IMPR 2019

Exercise #2

Due date: Dec. 19, 2019, 11:55pm.

This exercise can be submitted in pairs.

The goal of this exercise is to practice the basic concepts of image transformations, applying spatial operations on images and using the Hough transform to detect parametric shapes 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: Geometrical transformations

a) Write a function that gets at least 3 pairs of matched points between images A and B, and returns an affine transformation from image A to image B. You may use the built-in numpy method for finding the least-squares solution for linear systems using Singular Value Decomposition (SVD): https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.linalg.lstsq.html
See https://www.cs.bgu.ac.il/~na171/wiki.files/SVD_application_paper[1].pdf to better understand why it is more robust to use SVD rather the direct pseudo-inverse in many image-processing/compute vision problems.

b) Write a function that gets an image A and an affine transformation T, and return the transformed image T*A. You should implement a bi-linear interpolation function to calculate new pixel values.

c) Write a function that gets an image and two sets of corresponding lines (draw on two different images) and return the deformed calculated from these lines using the Multiple Segment Warping algorithm. Each line is represented by two points (start and end). The function also gets as input the two algorithm parameters (p, b) You should use the bilinear interpolation you implemented in 1.b.

Task 2: Image Gradients

a) We saw in class that gradient calculation over an image is approximated by the difference between neighboring pixels, which in turn can be viewed as a convolution operator. Determine the operator represent each of the three difference operations that are shown below.
   1. $f(x,y) - f(x-1,y)$
   2. $f(x+1,y) - f(x,y)$

3. $f(x+1,y+1)-f(x-1,y+1)+2[f(x+1,y)-f(x-1,y)]+$
   $f(x+1,y-1)+f(x-1,y-1)$

b) We saw in class the Sobel operator for image gradient calculation. The Definition of the Sobel operator is given by the following 3x3 2D matrix:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

For computing the gradient along the x-direction and its transpose to calculate the gradient along the y-direction. A two-dimensional kernel is separable if it can be expressed as the outer product of two vectors. Show that the Sobel operator can be decomposed into an outer product of 2 vectors.

c) Implement a function that gets an image and returns the gradient images (direction and magnitude. You should use the Sobel operator to calculate the image gradient. Note that you should implement the efficient version of the Sobel operator by using 2 vectors rather than a matrix kernel. You should implement the entire function yourself and not use any built-it convolutional functions from numpy or other libraries. Please consider carefully how to handle the boundaries of your input image efficiently and describe in your readme your approach.

Files included:

ex2driver.py: an example that defines and demonstrates the API for the functions you should implement.

ex2.pyc: the binary version of the school solution, so you can run and get an idea of what your solution should look like.

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

You should submit:

Ex2.zip file contains:

1. ex2.py: the code of your solution. The file ex2driver.py should run with your ex2.py without any error or warning.

2. README: include your names and the theoretical answers to task 2.

Submission instructions:

Through the course moodle website