CS540/440 – Digital Image Processing

Lab 3 – Filter Techniques for Image Enhancement, Edge Detection, and Noise Removal

Due: 16:00 p.m. Wednesday, Feb 21st, 2018

General Assignment Instructions: The same as Lab 2.

Note: If boundary extension is needed, please pad the boundary with 0's. Your functions should be general enough to accommodate any square filter with odd numbers of rows and columns.

1. [5 points]

Implement an **AverageFiltering** function to perform a filtering operation, as explained in class, on the input image. The prototype of this function should be:

function [filteredIm] = AverageFiltering (im, mask)

where im is the original grayscale image and mask is the square filter with odd numbers of rows and columns. Make sure that your function shows appropriate error messages when the mask's dimension is not an odd number, the mask is not a square, and the mask is not symmetric around the center. Note: both input and output images of the **AverageFiltering** function should be an array with data type uint8. You are not allowed to simply use the Matlab "filter2" or "conv2" or "fspecial" or "imfilter" or other built-in functions in your implementation.

Call **AverageFiltering** function to process the noisy image ***Circuit*** using a **weighted** 3×3 averaging filter and a **standard** 5×5 averaging filter, respectively. Display the original image and two processed images in figure 1 with appropriate titles.

2. [5 points]

Implement a **MedianFiltering** function to perform a filtering operation, as explained in class, on the input image. The prototype of this function should be:

function filteredIm = MedianFiltering(im, mask)

where im is the original grayscale image and mask is the square filter with odd numbers of rows and columns. Make sure that your function shows appropriate error messages when the mask's dimension is not an odd number and the mask is not a square. Note: both input and output images of the **MedianFiltering** function should be an array with data type uint8. You are not allowed to simply use the Matlab "medfilt2" or other built-in functions in your implementation. However, you can use the built-in sorting function to sort the values covered by the mask.

Call this function to process the same noisy image ***Circuit*** using a **weighted** 3×3 median filter **M** = [1 2 1 ; 2 4 2; 1 2 1] and a **standard** 5×5 median filter, respectively. Display the original image and two processed images in figure 2 with appropriate titles.

Note: The standard median filter explained in the class is a special kind of the weighted median filter. Each value in the median filter indicates the number of copies of the corresponding masked value in the original image involved in the standard median filtering.

Hint: use matlab built-in function median() to compute the median of an array.

3. [2 points]

Use the **strong** 3×3 Laplacian mask to filter the image ***Moon*** by calling an **appropriate Matlab built-in function**. Use the formula **Enhanced Image = Original Image - Filtered Image** to get the final enhanced image (Hint: This formula ensures that one of two strong Laplacian masks can be used). Use **imshow** to display four images including the original image, the filtered image, the scaled filtered image, and the enhanced image, in figure 3 with appropriate titles (Refer to Figure 3.40 in the textbook or the same figure on my notes).