

# COM-SGN.110 Introduction to Image and Video Processing

## EXERCISE 5

09.11.2020 - 11.11.2020

The tasks should be completed and presented to TA during the lab session. **Do not forget to upload your solutions to Moodle!** Questions about exercises should be addressed to the TA personally, through Moodle messages or via email, which can be found on the Moodle page of the course.

### 1. Laplacian filter with high-boost filtering

Create a Laplacian kernel of size 3x3 with the center values  $A = \{8, 9, 9.7\}$ . Use the created kernels to enhance edge information in “*cameraman.tif*”. Present your results in a 2x2 subplot.  
(Hint: `imfilter`)

### 2. Directional filtering

a. Load “*cameraman.tif*” and add random white noise with maximum intensity of 10. Display the original and the noisy images in the same figure.

b. Create a function called “*directional\_filtering*” in matlab.

- The function takes two input arguments: a grayscale image and the filter size.
- The function internally 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 :  $[0,0,0 ; 1,1,1; 0,0,0]/3$  )
- The function uses the created filters to perform filtering on the input image.
- The function outputs 4 filtered images.

Use “*directional\_filtering.m*” to filter the noisy version of “*cameraman.tif*”. Try kernels of the following sizes: 3, 5, 7. Present your results in 3 different figures with each one containing 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 area of size 100x100.

b. Implement a median filtering function called “*med\_filter.m*” which takes 2 input arguments: an image and the filter size. 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.

Use “*med\_filter.m*” to filter the noisy image from part a). Try with a 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. Implement another function, which differs from “*med\_filter.m*” as follows:

- if  $|I(x,y) - O(x,y)| > \alpha$ , then  $O(x,y) = I(x,y)$ . “alpha” is a threshold value which is given as an extra argument to the function. That is, during filtering we retain the original pixel intensity if the change exceeds a given threshold.

Adjust the value of threshold and observe the result with noisy “*miranda.tif*”. Explain: in what situation we would prefer to use a thresholded median filter?