LAB-07: Frequency Domain and Image Analysis in Frequency Domain

Objective:

The objective of this lab is to understand Fourier Transform, apply it on images and understand the results.

Theory:

Fourier Series tells us that any function can be represented as a sum of sines/cosines of different frequencies multiplied by a different coefficient. Similarly, non periodic functions can also be represented as the integral of sines/cosines multiplied by weighing function.
The Fourier transform of f(x) is given by,

$$\Im\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\exp[-j2\pi ux]dx$$

Fourier transform pair for a function f(x,y) of two variables;

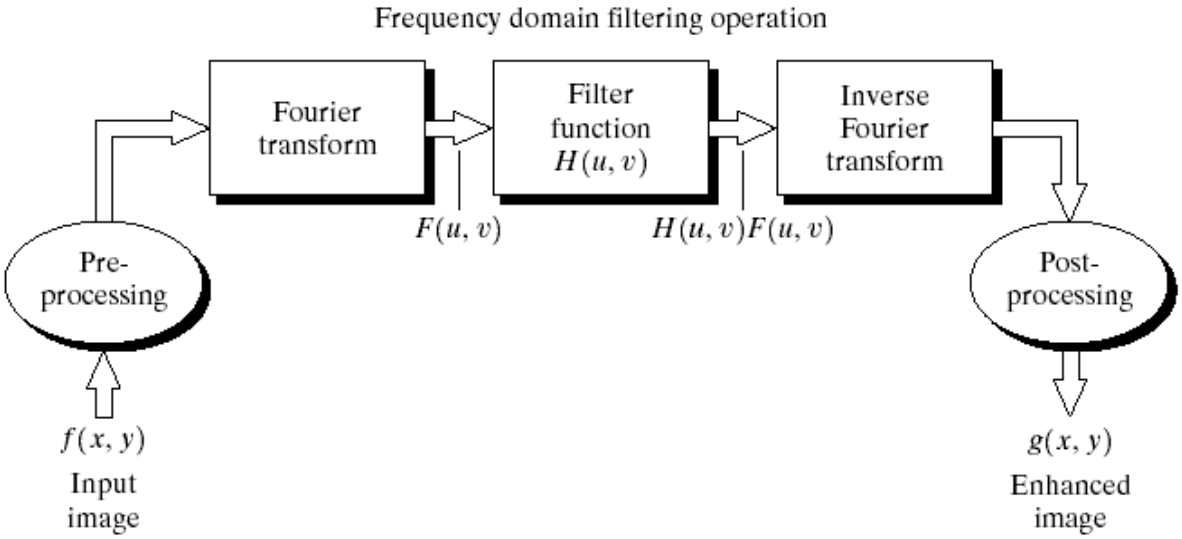
$$\Im\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp[-j2\pi(ux+vy)]dxdy$$

$$\Im^{-1}\{F(u,v)\} = f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\exp[j2\pi(ux+vy)]dudv$$

The filtering in frequency domain consists of following steps:

1. Compute F(u,v) the DFT of the image
2. Multiply F(u,v) by a filter function H(u,v)

Compute the inverse DFT of the result



Some Useful Commands:

1. To obtain the Fourier Transform of an image: my_transformed_image = **numpy.fft.fft2(my_image)**
2. To obtain the Inverse Fourier Transform of an image: my_inverse_image = **numpy.ifft2(my_image)**
3. To shift the DC component of a Fourier Transformed Image to center: my_shifted_image = **numpy.fft.fftshift(my_transformed_image)**
4. To shift the DC component back to the top left corner: my_inverse_shifted_image = **np.fft.ifftshift(my_shifted_image)**
5. To calculate absolute of a value: my_absolute = **numpy.abs(my_image)**
6. To calculate the exponential of an element: my_exponential = **numpy.exp(my_input)**
7. To use the value of pi: **numpy.pi**
8. To denote a complex number: **simply put j after it** e.g. **-1j**
9. To obtain the dot product of two arrays: my_dot_product = **numpy.dot(my_array1, my_array2)**
10. To normalize an image: my_image_normalized = **cv2.normalize(my_source_image, my_desitnation_image, new_min_value, new_max_value, NORM_MINMAX, CV_8UC1)**

    The argument NORM_MINMAX tells the function which method to use to normalize the image while the argument CV_8UC1 tells the number of channels that the destination image will have. In other words, CV_8UC1 means a grayscale image in unit8.

Lab Tasks:

Lab Task 1:

The discrete Fourier Transform of an image can be calculated as follows:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

Use the image Fig01.tif to compute the DFT.

The following transformation can be applied to the images (so that the Fourier transform the spectrum will be shifted at the R/2, C/2 location).

F(x, y) = F(x, y)*(-1) ^ (x + y)

Now, repeat the above using the built in functions and display the results of both.

Lab Task 2:

Create an ideal low pass and high pass filter using the distance map. The size of the distance map should be equal to the size of the image. Take DFT of the Fig01.tif and dot multiply the transformed image and the filter. Take inverse DFT of the image and display the results.

Conclusion:

This lab has given an introduction of Fourier transform for image processing and application of different filters for image enhancement.