Experiment 4

Frequency Filtering

Image filtering is useful for many applications, including smoothing, sharpening, removing noise, and edge detection. A filter is defined by a kernel, which is a small array applied to each pixel and its neighbors within an image. In most applications, the centre of the kernel is aligned with the current pixel, and is a square with an odd number (3, 5, 7, etc.) of elements in each dimension. The process used to apply filters to an image is known as *convolution*, and may be applied in either the spatial or frequency domain.

Within the frequency domain, convolution can be performed by multiplying the FFT (Fast Fourier Transform) of the image by the FFT of the kernel, and then transforming back into the spatial domain. The kernel is padded with zero values to make it of the same size as the image before the forward FFT is applied. Other method could be that filters are usually specified within the frequency domain itself and then it is straightaway multiplied with image pixels element by element and the IFFT is found to get the original image.

Filters are classified into low-pass and high-pass depending on their filtering characteristics. Some common types of filters used in image processing are *Ideal*, *Gaussian*, *Butterworth*, etc. A DFT decomposes a sequence of values into components of different frequencies. A fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and it's inverse.

Problem Objective

Write C++/Image-J modular functions to perform the following operations on the 512 × 512 grayscale test images, e.g. lena_gray_512.jpg, jetplane.jpg, lake.jpg, livingroom.jpg, mandril_gray.jpg, pirate.jpg, walkbridge.jpg.

1. **FFT2** (takes input image filename as the argument; gives 2D FFT coefficients as output)
2. **IFFT2** (takes 2D FFT coefficients as input argument; gives the back-projected/ reconstructed image as output)
3. Perform *Ideal*, *Gaussian*, and *Butterworth* low-pass and high-pass filtering, taking cut-off frequency, D_0, and image filename as input arguments) respectively with

Ideal_LPF	**Ideal_HPF**
Gaussian_LPF	**Gaussian_HPF**
Butterworth_LPF	**Butterworth_HPF**

Display the (shifted) magnitude spectrums of the input, the filter and the filtered output. You may make use of the tracker/slider function to choose images, filter types and cut-off frequencies.

Note

1. Do not hardcode the filenames and/or image size/any user arguments into the code.
2. Use proper code commenting, documentation, and self explanatory identifiers for variables/functions etc.

References

R. C. Gonzalez and R. Woods, *Digital Image Processing*, Reading, MA: Addison-Wesley, 1992.