EXPERIMENT: TS

Academic responsible: Dr. Tania Stathaki (Room 812, Ext. 46229)
Equipment required: MATLAB Image Processing Toolbox

General Guidelines

- Standard images will be used in this experiment. You can find them in the MATLAB subdirectory **"toolbox\images\imdata"**.
- Some of these files are in **"tiff"** format and you may have to use the MATLAB **"imread"** function to retrieve them.
- You can find out how to use functions of the MATLAB Image Processing Toolbox via the **"help"** command or at:
  http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml

Part I: Image Transforms

1. Fast Fourier Transform (FFT)
   a. Display the log magnitude of the FFT of one of the standard images. Observe that the FFT coefficients with low index usually possess higher values compared to the coefficients with high index. Explain why.
   b. Create an artificial image of a grid of impulses and repeat the previous process described in 1a. Repeat also the same process with other regular periodic and non-periodic signals. Provide observations and possible explanations at each stage.
   c. Read two digital images A and B. Compute the phase of the FFT of image A and combine it with the amplitude of the FFT of image B to form a new image. Compute the inverse FFT of this new image and comment on the result.

2. Discrete Cosine Transform (DCT)
   a. Compute the Discrete Cosine Transform (DCT) of one of the standard images. It is suggested that you use the image **"autumn"**.
      The following program computes the DCT:
      ```
      X = imread('autumn.tif');
      I  = rgb2gray(X);
      J = dct2(I);
      colormap(jet(64)), imagesc(log(abs(J))), colorbar
      ```
      Note that most of the energy in the transform domain is concentrated within the upper left corner.
   b. Compare the results of the DCT with those of the FFT.

3. Hadamard Transform
   Write a MATLAB program that computes the Hadamard transform of an image. Comment on the advantages of Hadamard transform.

Part II: Image Enhancement

1. Use a standard image and equalise its histogram. The command **"histeq"** allows you to specify the number of bins. Try a wide range of bins and comment on the results obtained.

The commands below create two histograms of equalised images, one with 32 bins and one with 4 bins.

```
X = imread('cameraman.tif');
I  = rgb2gray(X);
J  = histeq(I, 32);
K = histeq(I,4);
subplot(2,2,1), imhist(J,32)
subplot(2,2,2), imshow(J)
subplot(2,2,3), imhist(K,32)
subplot(2,2,4), imshow(K)
```

2. Write a set of commands to carry a histogram modification of the intensity of a given image. You can assume that the histogram is given as (a) a data file (b) as a function.

3. The commands below generate the edges in the image **"trees"** based on different methods. Run the following code. Study and comment on the results.

```
X = imread('trees.tif');
I  = rgb2gray(X);
subplot(2,2,1), imshow(edge(I,'sobel'))
subplot(2,2,2), imshow(edge(I,'roberts'))
subplot(2,2,3), imshow(edge(I,'prewitt'))
subplot(2,2,4), imshow(edge(I,'Log'))
```

Write a short code to generate only those edges in an image inclined at +/- 45^o and test your program.

4. Load the image **"autumn"** and add noise of the so called **salt and pepper** type. Then with the command **"medfilt2"** carry out a median filtering on it. The following code carries out the task for a 3x3 neighbourhood. Try different neighbourhood dimensions and comment on the results.

```
X = imread('autumn.tif');
I  = rgb2gray(X);
J  = imnoise(I,'salt & pepper');
K = medfilt2(J,[3 3]);
subplot(1,2,1), imshow(J)
subplot(1,2,2), imshow(K)
```

Repeat with different noise characteristics.

Part III: Image Compression

Compute the DCT of an image. Set values less than 10 in the DCT matrix to zero, then display the image formed using the inverse DCT function. This can be done using the following program:

```
X = imread('autumn.tif');
I  = rgb2gray(X);
J  = dct2(I);
nz = find(abs(J)<10);
J(nz) = zeros(size(nz));
K = idct2(J)/255;
imshow(K), axis off
```

Repeat with different thresholds and comment on the results. Repeat with smaller blocks and different thresholds of the image and reassemble the inverse blocks. Explain the results.

Part IV: Design Exercise

Images of size approximately 128x128 pixels, are to be transmitted from a space probe. They are contaminated by noise which may be assumed to be Gaussian. The images are to be transmitted to earth on a frame-by-frame basis and in order to save bandwidth a compression of 4:1 is required. It is proposed to process a frame through DCT in blocks of 8x8 or 16x16. Carry out a study and recommend appropriate block sizes and thresholds for a visually acceptable reconstructed image. The study should be done both for the noiseless and for the noisy case.

Part V: Image Restoration

Create a program that restores a degraded image using:

1. The technique of Inverse Filtering.
2. The technique of Wiener Filtering. To calculate the power spectrum of the additive noise, assume that the additive noise is a white, Gaussian process. For the power spectrum of the original image, use the power spectrum of the degraded image as an estimate.

For each of the above cases use a 5x5 and a 7x7 Gaussian degradation with 20 dB noise.