Exercise Topic 2

Introduction to digital image processing – Part 2

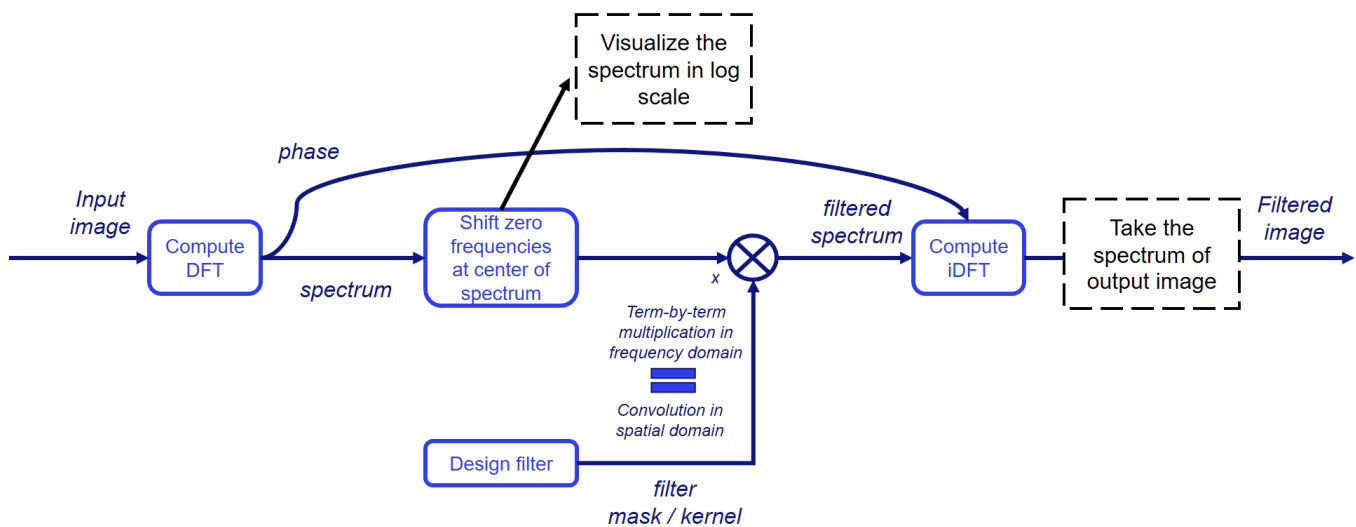### Part 1 - Linear spatial filtering *(see useful functions below)*

1. Blur (on the image test_pattern.tif and another)
    a. Blur the image with a boxfilter of different sizes (e.g. 3x3, 5x5, 11x11, …): create the filter mask yourself and apply it with imfilter
    b. Visualize the corresponding histograms
    c. Do a. and b. with a Gaussian filter
    d. Apply a threshold the Dumbbell Nebula image. Blur it first and apply the same threshold again.

1. Image enhancement (to do on 2 images)
    a. Blur slightly the original image, then compute the laplacian of the result (using fspecial) and subtract the result to the blurred image. Compare with the blurred version
    b. Perform unsharp masking and highboosting of your image

Part 2 - Filtering in the frequency domain *(see useful functions below)*

Using zero-phase shift filters in the frequency domain, do the following:

1.   For ThumbPrint, remove the smudge (smooth variation of intensity in the image) of the fingerprint and threshold the result

2.   For Florida, remove the noise (horizontal bands)

3.   For MountainHalftone, remove as possible the dot effect

In each case, start by computing the fft of the image and visualizing the log of its spectrum (remember to center the frequencies with fftshift and add 1 to the spectrum). You can then design your filter. Apply the filter on the spectrum and reconstruct with the original phase. The whole process is illustrated in the figure below.

Related useful functions

Purpose	Matlab	Python
Load an image	`imread`	`img=Image.open('imagepath')` `np.asarray(img)` with Pillow and NumPy
Display an image	`imshow` (be aware that the data type sets the range of expected values → use [low high] range as parameter or [] to scale between min and max), `imagesc`	`plt.imshow`(img) from Matplotlib
Spatial filtering	fspecial to create filter masks (incl.laplacian), imfilter to apply filter imgaussfilt for Gaussian filter	`cv.Laplacian` to calculate laplacian of an image `cv2.filter2D()` to apply filter `cv2.GaussianBlur()` for Gaussian filter
plot histograms	`imhist`	matplotlib.pyplot.hist matplotlib.pyplot.bar
DFT	fft2 and ifft2	Cv2.dft and cv2.idft
Shift spectrum so null frequencies are in middle	fftshift	np.fft.fftshift