

CS474/674 Image Processing and Interpretation

Fall 2011 – Dr. George Bebis

Programming Assignment 4

Due Date: 11/29/2011

In this assignment, you will experiment with filtering in the frequency domain; the steps have been discussed in class extensively and are outlined on page 263 in your textbook.

Experiment 1 (noise removal): In this experiment, you will consider the effects of additive noise and the use of DFT to remove this kind of noise. The noisy image shown below (i.e., available from the course's webpage) has been generated by adding noise in the form of a cosine function. If we denote the original image as $f(x,y)$, the noisy image can be denoted as $f(x,y)+n(x,y)$ where $n(x,y)$ is a 2-D cosine function. Using frequency domain filtering, devise a procedure for removing the noise and show your results (note: a detailed description of your method should be provided in your report). For comparison purposes, try to remove the noise in the spatial domain by convolving the noisy image with a Gaussian filter (e.g., 3 x 3 and 5 x 5). Compare your results using Gaussian filtering with those obtained using frequency domain filtering.



Experiment 2 (convolution in the frequency domain): Perform the experiment described in Example 4.15 (page 267), using the “lenna” image. For comparison purposes, you should perform the convolution both in the spatial and frequency domains (i.e., see Figure 4.39(c,d), page 268).

Experiment 3 (image restoration/motion blur): In this experiment, you will experiment with (i) Inverse filtering and (ii) Wiener filtering.

$$\hat{F}(u,v) = (1/H(u,v))G(u,v) \qquad \hat{F}(u,v) = \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \cdot \frac{G(u,v)}{H(u,v)}$$

Assume that $G(u,v)=H(u,v)F(u,v)$ where $H(u, v) = \frac{T}{\pi(ua + vb)} \sin(\pi(ua + vb)) e^{-j\pi(ua+vb)}$

(i.e., motion blur). First, apply motion blur on the “lenna” image (i.e., obtain $G(u,v)$) using $a=b=0.1$ and $T=1$ (i.e., an example is shown in Figure 5.26, page 350). Then, add Gaussian noise to the blurred “lenna” image. Note that adding Gaussian noise to an image is not the same as convolving the image with a Gaussian (i.e., this will actually reduce image noise). You should be using the Box-Muller transformation to generate Gaussian noise (see the links on the course’s webpage). To add Gaussian noise to an image, do the following for each pixel $I[i][j]$:

- (1) Call the function provided; suppose the value returned is $G(\mu, \sigma)$
- (2) $I_{\text{noisy}}[i][j] = I[i][j] + \mu + \sigma G(\mu, \sigma)$

Experiment with the following (μ, σ) values: (0, 10) (0, 100), and (0, 1000) (note: make sure that final pixel values are in [0 255]; use $\{I_{\text{noisy}}[i][j] - \text{min}/(\text{max}-\text{min})\} \times 255$ to rescale the values where min/max are the minimum/maximum values of the noisy image). For each blurred/noisy image, first experiment with image restoration using inverse filtering. Try different “radius” values and show your results in the spirit of Figure 5.27 (page 352). Then, experiment with image restoration using Wiener filtering. Try different “k” values and compare your results with Inverse filtering in the spirit of Figure 5.29 (page 356).

Experiment (Graduate Students Only): Implement the homomorphic filtering technique. Many times, images suffer from shading problems due to uneven illumination. The role of homomorphic filtering is to alleviate such problems. In your experiments, use the image shown below (i.e., can be downloaded from the course's webpage). As discussed in class, the main idea behind homomorphic filtering is to separate the illumination and reflectance components by applying the logarithmic function on the image. Then, you would need to apply an appropriate high-pass filter, which will emphasize high frequencies and attenuate lower ones, preserving fine detail at the same time.



Here are the main steps:

1. Apply the **In** function (natural logarithm) on the image.
2. Take the Fourier Transform of the image from step 1.
3. Apply high-pass filtering
4. Take the Inverse Fourier Transform of the image from step 3.
5. Apply the **exp** function (exponential function) on the image from step 4.
6. Display the magnitude of the image from step 5.

Note that in certain cases, the gray levels will be bigger than 255; applying a regular normalization and displaying the image will not give good results and it is better if you use a **log** function as we did with the spectrum of the Fourier Transform. The high-pass filter to be used in your experiments should be a Butterworth filter, which is characterized by the following transfer function:

$$H(u, v) = \gamma_L + \frac{\gamma_H}{1 + [D_0/D]^2}$$

where $D = \sqrt{(u - N/2)^2 + (v - N/2)^2}$, D_0 is the cutoff frequency of the filter and γ_L γ_H are the gains for the low and high frequencies correspondingly. Note that before you apply the filter on the image, you must first center it as in the case of the Fourier Transform spectrum. Experiment with different parameter values. As a starting point, choose $D_0=1.8$, $\gamma_L=0.5$ and $\gamma_H=1.5$. Then, keep the cutoff frequency the same and increase/decrease γ_L and γ_H . For example, assume combinations of γ_L and γ_H , with γ_L taking values from [0.2-0.8] and γ_H taking values from [1.2-1.8]. Print and comment your results. Which set of parameters seems to be working best? Is there a consistency in your results as γ_L increases/decreases? What about γ_H ?

Laboratory Write-up: For each programming assignment, you are to turn in a brief report (see instructions posted on the course's website). **The report is very important in determining your grade for the programming assignment.** Be well organized, type your reports, and include figure captions with a brief description for all the figures included in your report. Motivation and initiative are greatly encouraged and will earn extra points.