EE435: Biometric Signal Processing

Project 4: Frequency Domain Processing

Assigned: Tues 2/8/11 Due: Tues 2/15/11

I. Frequency Filters: Low Pass Filters (LPF)

A Gaussian LPF can be created according to the equation:

$$H(u,v) = e^{-D^2(u,v)/2D_0^2},$$

where for each pixel location in the filter (at location (u,v)), $D(u,v)$ is its distance from the center of the image and D_0 is the "cutoff distance" (that is, the pixel distance from the center of the filter where the filter coefficient has fallen to 0.607 of its peak value).

An ideal LPF with cutoff distance D_0 from the origin is defined as:

$$H(u,v) = \begin{cases} 1, & D(u,v) \le D_0 \\ 0, & D(u,v) > D_0 \end{cases}$$

1. Write two MATLAB functions, one that creates a Gaussian LPF and another that creates an ideal LPF.

   Usage: ***y=glpf(rows,cols,cutoff);***

   Input: The number of rows and columns of the desired filter, and its cutoff distance.
   Output: The Gaussian low pass filter.

   y=ilpf(rows,cols,cutoff);

   Input: The number of rows and columns of the desired filter, and its cutoff distance.
   Output: The ideal low pass filter.

2. Download the "saddam_bw.png" image. Low pass frequency filter this image with a Gaussian LPF having cutoff distances of 10, 50 and 100. Present the original and these filtered images in a 2x2 subplot and discuss the effects of increasing the cutoff distance on the filtered image.

3. Next, low pass filter this image with the ideal LPFs having cutoff distances of 10, 50 and 100. Present the original and these three filtered images in a 2x2 subplot, and then compare the Gaussian LPFs results to the ideal LPFs results.

II. Frequency Filters: High Pass Filters (HPF)

A 2D high pass filter can be created from the coefficients of a low pass filter:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v).$$

Here, the low pass filter is created, then each of its coefficients is subtracted from a value of 1.0. So a MATLAB function to create a high pass filter can use the function that creates a low pass filter.

1. Write two MATLAB functions, one that creates a Gaussian HPF and another that creates an ideal HPF.

   Usage: ***y=ghpf(rows,cols,cutoff);***

   Input: The number of rows and columns of the desired filter, and its cutoff distance.
   Output: The Gaussian high pass filter.

   y=ihpf(rows,cols,cutoff);

Input: The number of rows and columns of the desired filter, and its cutoff distance.
Output: The ideal high pass filter.

2. Download the "blackwatch_bw.jpg" image. Display it, and then filter it with a Gaussian HPF. Adjust the cutoff distance until you feel the edges stand out well (this may be a large cutoff distance, but experiment with different values…try distance = 100, 200, 300, etc., and see what happens). Based on the statistics of the filtered result (to help you choose a threshold), threshold the high pass filtered image to create a black and white image that shows off the edges. Display 3 images in a 1x3 subplot: (1) the original image, (2) the HPF filtered image, and (3) the thresholded result. Ensure the title of the 2nd subplot includes the cutoff distance of the filter and the title of the 3rd subplot includes your threshold value. Printout and turn in this figure.

III. Frequency Filters: Notch Filters/Band Pass Filters

If an image is corrupted by periodic noise, a frequency filter can be used to remove or at least reduce it. The general procedure to remove periodic noise is to:

-compute and display the 2D FFT of the noisy image, and look for signs of periodic noise (typically these are bright spots that appear at regular intervals somewhere in the FFT). Be sure to view the FFT using some of the methods described in the frequency filtering slides handout.

-determine the characteristics of a filter that would remove the noise: either the inner and outer radius of a circular notch filter, possibly a low pass filter, or perhaps a region of the FFT that could be zeroed out with a rectangular shape for example.

-create the filter, then do frequency filtering as you did in parts I and II above.

.

Download the "reagan_noisy.png" image, which is corrupted by some periodic noise. Investigate its frequency spectrum, then create an ideal notch frequency filter that can be used to reduce or eliminate the periodic noise. Note: a notch filter can be created from an ideal LPF added to an ideal HPF. Perform the frequency filtering, then turn in figures of: (1) your filter, (2) the original noisy image and (3) your "improved" image.

In the report for this project… include your code and any images that you created. Answer the questions asked.