Digital Image Processing Laboratory: 2-D Random Processes

April 12, 2011

Introduction

This laboratory explores the use of 2-D random process models for images. You may implement your programs in Matlab. Make sure that all plots have accurate and clearly labeled axes, and have titles that indicate what is being plotted.

In some of the following exercises, you will be asked to display images in the 8-bit range of 0 to 255. To do this in Matlab, first set the colormap of the current figure to 256-level gray scale using the command

```
colormap(gray(256))
```

and then display the image x using the command

```
image(uint8(x)) .
```

To ensure the displayed aspect ratio is correct, you may also use the command `axis('image')`, or preferably `truesize` if the image processing toolbox is available.

If you are producing an electronic version of your report, it is usually best to export 8-bit images directly to a file using *imwrite*, as opposed to exporting through a figure window. For example, to export the image matrix x (with assumed range [0,255]), use the command

```
imwrite(uint8(x),'filename.tif') .
```

Other output, such as mesh plots, can be exported through the figure window menu.

Questions or comments concerning this laboratory should be directed to Prof. Charles A. Bouman, School of Electrical and Computer Engineering, Purdue University, West Lafayette IN 47907; (765) 494-0340; bouman@ecn.purdue.edu

1 Power Spectral Density of an Image

In this problem, you will use Matlab to read and analyze the gray scale image *img04g.tif*. If you are unfamiliar with Matlab, please refer to the Matlab tutorial information listed on the main web page for this laboratory.

1. Download the Matlab m-file *SpecAnal.m* and the gray scale image *img04g.tif*. The m-file estimates the power spectral density by computing the logarithm of the normalized energy spectrum over a 64×64 window of the image. The comment lines in *SpecAnal.m* explain how the m-file operates.

2. Run *SpecAnal.m*. The m-file will display the image *img04g.tif* and show a mesh plot of the estimated log power spectral density. Export the plot for your report.

3. Run *SpecAnal.m* for 128×128, and 256×256 block sizes. Notice the power spectrum estimates remain noisy even when the block size is increased. Export the two mesh plots for your report.

4. Write a Matlab function, *BetterSpecAnal(x)*, which computes a better estimate of the power spectral density of the 2-D array `x`. Your new m-file should:
   - Use 25 non-overlapping image windows of size 64×64. These windows should be selected from the center of `x`.
   - Multiply each 64×64 window by a 2-D separable Hamming window. You can create the 2-D Hamming window as the outer product of 1-D windows:
     `W=hamming(64)*hamming(64)';` .
   - Compute the squared DFT magnitude for each window.
   - Average this power spectral density across the 25 windows.
   - Display a mesh plot of the log of the estimated power spectral density.

5. Use *BetterSpecAnal(x)* to compute the power spectral density estimate of *img04g.tif*, and export the mesh plot for your report.

Section 1 Report:
Hand in:

1. The gray scale image *img04g.tif*.
2. The power spectral density plots for block sizes of 64×64, 128×128, and 256×256.
3. The improved power spectral density estimate.
4. Your code for *BetterSpecAnal.m*.

2 Power Spectral Density of a 2-D AR Process

In this problem, you will generate a synthetic 2-D autoregressive (AR) process using Matlab, and analyze its power spectral density.

1. Use the Matlab function *rand* to generate a 512×512 image, x, with independent random numbers each uniformly distributed on the interval $[-0.5, 0.5]$. Display the image `x_scaled=255*(x+0.5)` using the image command, as described in the introduction, and export the result for your report.

2. Filter the image x to produce the image y using an IIR filter with transfer function

$$H(z_1, z_2) = \frac{3}{1 - 0.99z_1^{-1} - 0.99z_2^{-1} + 0.9801z_1^{-1}z_2^{-1}} \ .$$

   Hint: Find the corresponding difference equation, and use that to implement the filter.

3. Display the image $y + 127$, and export the result for your report.

4. Theoretically calculate $S_y(e^{j\mu}, e^{j\nu})$, the power spectral density of y. Plot the magnitude of S_y using *mesh*, and export the result.

5. Use *BetterSpecAnal(y)*, your Matlab function from the previous exercise, to estimate the power spectral density of y. Plot the estimated power spectral density and export the result.

Section 2 Report:
Hand in:

1. The image $255 * (x + 0.5)$.
2. The image $y + 127$.
3. A mesh plot of the function $\log S_y(e^{j\mu}, e^{j\nu})$.
4. A mesh plot of the log of the estimated power spectral density of y using *Better-SpecAnal(y)*.