Answers to questions in Lab 1: Filtering operations

Name: *Javier López Iniesta Díaz del Campo* Program: *TSCRM*
Name: *Mathias Näreaho* Program: *TMAIM*

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?

We see as we change the values of p and q the orientation of the lines change. We see that the closer the value of p is to 1 or 128, the slower the variations caused in the vertical direction are. The same occurs with q, but with the horizontal variation. For example, $p = 1$ means no variations in the vertical direction, while $q = 1$ no variations in the horizontal direction.

Furthermore, we have figured out that with the third and the fourth case we obtain the same result but rotated approximately 90º (different sign in v_c of the centered $\hat{F}$), since the centered $\hat{F}$ is $(16, 8)$ and $(16, -8)$, respectively.

Finally, in the last case, we can see that the real part of the Fourier Transform is the same. However, the imaginary part is shifted, because we have a different sign in u_c of the centered $\hat{F}$.

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.

A position (p, q) in the Fourier domain will produce a sine wave in the spatial domain that varies along the direction of the vector. It goes from the center of the image towards the centered $\hat{F}$ (u_c, v_c).

The ratio of change of the sine wave in the spatial domain is proportional to the length of this vector. Hence, the further from the center, the faster the variations are.

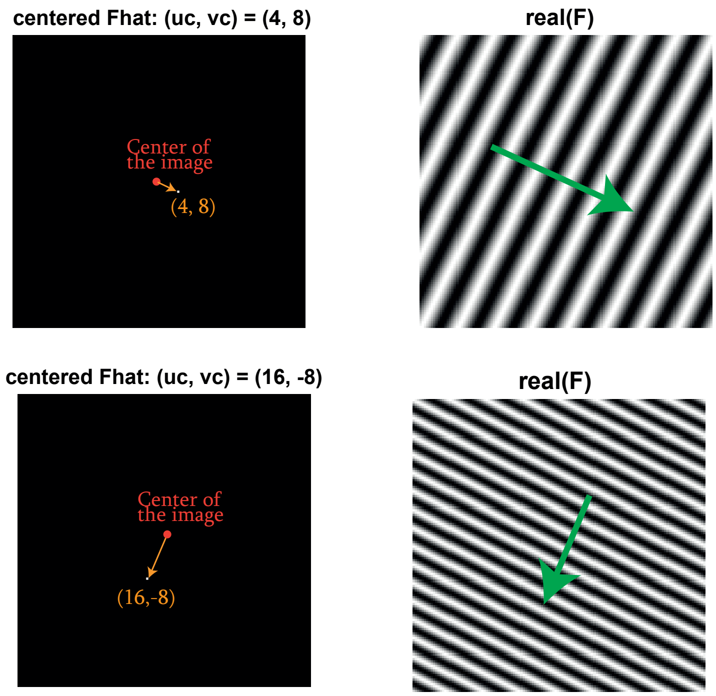


Figure 1: Examples of the variation of the vector's direction.

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

According to Eq. 4 [1] the total amplitude is $1/N$, where N is the total number of pixels. In our case, the total number of pixels is $s_z = 128$.

```
wavelength = sz/sqrt(uc^2 + vc^2);
amplitude = 1/sz;
```

Code 1: Updated code of Matlab to compute the amplitude and the wavelength.

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.

As we have stated in the Question 2, the direction of the vector depends on the vector between the center of the image and the centered coordinates of the image $\hat{F}$ (u_c, v_c).

The ratio of the change, or the frequency is proportional to the square root of the sum of square of each one of these coordinates:

$$w_1 = \frac{2\pi}{s_z} \cdot u_c \;\; ; \;\; w_2 = \frac{2\pi}{s_z} \cdot v_c \tag{1}$$

$$\lambda = \frac{2\pi}{\sqrt{w_1^2 + w_2^2}} = \frac{s_z}{\sqrt{u_c^2 + v_c^2}} \tag{2}$$

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!

When we pass the point in the center, the frequencies are in the range $[\pi, 2\pi]$, which are equivalent to the frequencies in the range $[-\pi, 0]$.

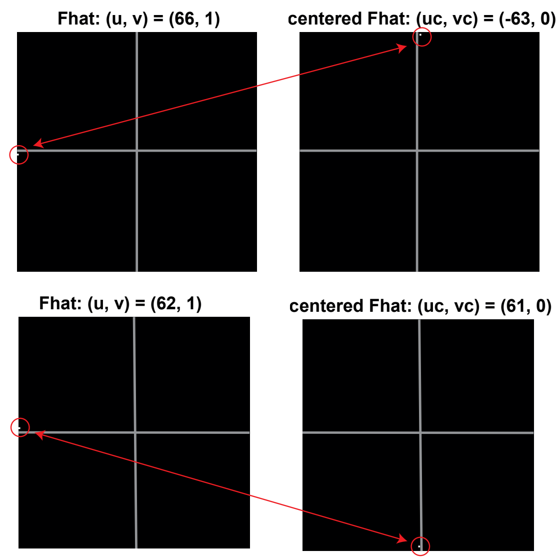


Figure 2: Examples to see what happens if the values of p or q exceed the half of the image.

Therefore, as we can see in Figure 2 when we cross the halfway point $(N/2)$ of the image the center point will move from one part of the original image to the opposite direction in the spatial image $(-\pi \rightarrow +\pi)$.

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

This part of the codes determines the position (u_c, v_c) of the centered $\hat{F}$. If the point u is bigger than the half the pixels of the image, to compute u_c we subtract the total number of pixels plus 1 to the coordinate u. If not, we only subtract 1 to the coordinate u. Analogously, we do to the same to compute v_c.

The objective of this instructions is to translate the period of the image from $[0, 2\pi]$ to $[-\pi, \pi]$ because it is easier to understand.

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!

If we look at the source images we see that F image only shows variations along the vertical direction, the G image along the horizontal direction and the H shows both directions at the same time, but not diagonal ones.

According to this, if we apply the Fourier Transform to this images we see that the Fourier spectra is only different from 0 in the first row/column of one of its coordinates, whereas the others are set to 0. For instance, $\hat{F}$ the x coordinates is fixed 0 and only it is different from 0 in the first column (no horizontal variation). The same occurs with Ghat that the y coordinate is fixed to 0 (no vertical variation). Finally, we see that $\hat{H}$ corresponds to a combination of $\hat{F}$ and $\hat{G}$, where is formed a star in the middle of the image.

Question 8: Why is the logarithm function applied?

We need the logarithm function to make the finer details appear, as we have a large range of values (with different orders of magnitude) that is compressed into a grey scale.

In addition, the term “1” of the expression is used to maintain the 0 level in the pixels that previously were 0.

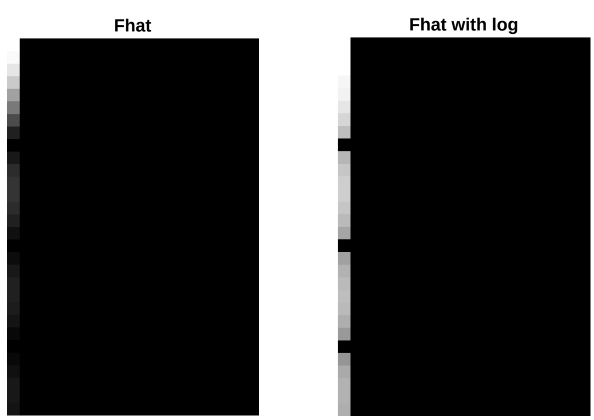


Figure 3: $\hat{F}$ with and without the logarithm function.

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

We can see that the Fourier transform of $F + 2G$ is a superposition of the Fourier transformations of F and 2G, following the property:

$$\mathcal{F}(F + 2G) = \mathcal{F}(F) + 2 \cdot \mathcal{F}(G)$$

Or in the general case, we can write: $\mathcal{F}(aF + bG) = a \cdot \mathcal{F}(F) + b \cdot \mathcal{F}(G)$

If for example we observe the pixel (0, 0) of our images wee see that $\hat{H}(0, 0) = 6144$ which is equal to $\hat{F}(0, 0) = 2048$ plus $2 \cdot \hat{G}(0, 0) = 2048$.

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.

Instead of multiplying in the Fourier domain we can make a convolution in the spatial domain.

$$\hat{Z} = \mathcal{F}(F \cdot G) \tag{3}$$

$$\hat{Z} = \hat{F} * \hat{G} \tag{4}$$

```
% Zhat with multiplication in the spatial domain
     Zhat_1 = fft2(F.* G);

% Zhat with convolution in the Fourier domain
     Fhat = fft2(F);
     Ghat = fft2(G);
     Zhat_2 = conv2(Fhat,Ghat);

     M = length(Fhat);
     N = length(Ghat);
     Zhat_2_norm = Zhat_2(1:M, 1:N)/(M * N);
```

Code 2: Multiplication.

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.

As we have made a compression in the spatial domain, we have an expansion in the Fourier domain.

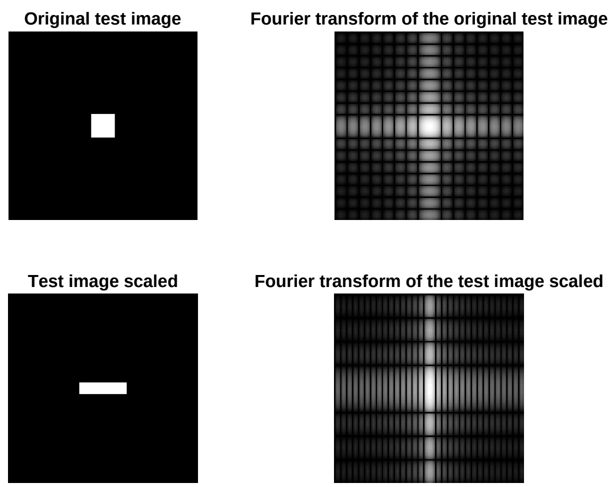


Figure 4: Compression in the spatial domain.

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

As we can see in Figure 5, after the rotation of the Fourier transformed image, a distortion appears at 30º and 60º, whereas with 45º and 90º we preserve the transformed image more less the same. After rotating back, the original image is restored, but plotted at an angle. In the case of distortions, these remain after rotating back.

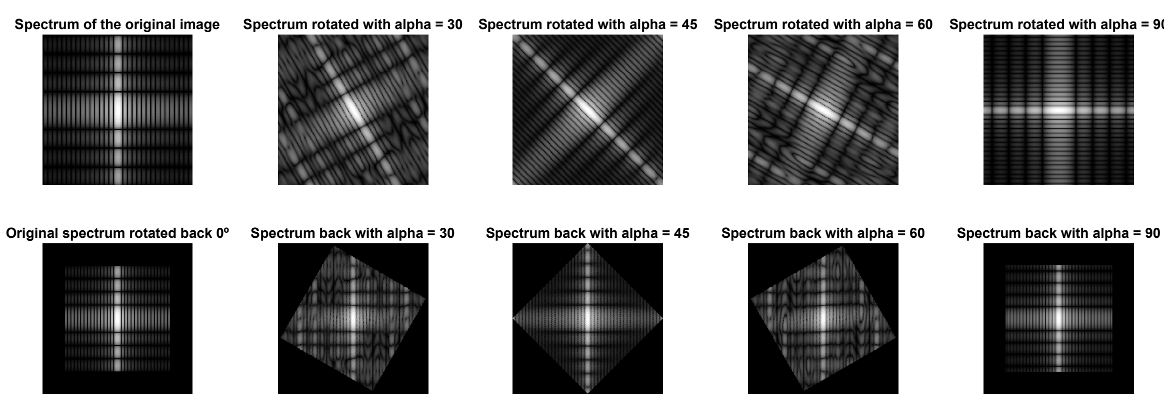
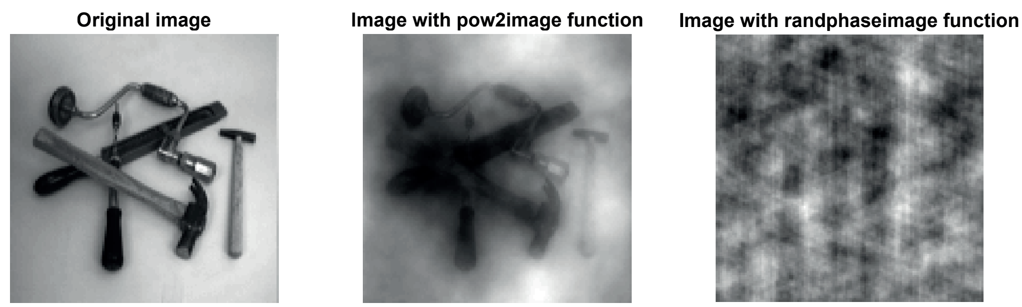


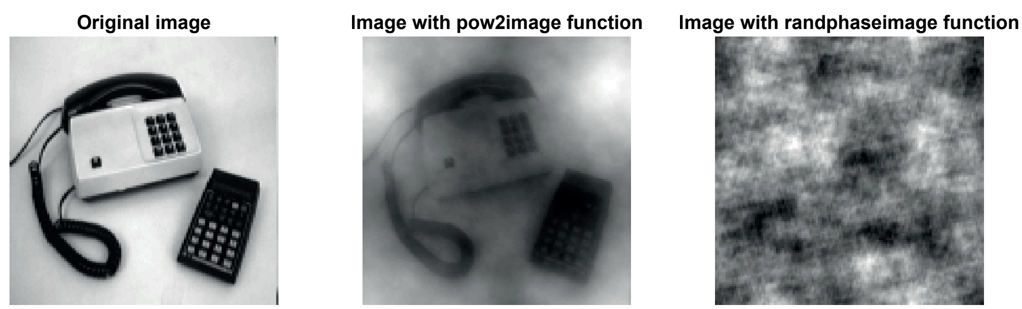
Figure 5: Rotations of Fourier transform.

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

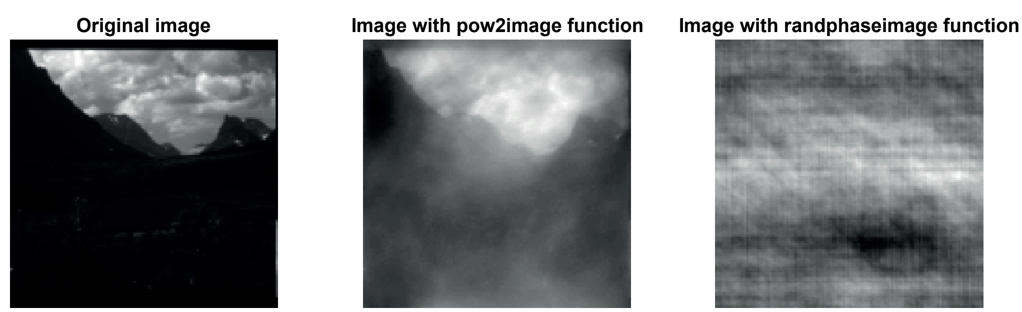
The phase contains information about where edges will end up in the image. When we randomize the phase, we lose information about the edges in the image. The magnitude contains information about what grey-levels to put on either side of the edges. This is illustrated in Figure 6.



(a) few128image.



(b) phonecalc128 image.



(c) nallo128 image.

Figure 6: Different images filtered with *pow2image* and *randphaseimage* filters.

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?

After computing the discretized Gaussian kernel g , the expected covariance matrix of the kernel with variance t should ideally be:

$$Cov(g) = \begin{bmatrix} t & 0 \\ 0 & t \end{bmatrix} = tI$$

While our covariance matrices were multiples of the identity matrix, the variance was not equal to the input variance for all values of t. For $t \in \{1, 10, 100\}$ the covariance matrix fulfilled $Cov(g) = tI$, but for the remaining values of t the resulting variance was $\sigma^2(t = 0.1) = 0.013$ and $\sigma^2(t = 0.3) = 0.281$.

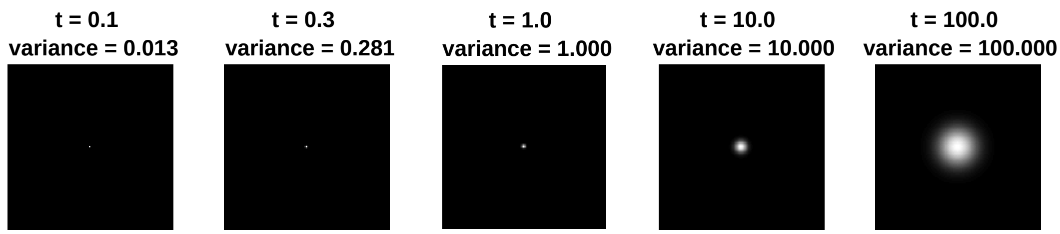


Figure 7: Impulse response for different values of t.

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.

The results differ for the smaller values of t, and are similar for higher values of t. In the ideal continuous case, there is no discretization being made, and thus the smaller variance can catch the behaviour between the discrete points. In the discrete case, the calculations do not work out with such a low variance, as the grid is simply too coarse.

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?

Increase values of the variance is increasing the blurring effect of the Gaussian kernel. For the largest values of the variance, only the darkest and lightest areas are the same color as the original image.

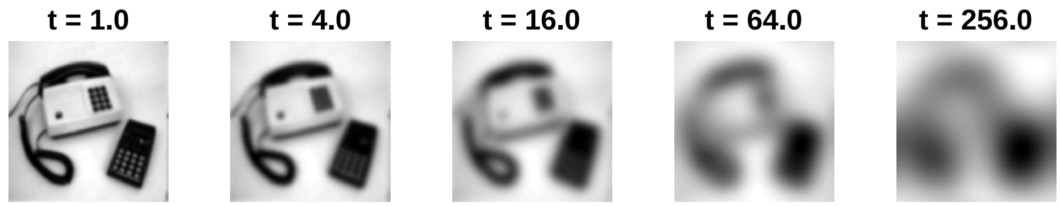
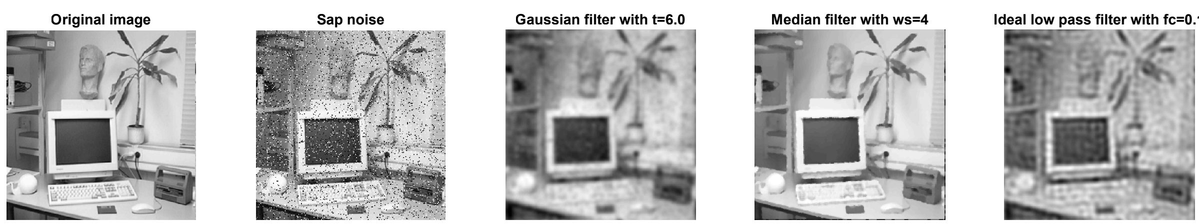


Figure 8: Image filtered with different values of the variance.

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).

(a) Gaussian noise.



(b) SAP noise.

Figure 9: Noisy images filtered with different types of filters.

- **Gaussian filter**: In this filter the value of each pixel is a weighted sum of the neighboring pixels. This weight decreases as the distance from the pixel increases. Regarding the results, it provides good results when you are dealing with white noise, like the Gaussian noise since the new pixel values are a weighted average of its surroundings pixels. However, it achieves poor results when dealing with Salt-And-Pepper (SAP) noise since the corrupted pixels will turn into wider ones.
- **Median filter**: In this filter we pick a window of n by n around each pixel and we choose the value in the median to be the final value of it. The resulting images look as a drawing if the window size of the filter is very big. Thus, we appreciate good the edges, although we lose a lot of details in the final image. It yields good results when dealing with Gaussian or SAP noise, because the corrupted pixels will be always in the edge of the window.
- **Ideal low pass filter**: Finally, this filter removes quick changes in the image higher than the cut-off frequency. This filter produces worse results than the Gaussian filter or the

Median filter because it does not remove the lower frequencies of the Gaussian noise and extends the corrupted values of the pixels with SAP noise.

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

As we have stated in the Question 17, it seems that we obtain better results for the Gaussian noise with a Gaussian kernel of input variance 1. Whereas, for the SAP noise a Median filter with window size 4 is the filter that gives us the best results.

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.

We observe that less information in the image is lost when we apply smoothing before subsampling. The filters both work fine for the fourth iteration (Fig. 11), but the Gaussian is blurring the image more, e.g the button on the phone is becoming a smudge, unlike in the low-pass filtering where the button is clearly visible. In order for the filtering to work well, we need to find a balance, as too much blur will cause loss of information in the image, and too little blurring will leave us with aliasing.

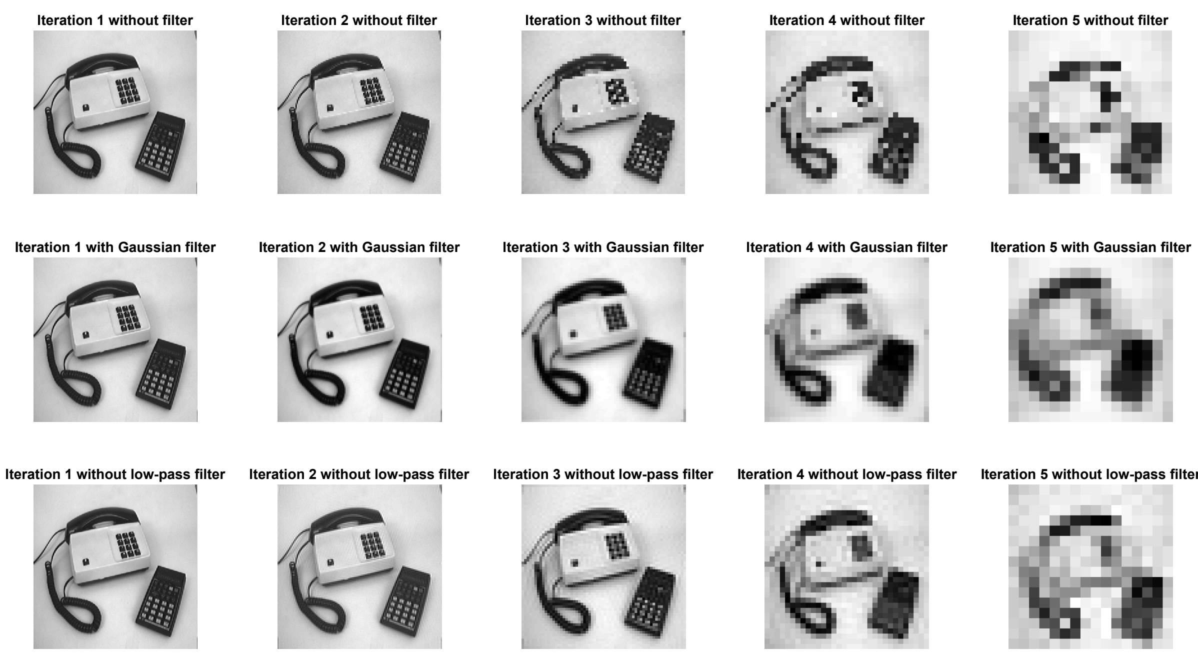


Figure 10: Smoothing and subsampling.

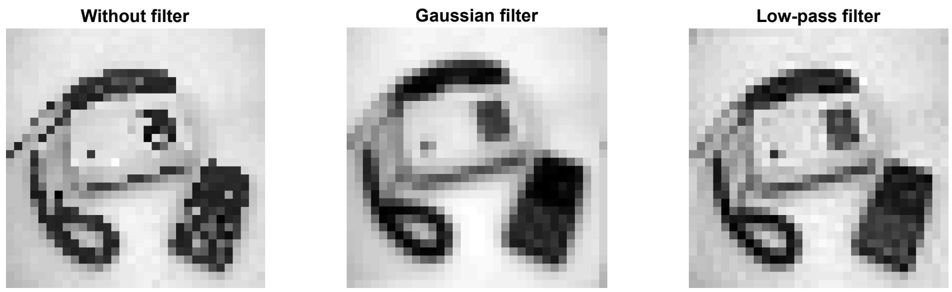


Figure 11: Smoothing and subsampling: Iteration 4.

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.

When we sample the image, we get aliasing, as the features are harder to distinguish (keypad, buttons). One of the best ways to avoid this is to first blur the image (in our case by Gaussian filter or Low-Pass filter) and then to do the subsampling. If we do it in the opposite order , we will simply get a blurred and aliased image.

References

[1] School of Computer Science and Communication, “Lab 1: Filtering operations,” 2022, [Online] Available: https://canvas.kth.se/courses/36072/files/5897822?wrap=1 [Accessed: 08-11-2022].