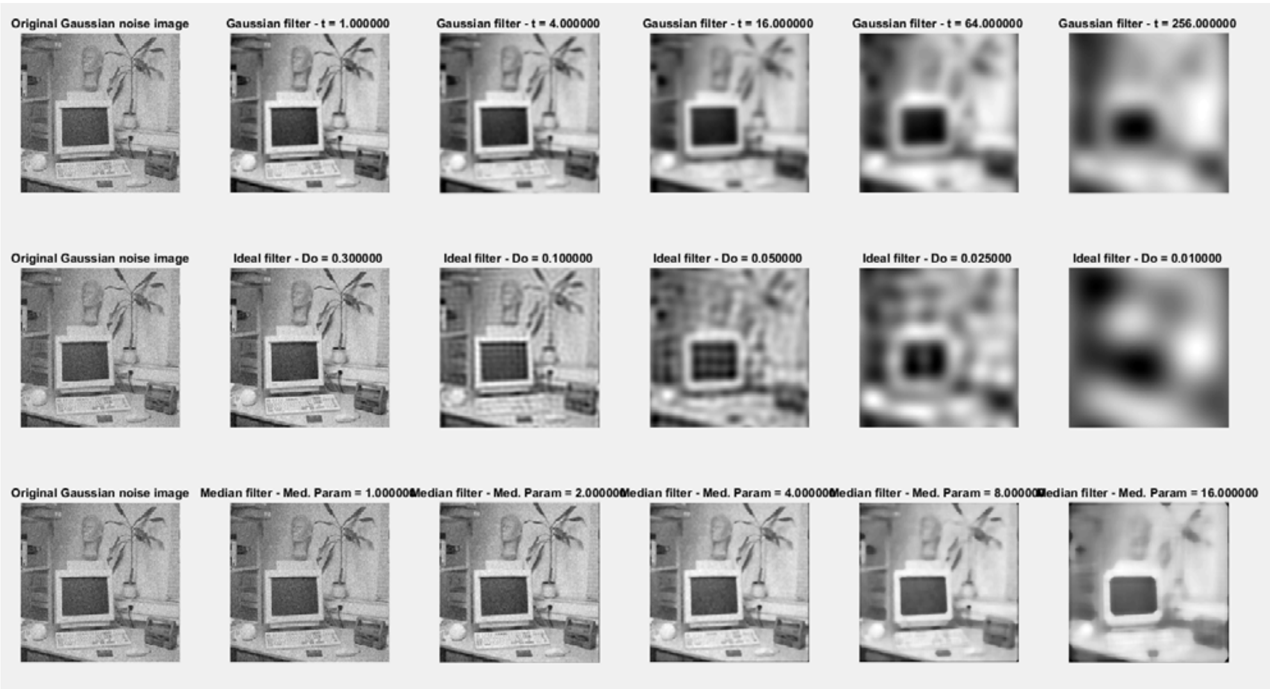

Figure 24: Results of using Gaussian, Ideal and Median filters with different parameters on an image with Gaussian noise.

If we analyze the figure 25, we can observe that only the Median filter works well with the Salt and Pepper noise. The noise disappears completely and the edges are perfectly visible. The image shows a painted–like effect. The Ideal filter shows the same output as when it was used with Gaussian noise picture, and the result is not good enough. The Gaussian filter is not as good as the median filter when filtering SAP noise as it needs to blurry the image too much and the edges are harder to detect.

Question 18: **What conclusions can you draw from comparing the results of the respective methods? Can you give a mathematical interpretation to explain the effects of each filter?**

The filter used in the previous exercise are just an example, there are many more filters and we cannot determine which of the filters is the best one, as it depends on the noise we want to filter and the type of image. As an example of this, we have the fact that the Gaussian filter was a better option for the Gaussian noise and the Median filter for the SAP one. Besides, we can use more than one filter in the same image in order to filter the noise.

The Gaussin filter, as it was explained before, is a low pass filter that filters the frequencies that are not included in the Gaussian distribution in the Fourier domain (high frequencies). As we filter high frequencies, we loose information and the image gets blurrier as we increase the value of the variance and the Gaussian distribution gets narrower.

The median filter calculates the median of each pixel's neiborghs, and substitute the value of every pixel for its median. This is why the edges are still preserved, but the surfaces looks like painted with brush.

The ideal filter multiplies by 1 the value of a frequence if it is greater than a cutoff threshold, and multiplies this frequence by cero if not. The frequency response is a rectangular function and it is consider a brick—wall filter.

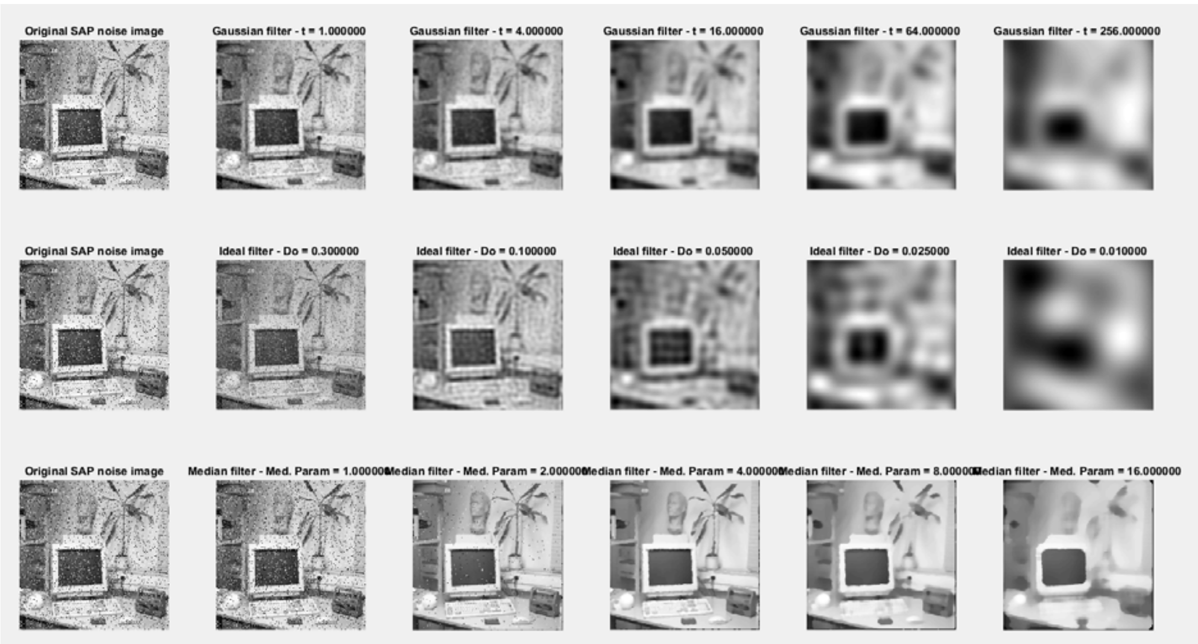

Figure 25: Results of using Gaussian, Ideal and Median filters with different parameters on an image with SAP noise.

9. Smoothing and subsampling

<u>*Question* 19:</u> **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.**

In the 4^{th} iteration in the subsampling of the original image we can observe some big pixels that are really different from their neighbors. (Figure 26 , first row) This is cause by the aliasing effect. This effect happens when the subsampling

frequency (Nyquist frequency) is lower than the image frequency and we get an output signal different from the original one. In other words, if this frequency is not high enough we can miss some information contained in the higher frequencies. This is why, in some part of the subsampled image for the iteration 4 there are this isolated different pixels.
To avoid this aliasing effect we can smooth the image before subsampling it. We have to use a low pass filter (Gauss or ideal for instance), so the higher frequencies are filtered and we make sure that all the frequencies of the image are lower than the Nyquist frequency, and that any frequency is interpreted incorrectly.

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.**

This was already explained in the previous question. If we don't smooth the image first (by applying a low pass filter) the higher frequencies will be processed wrongly. We need then to filter the higher frequencies to make sure that all the frequencies are subsampled correctly.
The figure 27 shows de difference between only subsampling, Gaussian smoothing and posterior subsampling and Ideal smoothing and posterior subsampling.

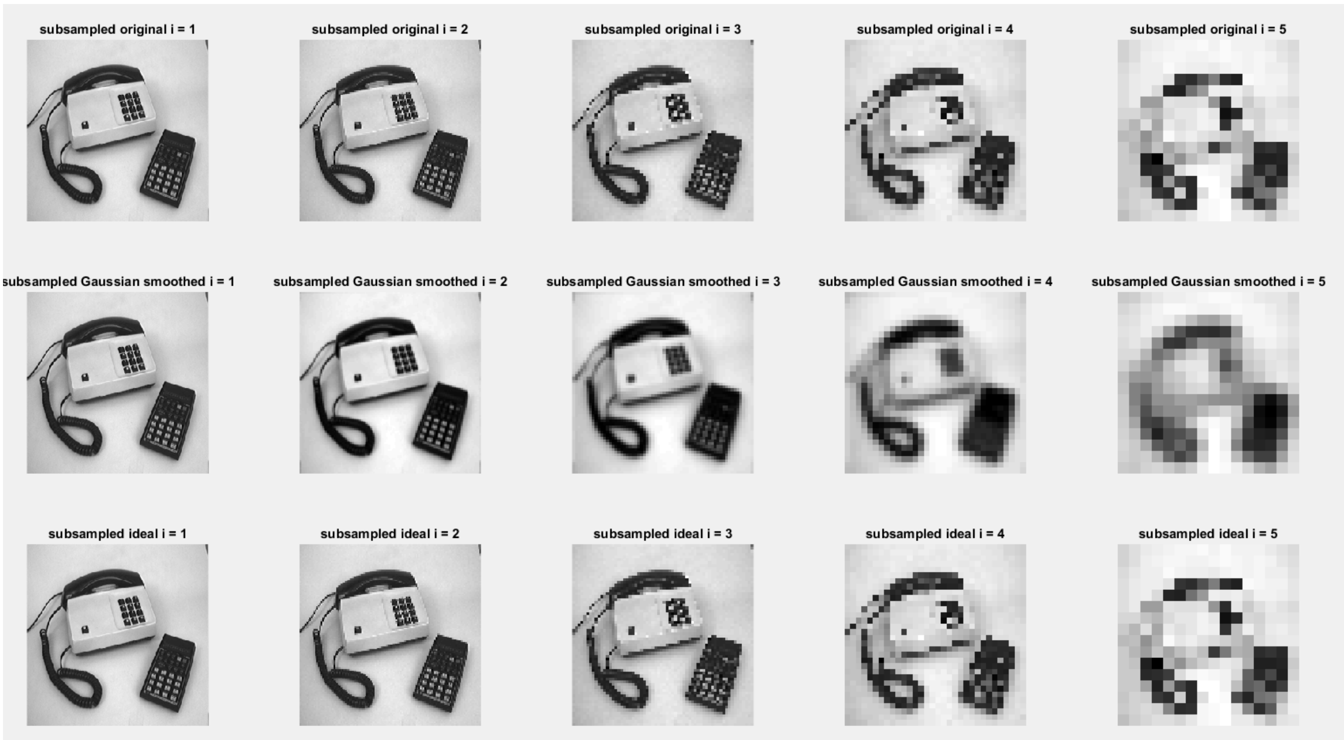

Figure 27: Iteration for: only subsampling, Gaussian smoothing and posterior subsampling and Ideal smoothing and posterior subsampling.

Laboratorion 1 in DD2423 Image Analysis and Computer Vision

. . . .
Student's personal number and name (filled in by student)

. . . .
Approved on (date) Course assistant