IMPR 2017

Exercise #5

Due date: Jan. 25, 2018, 11:55pm.

This exercise can be submitted in pairs.

The goal of this exercise is to practice image pyramids representations and their application to image blending.

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.

Specifically, in this exercise you may use: scipy.signal.convolve2d (https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.convolve2d.html) to perform convolutions. However, a bonus of 10 points will be given to students who will use their own implementation for the convolution operation. Additional bonus of 10 points will be given for convolution implementation that will match or even beat the running time of the school solution.

Task 1: Gaussian pyramid

You should write a function that gets as input a grayscale image, the number of levels in the Gaussian pyramid, and the filter parameter (a) that will be used to reconstruct the filter according to the equation we saw in class.

The function should return as output a Gaussian pyramid of the input image represented as a python dictionary, where G{0} is the highest resolution level (i.e the original image).

You may use scipy.signal.convolve2d to perform the convolutions required to build the Gaussian pyramid.

Bonus 1 [10 pt] will be given for using your own implementation for the convolution operator rather than using any built-in function.

Bonus 2 [10 pt] will be given for convolution operator implementation that will achieve better running time compared to the school solution. **In order to get this bonus, you should implement the function ex5.imConv2(img, kernel1D) as used in ex5driver. test_1_bonus2() and add to the README.txt file output of running times test that shows that your implementation is same or even better than the school solution.**

See https://docs.python.org/2/library/timeit.html for example how to do running time tests from your code. ex5driver.test_1_bonus2 () can provide you such an example and the running time of the school solution you're competing with.

Task 2: Laplacian pyramid

a. You should write a function that gets as input a grayscale image, the number of levels in the Laplacian pyramid, and the filter parameter (a) that will be used to reconstruct the filter according to the equation we saw in class.

   The function should return as output a Laplacian pyramid of the input image represented as a python dictionary, where L{0} is the highest resolution level.

b. You should write a function that gets as input a Laplacian pyramid of some image and return as output the image reconstructed from its Laplacian pyramid. The inputs include, the Laplacian pyramid, the number of levels in the pyramid and the filter parameter used to build the convolution kernel.

Note: The expand operation involve spreading the intensity of the image over a domain which is K^2 times larger than the size of the level you perform the expand on. You may take this into account when you're implementing your expand operation.

Task 3: Image blending

In this task you should implement an image blending function, that performs Laplacian pyramid based blending according to the algorithm we saw in class.

The function gets as input the images to blend, the mask to blend according to it, and the Laplacian pyramid parameters such as the number of levels and the filter parameter. The output of the function should be the blended image.

Files included:

ex5driver.py: an example that defines and demonstrates the API for the functions you should implement.

ex5.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:

Ex5.zip file contains:

1. ex5.py: the code of your solution. The file ex3driver.py should run with your ex3.py without any error or warning.

2. README.txt: include your names and id numbers in the first two lines. The output for bonus 2 in task 1 should be added to the README.txt file **after** the lines with your names and your id numbers.

Submission instructions:

Through the course moodle website