MLSP – Lab 3
September 18th, 2018

Exercise 1: Gaussian Smoothing

To perform blurring/remove detail and noise from the images we will use the Gaussian smoothing operator through a 2D convolution operation. The isotropic (circularly symmetric) Gaussian is defined as follows:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The Gaussian kernel is ‘bell-shaped’ and the distribution is shown in Figure 1.

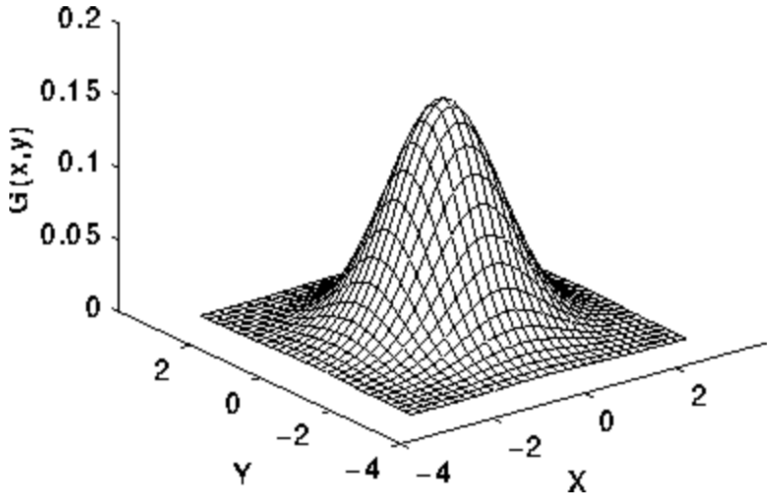

Figure 1. 2-D Gaussian distribution with mean (0,0) and $\sigma = 1$

Smoothing filters (like the Gaussian) act as *lowpass frequency filter* as they attenuate the high frequency components(noise) in the signal. Figure 2 show the frequency response of a Gaussian filter. Frequency response can be observed by taking the *Fourier transform of the filter*

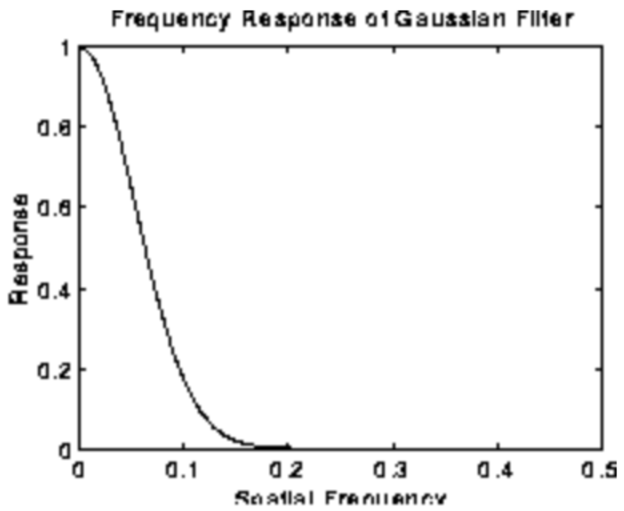

Figure 2. Frequency Response of a Gaussian filter ($\sigma = 3$)

Steps:

1. Illustrate the effect of smoothing with
   a. Load image "Lena_noisy"
   b. Median filter with kernel size of 5x5 (medfilt2)*
   c. Gaussian ($\sigma = 1$) with kernel size 5x5 (imgaussfilt)
   d. Compare results between mean and Gaussian filter
   e. Gaussian ($\sigma = 2$) with kernel size 9x9
   f. Gaussian ($\sigma = 2$) with kernel size 15x15
   g. How does σ and kernel size of Gaussian affect the output images?

2. Frequency analysis
   a. FFT and Histogram
      i. Load image named "Lena"
      ii. Take the FFT of the Lenna image**
      iii. Perform Gaussian filtering of "Lena". Take FFT of filtered image
      iv. Perform Median filtering of "Lena". Take FFT of filtered image
      v. Compare the FFT obtained in parts ii,iii.iv.
      vi. Plot a histogram of the images – Lena, Gaussian and Mean filtered images. What information does the histogram provide as compared to the FFT?
   b. FFT Algebra
      i. Load images "stp1" and "stp2"
      ii. Take FFT of "stp1" and "stp2". Compare and explain
      iii. Add them using blend
      iv. Take the inverse FT of the blend. Explain the result
      v. Multiply the images "stp1" and "stp2"
      vi. Take the FFT of the product
      vii. Multiply the FT of images obtained in step ii. Compare the results

* Create the filter based on the type and kernel size and perform a 2D convolution of the filter on the image

**Steps to take plot FFT of a 2D image

```
      F= fft2(Image);
      F = fftshift(F); % Center FFT

      F = abs(F); % Get the magnitude
      F = log(F+eps); % Use log, for perceptual scaling, and +eps since
      log(0) is undefined
      F = mat2gray(F); % Use mat2gray to scale the image between 0 and 1

imshow(F,[]); % Display the result
```

Exercise 2: Image compression with Discrete Cosine Transform (DCT)

In this exercise, we will see how can we reduce the amount of data required to store an image using DCT.
Code the following steps in Matlab:

1. Read the image 'Cameraman.tif' and show it on screen
2. Calculate the DCT coefficients (`w`) of the image using matlab function dct2.
3. Calculate and plot the energy of the coefficients:
    a. `Energy = log(abs(w));`

4. Now, create a new coefficient matrix `w2` with the same size and values of `w`. Set to 0 all the values of `w2` beyond the row=`nPrinipal` and column=`nPrincipal` (we are eliminating this information and keeping only a sub-matrix of `w`).
    a. `nPrincipal = 128;`

5. Plot the energy of `w2`
6. Show the inverse DCT of `w` (reconstruction of the original image);
7. Show the inverse DCT of `w2` (reconstruction of the compressed image);