Image Enhancement in the Frequency Domain

Frequency Domain Filters are often used for the smoothing or the sharpening of an image by removal of high or low frequency components. Frequency domain filters are different from spatial domain filters as they basically focus on the frequency components of the image. The reason for doing the filtering in the frequency domain is generally because it is computationally faster to perform two 2D Fourier transforms and a filter multiply than to perform a convolution in the image (spatial) domain. This is particularly so as the filter size increases.

A. As elements of F(u,v) are complex numbers, we cannot view them directly, but we can view their magnitude |F(u,v)|. The display of the magnitude of a Fourier transform is called the spectrum of the transform. The 2D FT and its inverse are implemented in MATLAB by functions fft2 and ifft2, respectively.
B. The value F (0,0) of the Fourier transform is called the DC coefficient which equals to the sum of all terms in the original matrix. For purposes of display, it is convenient to have the DC coefficient in the center of the matrix by applying the function fftshift in MATLAB (the DC coefficient will have the highest value in the Fourier transform).
C. One trouble is that the DC coefficient is generally very much larger than all other values. This has the effect of showing a transform as a single white dot surrounded by black. One way of stretching out the values is to take the logarithm of |F(u,v)| and to display log(1+|F(u,v)|).
D. The following steps are required for computing and visualizing the FT of an image using MATLAB: read the image, change it to double, get the FT of the image (fft2), get the magnitude(amplitude) of the image by getting the absolute value of the FT (this is called the spectrum of the FT and it is obtained using abs function), shift the DC coefficient to the center of the image for better visualization (fftshift), apply the log transformation on the shifted FT to compress the dynamic range and be able to view all the details.
E. There are two options for designing and implementing image filters in the frequency domain using MATLAB:
    a. Obtain the frequency response function from the spatial domain filter. The Image Processing Toolbox has a function called freqz2 that does exactly that. Make sure you pad the original image with zeros before applying FT in order to avoid wraparound error (Use the provided function paddedsize).
    b. Generate filters directly in the frequency domain.

Dr. Omar Falou

What to do for this lab:

1. <u>Computing and Visualizing the Discrete 2D Fourier Transform:</u>
   a. Load the cameraman image ('cameraman.tif'), convert it to double (one of the data classes accepted as an input to fft2), and generate its Fourier Transform (FT).
   b. Shift the FT array of results.
   c. Calculate the absolute value of the FT. Why did we use the absolute value?
   d. Display the FT results, remapped to a grayscale range with the title 'Direct Remap'.
   e. As you may have noticed, directly remapping the image to the grayscale range does not give us any useful information (only a white pixel at the center of the image—the DC component of the FT results—with all other pixels displayed as black). This suggests that there might be a significant amount of detail in other frequencies that we just cannot see. Images with similar attributes (very large dynamic range of gray values) can be brought up by remapping the image with a log transformation. Apply the log transformation to compress the dynamic range and call it 'Log Remap')
   f. How does the log remap compare to the direct remap? Hint: use subplot to display both images.
2. <u>Obtaining Frequency Response of a Spatial Domain Filter:</u>
   a. Generate the frequency response of the 'average' spatial domain filter. Apply the generated response on the original cameraman image. Display the cameraman image before and after filtering.
3. <u>Generating Filters Directly in the Frequency Domain:</u>
   a. Load 'cameraman.tif' image and calculate its Fourier Transform.
   b. Generate a distance matrix having a size equals to that of the cameraman image (use the distmatrix function provided).
   c. Create an Ideal Low Pass filter by first creating a matrix of zeros (whose size equals to that of the image) and then by using the distance matrix to assign a value of '1' to radii below the cut-off frequency (assumed to be equal to 25)
   d. Display the filter's frequency response.
   e. Apply the filter to the FT of the image and display the filtered image in the spatial domain along with the original image.
   f. How does the filtered image compare to the original image? Do you notice any artifacts?

Dr. Omar Falou