SGN-12007 Introduction to Image and Video Processing
Exercise 5
13/11/2017-17/11/2017

The tasks should be completed and presented to TA during the lab session. Questions about exercises should be addressed to the TA personally or via email: (firstname.surname@tut.fi).

1. Laplacian filter with high-boost filtering

Create a Laplacian kernel of size 3x3 with the center values A={8, 9, 9.7}. Using the created kernels to enhance edge information in "cameraman.tif". Present your result in a 2x2 subplot. (Hint: imfilter)

2. Directional filtering

a. Load "cameraman.tif" and add random white noise with maximum intensity of 10. Show original and noisy image in the same figure.

b. Create a function called "directional_filtering" in matlab.
- The function takes two input arguments: a grayscale image and the size of the filter.
- Inside the function, creates 4 directional filters of the following degrees (0, 45, 90, 135). (Hint: a directional filter of size 3x3 at 0 degree is the following : [0,0,0 ; 1,1,1; 0,0,0]/3)
- Using the created filters to perform filtering on the input image.
- The function output 4 filtered images.

Using "directional_filtering.m" to filter the noisy version of "cameraman.tif". Try with kernel of size 3,5,7. Present your results in 3 different figures with each one contains 2x2 subplots of the filtered images.

c. How would you combine the results from 4 filtered images?

3. Threshold Median Filtering

a. Load "miranda1.tif" and add some white noise in the image center of size 100x100.

b. Implement a median filtering function called "med_filter.m" which takes 2 input arguments: an image and the size of the filter. The function outputs the filtered image.

Hint: given a filter of size 3x3 and an image patch [3, 4, 1; 0, 0, 1; 3, 20, 4], the filter will return value 3, which is the median value in the image patch.

Using "med_filter.m" to filter the noisy image in part a. Try with different filter size and present your results in the same figure.

c. Let I be the input and O be the output image of your "med_filter.m" function. Given a threshold "alpha", make the following changes in your "med_filter.m":

- if | I(x,y)-O(x,y)| > alpha, then O(x,y)=I(x,y). That is, we retain the original pixel intensity if the changes exceed a given threshold.

Adjust the value of threshold and observe the result with noisy "miranda.tif". Explain in what situation we would like to use a thresholded median filter?