IMPR 2017

Exercise #3

Due date: Dec. 21, 2017, 11:55pm.

This exercise can be submitted in pairs.

The goal of this exercise is to practice the basic concepts of using the Hough transform to detect parametric shapes on images and image denoising using the bilateral denoising filter.

You are not allowed to use opencv built in functions for the tasks in this exercise except reading images from disk. Only numpy functions are allowed.

Task 1: Hough transform to detect circles on images

You should write a function that get a binary image represent edge pixels only (for example a result of applying Canny edge detector on a grayscale image) and return all the circles (center (x,y) and radius) detected on the image as an Nx3 array where each row is x,y,r. In addition, the function should get as input an array represent the different radiuses to search circles at, and two threshold parameters: The first represent the minimal number of votes required to declare a circle and the second represents the minimal distance between the centers of two different circles.

You should implement the edge version of the circles Hough transform described at: https://en.wikipedia.org/wiki/Circle_Hough_Transform.

To clean-up your detected circles and return only local-maxima circles you may use the function ex3Utils.selectLocalMaxima provided in the ex3utils.py file.

Relevant numpy functions:

numpy.nonzero: https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.nonzero.html

numpy.argwhere: https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.argwhere.html

Task 2: Implement a bilateral denoising filter

a. Bilateral filter is a non-linear filter that involves assigning kernel according to the local intensity variation, hence make it edge-preserving. The Gaussian filter use Gaussian function on the neighborhood pixel location distance. Bilateral filter multiplies the Gaussian kernel with a second factor based on the neighborhood pixel intensity distance.

You should implement the bilateral filter function with an API as used in the driver file.

b. Run the bilateral filter with different parameter range for the spatial std and for the intensity std. Show the noise removed for the different settings of parameter by creating a matrix of images with two axes. one represents spatial std and one represents intensity std parameters of the filters. Each entry in the matrix should be an image with the removed noise (see the driver for basic example of obtaining the removed noise). Save the matrix as a pdf file.

Relevant python functions:

Printing your matplotlib figures as pdf file: https://stackoverflow.com/questions/21364405/saving-plots-to-pdf-files-using-matplotlib

Files included:

ex3driver.py: an example that defines and demonstrates the API for the functions you should implement.

ex3Utils.py: including helper functions you may use in your implementation.

ex3.pyc: binary version of school solution, so you can run and get an idea what your solution should look like.

./Images/* :set of images to be used with the driver

You should submit:

Ex3.zip file contains:

1. ex3.py: the code of your solution. The file ex3driver.py should run with your ex3.py without any error or warning.

2. ex3.pdf: a pdf file contains the figure you should generate in task 2.b

3. README: include your names

Submission instructions:

Through the course moodle website