Louisiana State University

Department of Electrical and Computer Engineering
EE 4780 – Introduction to Computer Vision
Spring 2007

Problem Set 1

Assigned: February 2, 2006
Due: February 12, 2006
What to Return: Create a folder and name it as your last name. Put the functions in the folder and email it to course TA, Stephen Bishop (sbisho4@lsu.edu), by the due date midnight.

Problem 1:
Write a MATLAB function (m-file) that calculates the mean absolute difference (MAD) and mean square error (MSE) between two grayscale images. The formulas of MAD and MSE are given below for *MxN* size images.

$$MAD(I,J) = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left|I(x,y) - J(x,y)\right|$$

$$MSE(I,J) = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I(x,y) - J(x,y)\right]^2$$

Your function should have the following header file:

```
function [mad, mse] = mad_mse (I, J)
%
% This function computes the MAD and MSE between two images.
% Input:
%     I, J: Input images. They are assumed to be grayscale (not color)
% Output:
%     mad: The mean absolute difference between the images.
%     mse: The mean square error between the images.
%
% Date: Put date here
% Author: Write your name here
%
```

Evaluation: The TA will run a script similar to the following:

```
a = imread('cameraman.tif');
b = imnoise(a,'gaussian',0,0.05);
[mad, mse] = mad_mse(a,b)
```

He should get what he expected from your function.

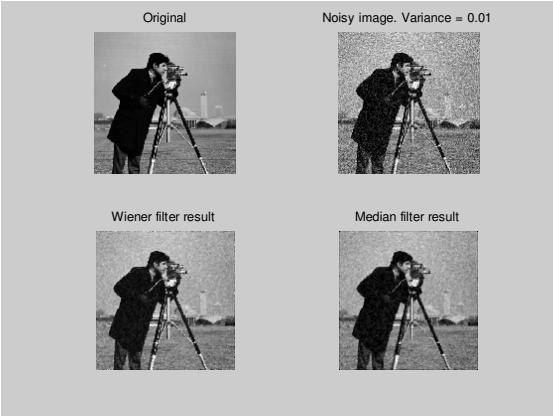
Problem 2:
In this exercise you will compare several image denoising algorithms. Read the *Cameraman* image (cameraman.tif) that is available in the MATLAB image processing toolbox. You may add random noise to it using the following lines.

```
% Read image
I = imread('cameraman.tif');
% Add Gaussian random noise with mean=0 and variance=0.01
J = imnoise(I,'gaussian',0,0.01); % You may try different noise types
figure; imshow(I,[]); title('Original');
figure; imshow(J,[]); title('Noisy');
% You may also use imview function to view images
```

(a) Create two noisy images. The first one is corrupted by "Gaussian" noise, with mean=0 and variance=0.005. The second image is corrupted by "salt & pepper" noise. (Use the default density given in `imnoise` function.) `>> help imnoise`

Use wiener filter (`wiener2`) and median filter (`medfilt2`) to denoise the images. Plot the results.

The results of an experiment should be given in a single figure. For example, the wiener filter results should look like this:



(b) Write a script file that will add Gaussian noise to the image, denoise it with wiener filter, and compute the mean absolute difference (MAD) and the mean square error (MSE) between the original and the denoised images. (Use the function you have written for Problem 1.) Change the noise variance from 0.0001 to 0.01. (You may take as many samples as you want.) Save the MAD and MSE result for each experiment. And then, plot the values as function of noise variance. Be sure to label your plot appropriately.

Evaluation: Write a script file that will produce all the results. Name the script file as problem2.m. The TA will run that script file. All the figures should be produced.

Problem 3:
Write a MATLAB function that applies histogram equalization on a specific region of an image. You may use the `histeq` function of MATLAB. However, you will specify the region on the image interactively. (You may use `imcrop` function to select a region on an image.) Name the function as my_histeq.

The header of the function is as follows:

```
function [ Region ] = my_histeq( I )
%
% This function takes a region of an image using imcrop,
%  applies histogram equalization to it using histeq,
%  returns and displays the result.
%
% I is the input image.
%
% Region is the histogram equalized region.
%
```

Evaluation: The TA will run a script file like this:

```
a = imread('cameraman.tif');
[ Region ] = my_histeq(a);
```

He should be able to select a region from the image. Once he selects a region, your function should apply histogram equalization to it and display the result.