**IMPR 2017**

**Exercise 1**

**Due date:** Nov. 23, 2017, 11:55pm.

This exercise can be submitted in pairs.

The goal of this exercise is to practice the basic concepts of image sampling and quantization and performing point and histogram-based operations on images.

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: Spatial sampling

1. Write a function that return an image sampled at the domain [0 1] x [0 1] from the analytic function:

f(x,y) = cos(K\*π\*(3X+2Y))

at a given pixel size. You can assume that pixel size is the same on both x and y axes.

Your function should get the value of K as a parameter. The default value for k is 2 (i.e. default value is used when no k argument is provided while calling to the function).

You can use the function numpy.meshgrid (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.meshgrid.html) to create your sampling coordinates

1. Use the provided driver to present your images at different resolution and with different values for K. Explain in the README file why changing the value of K cause different results to appear for the pixel size of 0.1.

Task 2: Quantization

Write function that performs Optimal Quantization on grayscale images. Given an image with gray-values in [0... 255] (8 bits per pixel), represent the image using only K different values (so that it can be represented now by log(k) bits per pixel).

The function should get as input the image and the value of k and return the quantized image.

Task 3: Image histograms

1. Write function that returns image histogram for a 256 grayscale images. The function doesn’t need to check whether the image is indeed a 256 grayscale image.
2. Write a function that get a grayscale image as input and return the same image after applying to it linear contrast enhancement.
3. Write a function that get a grayscale image as input and return the same image after applying a histogram equalization procedure.

Files included:

ex1Driver.py: an example that defines and demonstrates the API for the functions you should write

ex1.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:

Ex1.zip file contains:

1. ex1.py: the code of your solution. The file ex1driver.py should run with your ex1.py without any error or warning.

2. README: include your names, and the answer to question 1.b.

Submission instructions:

Through the course moodle website