RRY025-- Image Processing

Exercises 3 – Lecture 3 – Fourier I

EX 1. Visualize the 1D Fourier transform.

Open the website: http://nain.oso.chalmers.se:8501

The page shows an applet demonstrating Fourier transforms. The top left quadrant shows the input function and the bottom right shows the real part of its Fourier transform. The other quadrants show the kernel function and the kernel multiplied by the input function. Inspect especially the top left, top right, and bottom left frames, and relate them to the theory of the 1D Fourier transform.

Do you understand how the real part of the Fourier transform (bottom right) relates to the other frames? Try to figure it out. Use the 'select u-frequency' slider on the left-side panel to change the frequency that is being inspected.

Question: Choose values of u of 1, 2, 3 Hz. Why is the real part of the Fourier transform zero at these frequencies?

ANSWER: By definition, the real part of the Fourier transform is the integral over f(t)cos(…). In other words, the area under the curve defined by f(t)cos(…). This is shown in bottom left. You can see how the area under the curve is zero at these u-coordinate locations.

Question: Why does the Fourier transform have maxima and minima at u = 1.5, 2.5, 3.5 Hz?. Why does the amplitude of these local maxima and minima decrease as 1/u?

ANSWER: Same as above, this can be understood by looking at the area under curve in the bottor left frame.

Question: Do you understand why the imaginary part of the tophat function is zero? Change the function f(t) from 'tophat' to 'ramp'. Do you understand why now the real part of the Fourier transform is zero and imaginary part is not?

ANSWER: Imaginary part comes from the area under sine curve. Sine curve is an even function, and hence, the area under it is always zero. (area between the curve and the x axis.)

Additional insight: the difference between the Fourier transforms of even and odd functions.

EX 2. Basics of 2D Fourier transform basics in Matlab

Construct a 255 x 255 pixel sized image (*I*) that contains a centered, 2D circular tophat function. Construct it so that you can set the radius of the tophat function as an input parameter, *tophat_radius*.

Hint: The function bwdist() in matlab is likely helpful in constructing the tophat function.

Take the Fourier transform *F*(*I*) and inspect the amplitude image of *F*(*I*). How can you get it to be well visible (and why)? Display and visually inspect also the phase image of *F*(*I*).

Experiment with fftshift() and ifftshift(), and make sure you understand what is meant by "centering" of the Fourier transform.

How do you relate what you see to Exercise 1?

Change the *tophat_radius* to be larger and smaller, and study how those changes affect the amplitude image of F(I). Can you describe *why* the amplitude image changes the way it does?

ANSWER: One example given as an .mlx file.

Usually, one has to view the magnitude with a logarithmic color scaling to see anything in it. This is because the low-frequencies are usually much stronger (=brighter) than the higher frequencies.

As in Exercise 1, one sees the strong contribution of low frequencies (in any real image). The image is now analogous to f(t), only in 2D.

The tophat_radius parameter should anti-correlate with the size of the features seen in the amplitude image of F(I). This is because the frequency is proportional to 1/wavelength. Wavelength here indicates a size-scale.

EX 3. … continued…

Repeat EX 2, but for a tophat function that is box-shaped (instead of circular).

Make the tophat function rectangular, instead of a box.

Can you develop an understanding of how the frequency domain image (amplitude) responses to the changes in the *I*?

Finally, create a smoother version of the tophat function by convolving it with some Gaussian kernel. Study the effect of smoothing to the amplitude image of *F*(*I*). Any insight?

ANSWER: One example given as an .mlx file. One should see the boxy shape being repeated in the Fourier domain.

Smoothing should also smooth the Fourier space image. The insight: Smoothing greatly reduces the "ringing" in the Fourier domain image. In case of real images, strong features and large gradients cause ringing in Fourier space. Compare with the Fourier transform of the ideal tophat filter…

Insight from the inverse of the above: If one somehow reduces ringing in Fourier space, image features should become smoother, right?

EX 4. Continuing 2D Fourier basics…

Load the cameraman.tif image and display it (the image is a default Matlab stock image).

Take the 2D DFT of the image. Display the amplitude and phase. How can you get them to be well visible (and why)?

What features in the amplitude and phase images can you relate to the properties of the original image, if any?

Reconstruct the original image using the Fourier transformed image. Is all well?

Reconstruct the image using only the amplitude information of the Fourier transform. Then do the same using only the phase information. Which component seems to be more important in capturing human-recognizable features of the image?

ANSWER: One example given as an .mlx file. The goal here is to operate for the first time on a real image and get an idea of what the Fourier transform of it looks like.

One should find out that there is something to be seen in the magnitude image. For example, strong lines that indeed arise from the strong lines in the image. The phase information is not necessarily human-recognizable at all.

Overall, Fourier transforms do not usually contain sensible information for a human eye.

Reconstruction using magnitude and phase images: Using only magnitude does not lead to intelligible image. Using only phase, though, clearly shows some recognizable features. Insight: Phase is important for localizing features in the data.