DD2423 Image Analysis and Computer Vision
Lab#1

Xinmiao YU, xinmiao@kth.se
Giovanni Castellano, giocas@kth.se

Date: 2022/11/16

1.3 Basis functions

1. A dirac impulse in the fourier domain corresponds to a sinusoidal pattern in the spatial domain. In particular we observe that both the real and imaginary part of the function in the spatial domain behave as a sinusoidal function. Since the imaginary part is not zero we conclude that with only one dirac impulse the corresponding function in the spatial domain is a complex function.
   We also observe that the direction of the sinusoidal pattern corresponds to the direction of the dirac Impulse. Moreover, as we increase the distane from the center of the spectrum to the impulse, the frequency of the sinusoidal pattern increases.

2. Based on equation (4)

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

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

$$= \frac{1}{N} \hat{F}(p,q) [cos(\frac{2\pi (p,q)^T x}{N}) + isin(\frac{2\pi (p,q)^T x}{N})]$$

   As we can observe in the equations, the complex exponential in the formula can be splitted into a cosine and a sine. It is clear from the formula that the position of the dirac impulse affects the frequency and the direction of the sinusoidal pattern in the two dimensional space. An example is shown in figure 1

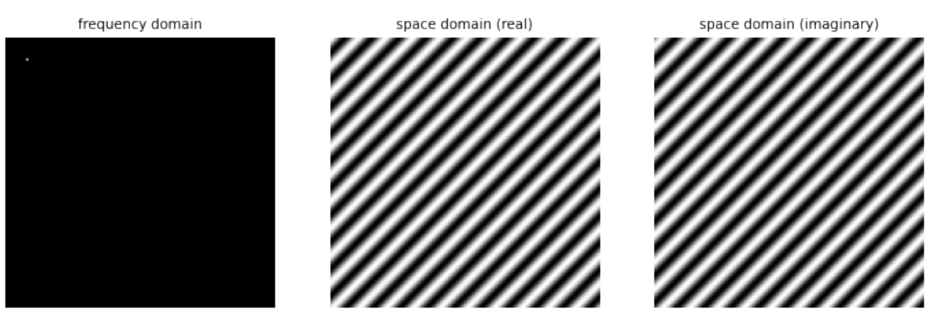


Figure 1: Inverse Fourier transform of a dirac impulse.

3. From the formulas presented in Question 2 we know that:

$$F(x) = \frac{1}{N}\hat{F}(p,q)[cos(\frac{2\pi(p,q)^T x}{N}) + isin(\frac{2\pi(p,q)^T x}{N})]$$

if $\hat{F}(p,q)$ is a dirac Impulse then, the real part of the function in the space domain will be a cosine with amplitude $\frac{1}{N} = \frac{1}{128}$, the imaginary part instead will be a sine function with the same amplitude. Then the total amplitude of the function is $\sqrt{Re^2 + Im^2}$ which depends on x.

4. $\omega_1 = \frac{2\pi u_c}{N} = \frac{2\pi u_c}{sz}, \omega_2 = \frac{2\pi v_c}{N} = \frac{2\pi v_c}{sz}$

$$\begin{aligned}\lambda &= \frac{2\pi}{\sqrt{\omega_1^2 + \omega_2^2}} \\ &= \frac{2\pi}{\sqrt{(\frac{2\pi u_c}{sz})^2 + (\frac{2\pi v_c}{sz})^2}} \\ &= \frac{sz}{\sqrt{u_c^2 + v_c^2}}\end{aligned}$$

where u_c, v_c is determined by (p,q).

5. If we have $p > sz/2$ or $q > sz/2$, then the coordinates of the point after centering the spectrum (u_c, v_c) should be calculated in a different way: $v_c = v - sz$, $u_c = u - sz$. We observe that a point in the corner of the spectrum correspond to a point in the middle of the spectrum after centering (Fig. 2).

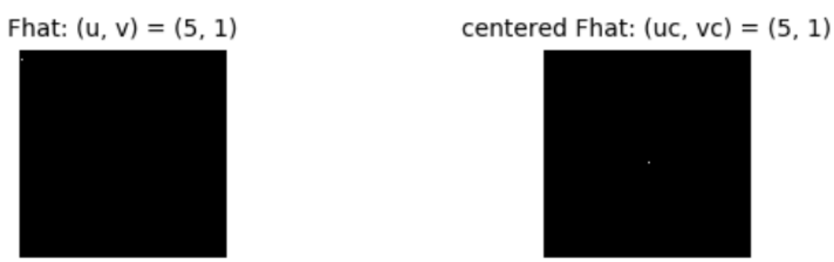


Figure 2: center (p,q) = (65,65)

6. The goal of those instructions is to calculate the coordinates of the dirac pulse after centering the spectrum. We then use the new values to compute the wavelength.

1.4 Linearity

7. Based on the given discrete Fourier transform equation

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

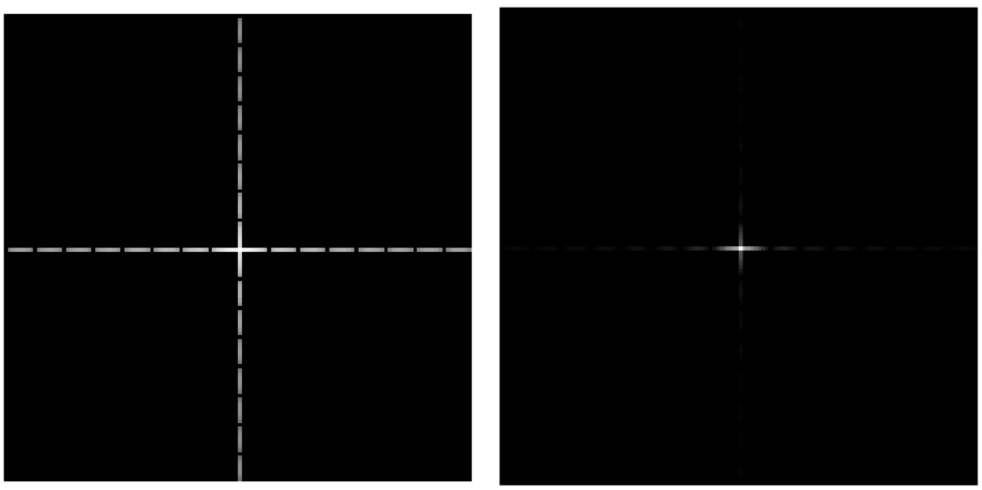
$$F(u,v) = \frac{1}{N}\sum_{x,y=0}^{128}\hat{F}(x,y)e^{\frac{-2\pi i(ux+vy)}{N}}$$

Based on how F is assigned the values, only rows $56 - 71$ contain one. Then for F, the formula could be written as

$$F(u,v) = \frac{1}{N}[\sum_{x=56}^{71} e^{\frac{-2\pi iux}{N}} [\sum_{y=0}^{128} e^{\frac{-2\pi ivy}{N}}]]$$

Both the sums contains exponential terms that go to zero as x or y increases, that's why the FFT is concentrated to the borders. Moreover, one of the two sums contains more terms than the other one, therefore we observe highest values in one of the two borders depending on direction of the white line in the spatial domain.

8. The `log` function reduces the difference between the lowest and highest value. The low frequencies become closer to the high frequencies and therefore the distribution of the colors among frequencies becomes more uniform, as shown in Fig. 3.



(a) `showgrey(np.log(1 + np.abs(Hhat)))` (b) `showgrey(1 + np.abs(Hhat))`

Figure 3: Compare the result with and without `log`

9. Since the fourier transform is linear we can conclude that the fourier transform of the linear combination of two functions is equal to linear combination of the fourier transforms of the functions:

$$F(aH + bG) = aF(H) + bF(G)$$

The linearity of the Fourier transform comes from the linearity of integrals.

1.5 Multiplication

10. Multiplication in the spatial domain corresponds to convolution in Fourier domain. Instead of multiplying the functions and then compute the FFT, we can first compute the FFT and then compute the convolution of the results: `showfs(fftshift(convolve2d(fftshift(Fhat) / 128, fftshift(Ghat) / 128, 'same')))` (Fig. 4).

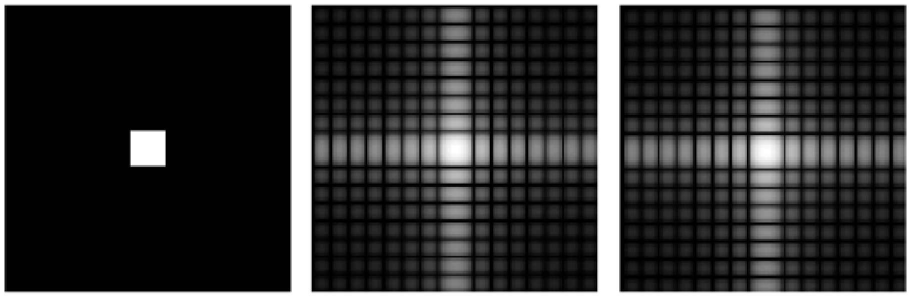
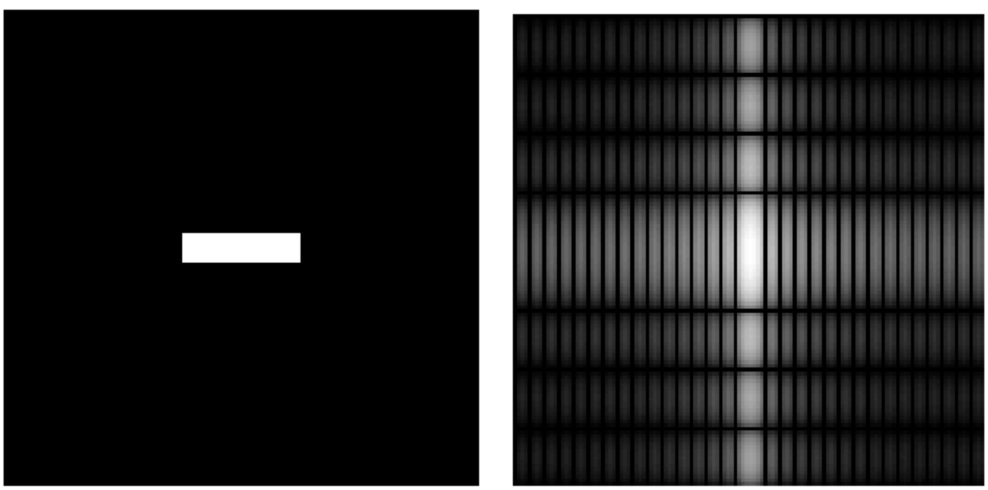


Figure 4: Multiplication in space domain is equal to convolution in fourier domain (last two figures are the same)

1.6 Scaling

11. Compression in spatial domain will result in an expansion in Fourier domain. As we can see, we compressed the y axis in the spatial domain and it resulted in a expansion in the y axis of the fourier domain. The x axis instead was expanded in the spatial domain and compressed in the fourier domain (Fig.5).

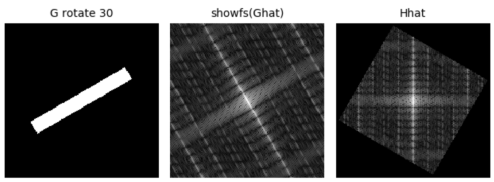


(a) `showgrey(F)` (b) `showfs(fft2(F))`

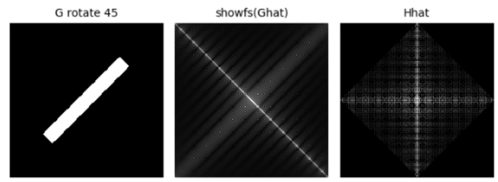
Figure 5: Scaling result

1.7 Rotation

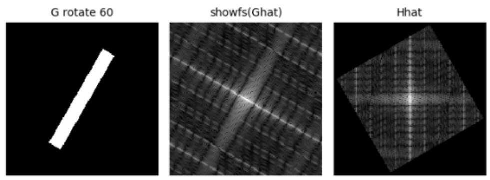
12. Rotating the spatial domain, we observe a rotation of the same angle in the fourier domain. The frequencies are the same in a different direction (Fig. 6).



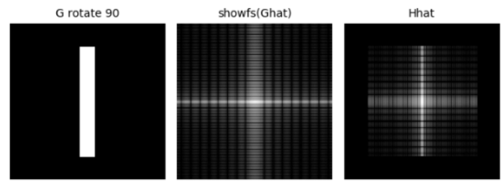
(a) `alpha=30`



(b) `alpha=45`



(c) `alpha=60`



(d) `alpha=90`

Figure 6: Rotation result with different angles

1.8 Information in Fourier phase and magnitude

13. As `pow2image` keeps the phase and `randphaseimage` keeps the magnitude, we can conclude that the phase keeps information about edges while the magnitude keeps information about intensity. However, phases are more important than magnitudes.

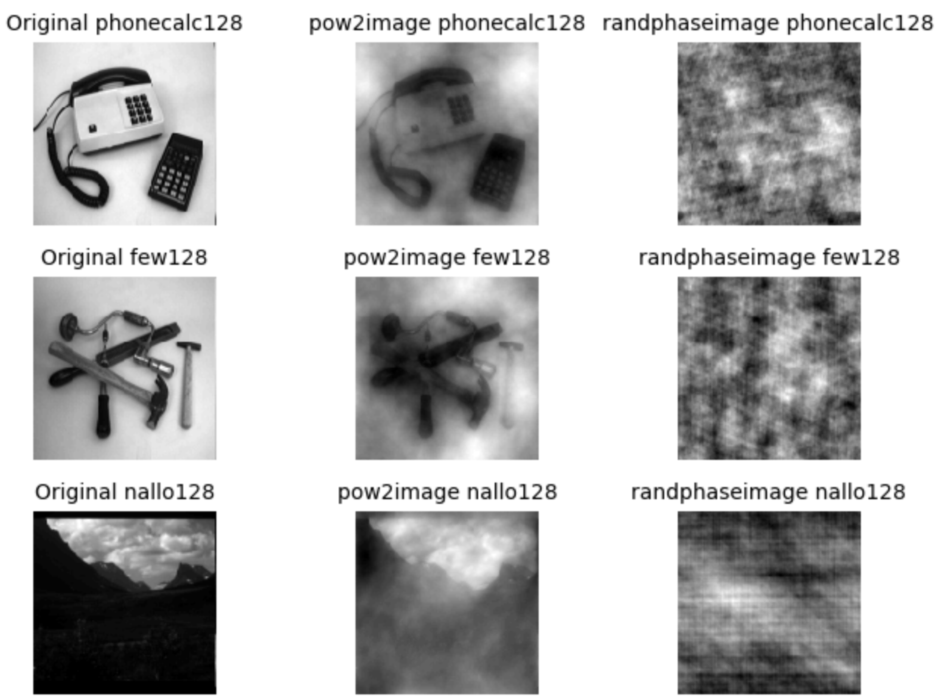


Figure 7: Fourier phase and magnitude

2.3 Filtering procedure

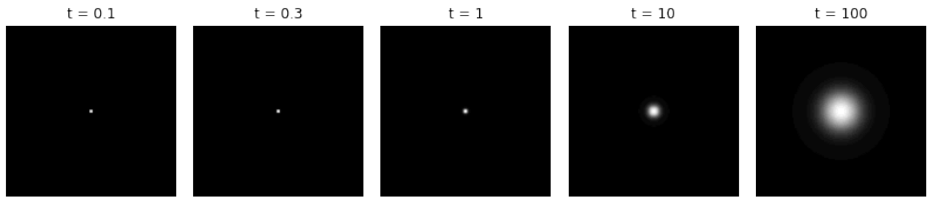


Figure 8: Gaussian filter impulse response as the variance increases.

14. Figure 8 shows the impulse response of the Gaussian filter. As we can observe after filtering a Dirac impulse with a Gaussian filter we get as output the same Gaussian.
    The tables below show the covariance matrices of the resulting gaussian for different values of t:

t = 0.1		t = 0.3		t = 1	
$2.50077e-01$	$-5.38415e-14$	$3.15692e-01$	$3.57091e-14$	$9.84436e-01$	$9.67161e-14$
$-5.38415e-14$	$2.50077e-01$	$3.57091e-14$	$3.15692e-01$	$9.67161e-14$	$9.84436e-01$

t = 10		t = 100	
$9.84436e+00$	$-3.60300e-14$	$9.84436e+01$	$1.50905e-13$
$-3.60300e-14$	$9.84436e+00$	$1.50905e-13$	$9.84436e+01$

15. As we can observe, the element of the matrices which are not in the diagonal are almost equal to zero. The elements of the diagonal instead are approximately equal to the variance. The approximation however is not good when the variance is low.

16. We now convolve the images in figure 9 with a gaussian filter to observe how it affects the images. Figure 10 and 11 show that the output image gets blurrier as we increase the variance of the gaussian filter, this happens because the filter removes high frequencies from the image.

Figure 9

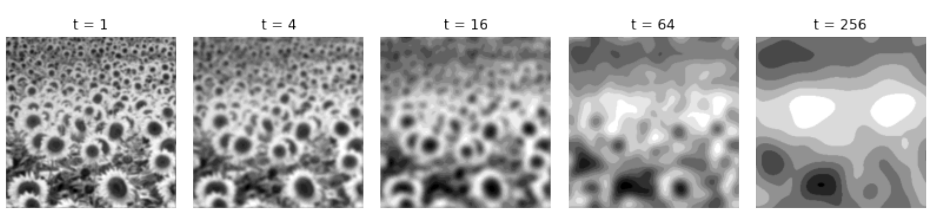


Figure 10: Image sunflower256.npy filtered with different gaussian filters.

Figure 11: Image cern256.npy filtered with different gaussian filters.

17. Below we try to remove gaussian noise and salt and pepper noise from an image using different types of filters.

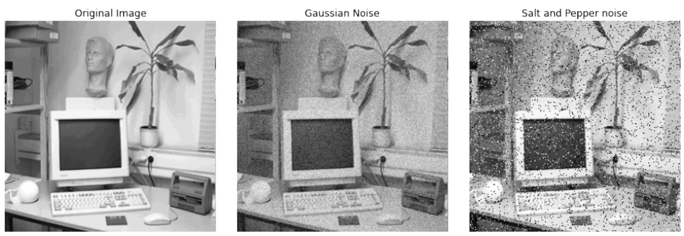


Figure 12: Different type of noise

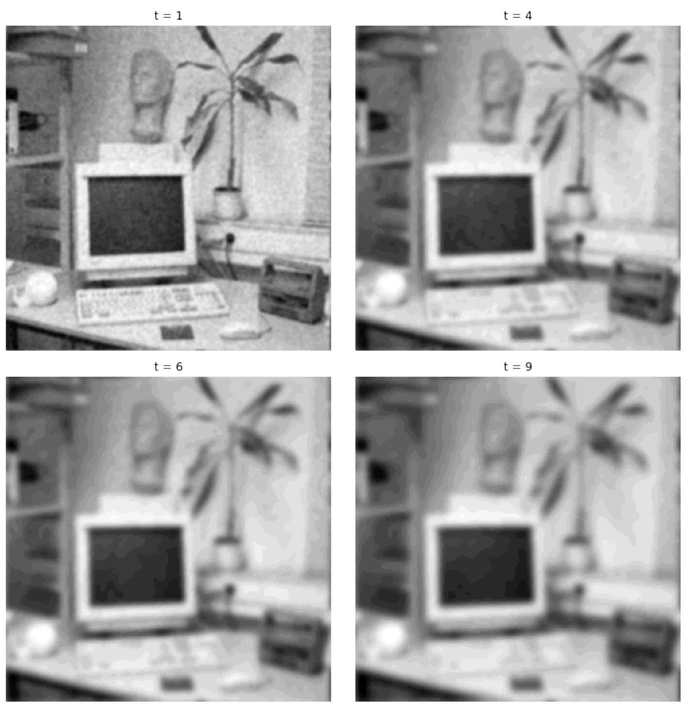


Figure 13: Noisy image (gaussian noise) denoised with gaussian filter.

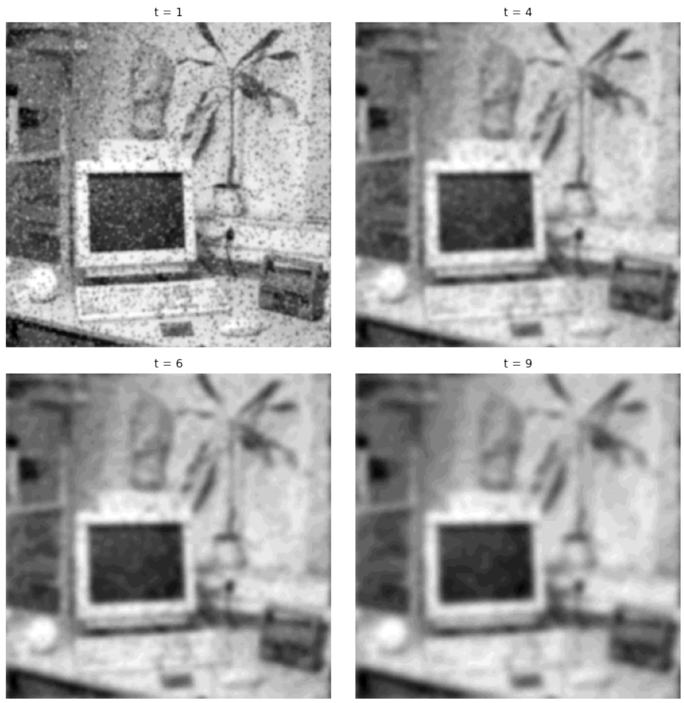


Figure 14: Noisy image (SAP noise) denoised with gaussian filter.

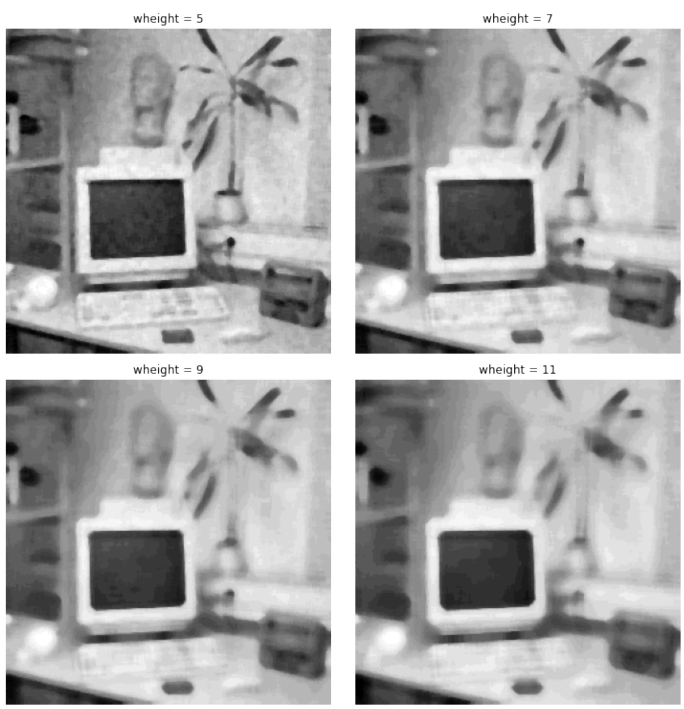


Figure 15: Noisy image (Gaussian noise) denoised with median filter.

Figure 16: Noisy image (SAP noise) denoised with median filter.

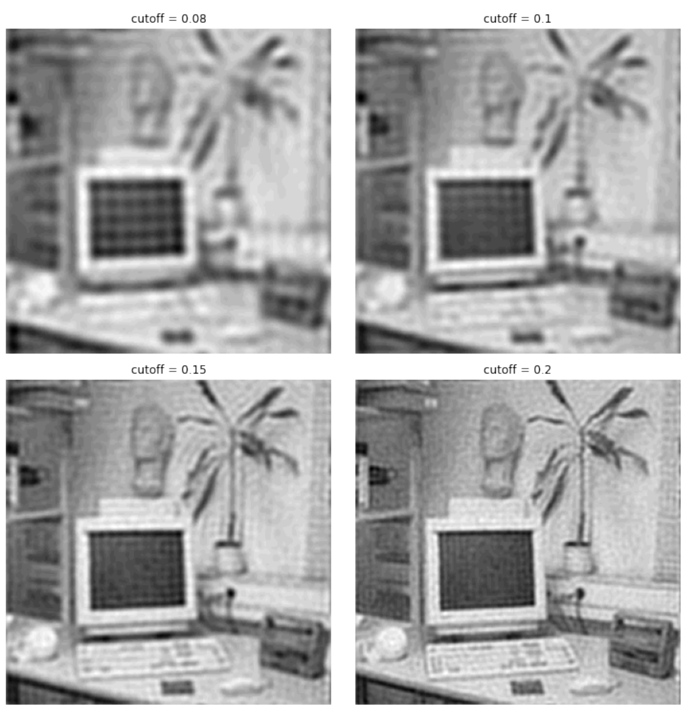


Figure 17: Noisy image (Gaussian noise) denoised with ideal low-pass filter.

Figure 18: Noisy image (SAP noise) denoised with ideal low-pass filter.

Gaussian Filter:
Using gaussian filter seems that we are able to remove gaussian noise but the image becomes blurry very quickly as we increase the variance (figure 13).
Trying to remove SAP noise instead, we get images that seem blurry and noisy at the same time. We conclude that the gaussian filter is not a good choice to remove SAP noise (figure 14).

Median Filter:
Filtering with median filter we are able to completely remove the SAP noise and at the same time preserving the edges of the picture (figure 16).
We obtain good results also when we try to remove the gaussian noise with the median filter; however in this case, the image seems blurrier (figure 15).

Ideal low-pass Filter:
The ideal low-pass Filter doesn't perform very well in these experiments (figure 17 and figure 18). We observe that the filter is not able to remove either type of noise. Moreover, as we suppress high frequencies the image looks very strange and not pleasing to the eye.

18. In general we can conlude that the median filter is the best one to remove these 2 types of noises. Morever, although noise is usually distributed among high frequencies, using a low-pass filter is not enough to denoise the image.

19. In figure 19 we illustrate how performing the smoothing before downsampling an image affects the quality of the result. We observe that thanks to smoothing we get a better quality image. However, if we remove too many pixels the image will be hard to recognize anyway. It seems from our experiments that the gaussian filter performs better for this task.

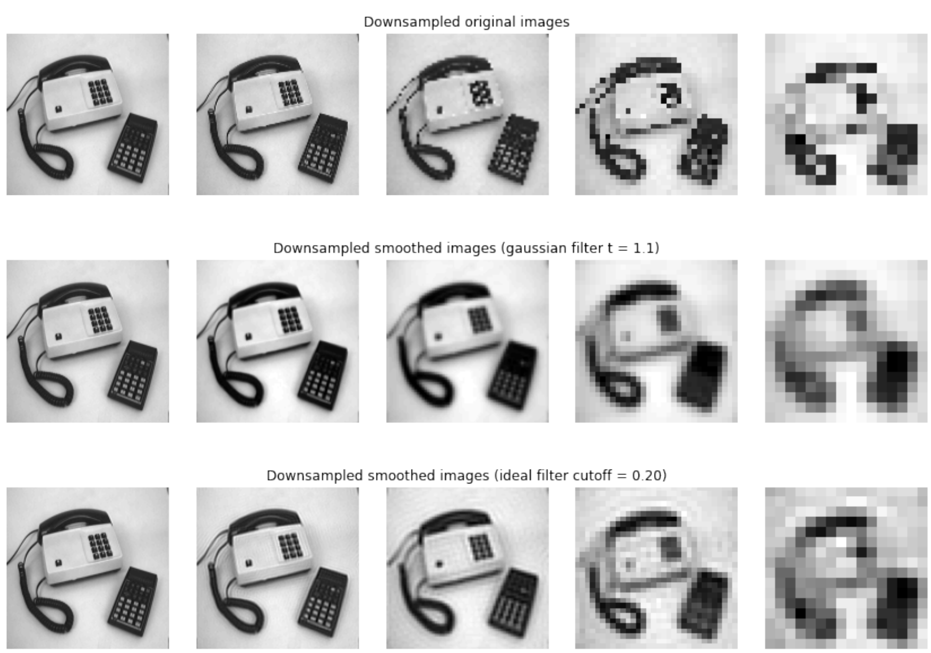


Figure 19: How smoothing affects downsampling.

20. It makes sense that smoothing helps to preserve the quality of the image. In fact, we know from

Nyquist theorem that in order to perfectly reconstruct a signal after sampling, the sampling frequency must be higher than two times the highest frequency of the signal. Performing smoothing we remove the high frequencies of the image, therefore, the Nyquist threshold decreases and we can downsample the image without loosing too much quality.