DD2423 Image Analysis and Computer Vision

Lab 1: Filtering Operations

Dario Del Gaizo, Giulia Rebay, Isabella Rositi

17 November, 2022

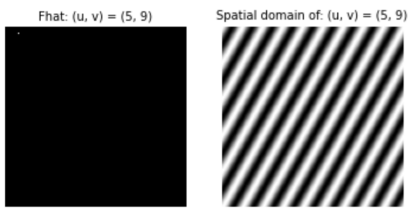
Question 1: Calculate the inverse discrete Fourier transform of an image $\hat{F}$, that is zero everywhere except for a single point (p, q) at which the value is one. Look at its real and imaginary parts, as well as its magnitude and phase. Repeat this exercise with the coordinates p and q set to (5, 9), (9, 5), (17, 9), (17, 121), (5, 1) and (125, 1) respectively. What do you observe?

We observe 6 different plots:

- 2-dimensional discrete Fourier Transform
- 2-dimensional centered discrete Fourier Transform
- real part of the inverse of the 2-dimensional discrete Fourier Transform
- imaginary part of the inverse of the 2-dimensional discrete Fourier Transform
- absolute value (with amplitude label set at 0.0 initially)
- angle of the complex argument (with wavelength label set at 0.0 initially).

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 Python figure

A position (p,q) in an image F in the Fourier domain will be projected as a sum of sine waves of different amplitude, frequency and phase decomposed between real(F) and imag(F).



Question 3: How large is the amplitude? Complement the code (variable amplitude) accordingly.

Amplitude is given by the absolute value of the Fourier Transform, which in 2D can be seen as $\frac{1}{MN}$ which in the square image case is $\frac{1}{N^2}$.

In the code, the amplitude is given by the variable: amplitude = 1/sz**2.

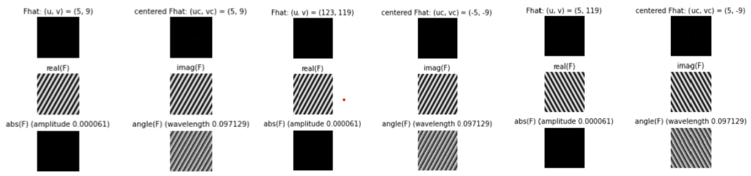
Question 4: How does the direction and length of the sine wave depend on p and q? Complement the code (variable wavelength) accordingly.

The slope of the sine wave is given by the slope of $m = \frac{v}{u}$. The direction of the wave is perpendicular to the slope, as seen in the lectures as $\phi = tan^{-1}(\frac{Im[\hat{F}]}{Re[\hat{F}]})$. The wavelenght is given by $\lambda = \frac{1}{\sqrt{u^2+v^2}}$.

In the code, the wavelength is given by the variable: wavelength = 1/np.sqrt(uc**2 + vc**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 Python.

Images with both coordinates symmetric with regards to the center $\frac{sz}{2}$ have the same frequency, the same wavelength and same phase. Those that have one point with the same coordinate, and the other one symmetric have the same frequency and wavelength, but opposite phase.



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

It provides the numerical shift that is performed by *np.fft.fftshift()*. Mathematically this means that it centers the coordinates of the image with respect to $\frac{sz}{2}$. Visually this means switching the first and third quadrant and second with the fourth.

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!

Matrix F has the greatest greyscale variation in the vertical direction, whereas matrix G, being defined as F.T, has greatest variation in the horizontal direction. This trend is clearly depicted by the Fourier spectra of both matrices, which only display variation in these two senses. This means that the frequencies associated to such images are 0 and $\frac{\pi}{2}$, which are then combined in Hhat's spectra to the borders of the image.

Question 8: Why is the logarithm function applied?

Log transformations are used to compress large dynamic ranges and make details visible (rescaling of the range).

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

We have defined H as a linear combination of matrices F and G, with coefficients equal to 1 and 2 respectively. Linearity results in superposition of the inputs F and G both in the spatial and in the frequency domain. Linear operators are additive: in fact, H is defined as a sum of two input signals and Hhat also corresponds to the sum of the individual responses of said signals.

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.

Convolution in the spatial domain is the same as multiplication in the Fourier domain. The results show that the (normalized) convolution of two Fourier transforms functions, is the same as multiplying the two original functions and plotting them in the Fourier domain.

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.

The different way the matrix is populated affects the Fourier representation of F, which becomes asymmetrical (in the opposite sense with respect to the spatial domain). Having a horizontally rectangular shaped non zero domain results in greater frequency variations (in the greyscale sense) in a vertical direction, which explains why we have different phase and spectrum variations in the second Fourier transform when compared to the first. Scaling in the spatial domain therefore clearly affects the Fourier transform too, resulting in a compressed and directional output.

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

The image is the exact translation in the Fourier domain of Fhat (a rotation in the spatial domain corresponds to a rotation in the Fourier domain). Rotations of 45° or 90° do not affect the frequencies of the spectrum at all, as opposed to the 30° and 60° which lead to a wavy distortion in the frequency domain which can still be spotted after rotating back the image to the original orientation. The 45° and 90° cases completely lack the noisy disturbances in the frequency domain, which appear in the 30° and 60° situations as black, wavy spots.

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

The magnitude of the Fourier transform of an image carries information about the gray scale intensity of the image, location by location. Mathematically, the magnitude of a signal equates to the length of the complex vector in the frequency domain: the higher the magnitude, the greater the intensity. On the other hand, phase provides information on how the image looks, once it is represented. In other words, phase affects the final distribution and spatial organization of the image's features. If magnitude essentially impacts the color intensity, phase directly impacts the visual disposition of these colors. Indeed, mathematically the phase represents the angle which the complex vector associated with a specific frequency of the signal forms with the positive x-semi axis. This intuitively explains why changing the phase of the images results in a chaotic cloud of points.

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?

(Values in the notebook). We can see that the variance-covariance matrix is approximable to the identity matrix multiplied by the variance of the gaussian t. Indeed, the variances on the diagonal are all equal to the variance t of the gaussian kernel and the covariances (outside the diagonal) are all equal to 0.

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.

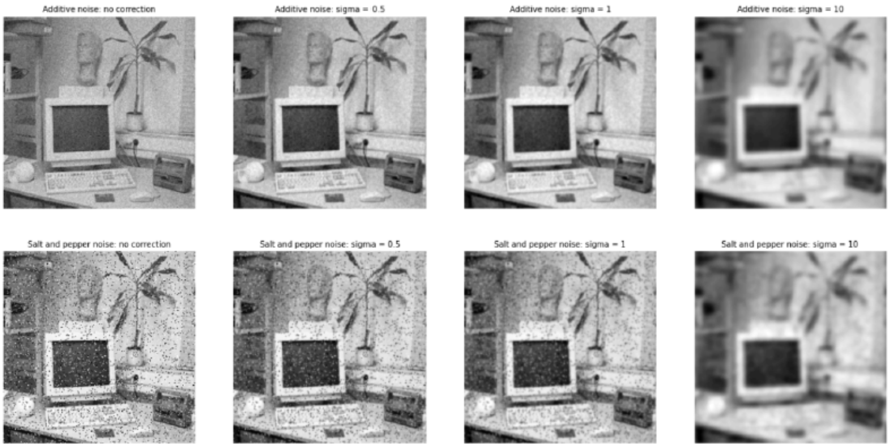
The results are approximable to the ideal continuous case. The parameter σ in the Fourier domain measures how much impact the filter has on the pixels of an image and it does so by influencing the values of the kernel. If the variance is low in the spatial domain, then the values in the kernel will decay very fast going from the center values of the matrix to the outer rows and columns, because the interval is smaller. On the other hand, for high variances, the interval will be wider and the values of the kernel will decay slower, meaning that the high frequencies will tend to be cut off in favor of the lower frequencies. In the spatial domain 2 is a division factor, whereas in the Fourier domain is a multiplication factor. This means that if the variance has a high value in the spatial domain, then in the Fourier domain it's very narrow. Indeed, the higher the variance in the spatial domain, the smaller in the frequency domain.

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?

The convolved images present a bigger point in the center as the variance gets bigger: this is because as the variance increases, high frequencies are filtered in favor of lower ones, which will cause a flattening of the gray levels of the images. With the higher variances, indeed, we can observe that the transition from white to black pixels is not as evident and direct as the transition in images which were convolved with gaussian with lower variances.

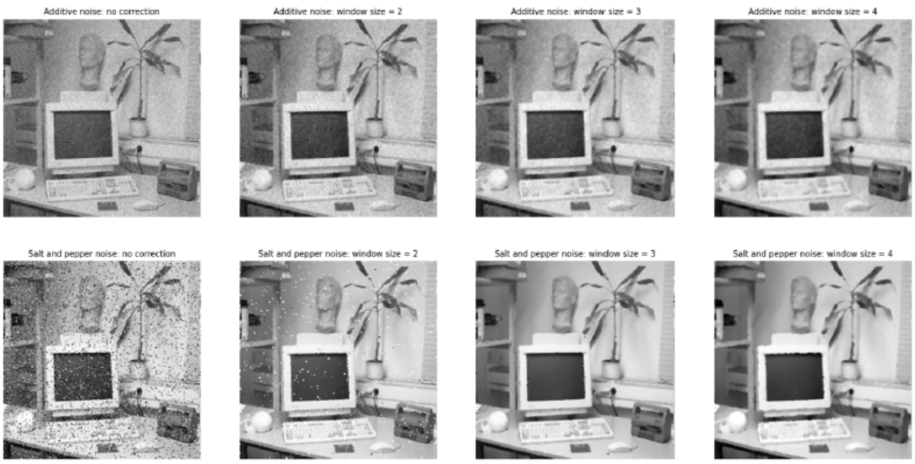
Question 17: What are the positive and negative effects for each type of filter? Describe what you observe and name the effects that you recognize. How do the results depend on the filter parameters? Illustrate with Python figure(s).

Gaussian filter:



The filter describes how much the gray-level value of a particular pixel spreads to its neighbors, so the further you are from a particular pixel, less information is spread from the center pixel to that pixel. For a high variance (10) the image is very blurred because every point of the image is affecting a large area of the image and since all pixels are affecting each other, then the image will be very blurred. For a low variance (0.5), the filter suppresses both the higher and lower frequencies less, hence there's more noise, but also more sharpness.

Median Filter:



We can observe the main properties of the filter: it doesn't blur the image, but it preserves the gradual changes in the gray levels the edges are preserved. For the salt and pepper noise, it creates a "painting" effect on the image, which increases as the window size of the filter increases. For example, with a window size of 4x4, this effect is too accentuated, whereas for a 3x3 window we can see that the noise is mostly corrected and this effect is not too strong.

Low-pass Ideal filter:

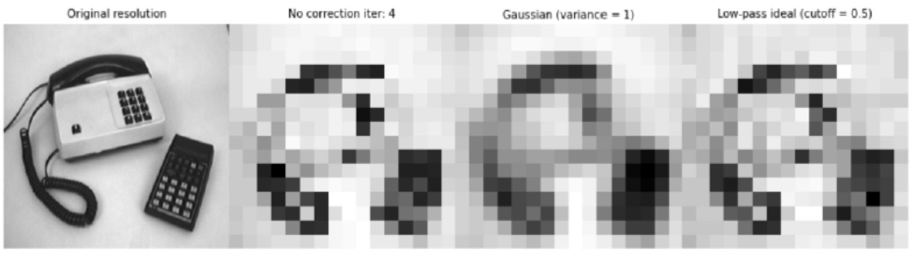
The ideal low-pass filter suppresses all the frequencies higher than the cutoff frequency. There's a main problem when we set the cutoff to a low frequency (0.15): we can see a waveform pattern on the image. This happens due to the shape of the low-pass filter: we're enhancing some low frequencies that are distorting the image. However, as we set a higher cutoff (0.5), these waveforms are pretty much unnoticeable.

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

We can see that the gaussian and median filters are more effective than the low-pass ideal filter in smoothing the image for both the types of noise. For the additive noise the gaussian filter provides the best results without blurring the image too much nor providing a "painting" effect. For the salt and pepper noise the median filter is without any doubt the best solution, even if it presents a "painting" effect.

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 $i = 4$.

Overall, with a low variance for the gaussian filter, the image is too under-corrected (not blended enough) and for high variances the correction is too evident (too much blurred). With a very low cut-off in the low-pass filter there are too many wave patterns in the final image, but if it is too high, then not enough high frequencies have been cut-off, so the correction is little to none.



For $i = 4$ we can observe that the Gaussian filter is much smoother in maintaining the general shape of the image, whereas the low-pass filter produces a very much fragmented image and it could be a better choice with noisier pictures (with higher frequencies).

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.

Overall, smoothing is necessary with subsampling because by downgrading the number of pixels of an image, it creates high differences among the remaining pixels, hence the characteristics of the image tend to become less clear as the resolution of the image decreases. This resonates with the wave sampling technique that imposes a minimum sampling frequency. However, for i = 4 the image has such low resolution that the smoothing is not enough to make the objects recognizable. The higher the variance in the gaussian filter, the more high frequencies are penalized, and the image appears more towards gray (similarly to having a low cut-off in the low-pass ideal filter). The smaller the variance, the narrower is the gaussian bell, so high frequencies are not going to be penalized (similarly to having a high cut-off).