

ECE5554 SU22 - Prof. Jones – HW 3

Due Wednesday, July 27, 2022 – 11:59 PM via Canvas

In this assignment you will implement Harris corner detection, and compare its results to the output of the OpenCV corner detection method. I am giving you four images to operate on: "AllmanBrothers.png", "CalvinAndHobbes.png", "Chartres.png", and "Elvis1956.png". They are in the "Image Files" section of the "Files" page in Canvas.

You are to write and test a Python/OpenCV program that will do the following:

1. For each of the images I have supplied:
 - a. Load the image and make a couple of copies of the color version of the image. The image contents are all grayscale, but the images still have three color planes so we can draw colored dots into them later on.
 - b. Convert the original image to grayscale.
 - c. Find the top 100 keypoints by using the OpenCV function `goodFeaturesToTrack`.
 - d. Print (to the console) the filename and the [X,Y] coordinates of the top three corner points (as determined by the corner strength function).
 - e. In the first copy of the color image, draw small green circles at each of the 100 keypoints.
 - f. Call your own implementation of the Harris corner detector (see below) to find the top 100 corner points.
 - g. Print (to the console) the filename and the [X,Y] coordinates of the top three corner points (as determined by the corner strength function).
 - h. In the second copy of the color image, draw small red circles at each of the 100 Harris corner points.
 - i. Put the two display images together (with your Harris results image on the right), using the `numpy concatenate()` function (with `axis=1`), and save this image to disk with an appropriate filename. Here is an example of what you should see:



Be sure and obey proper practice for loading, converting and using image files.

Your Harris corner detection function must be implemented using only numpy functions such as `np.multiply()`, `np.minimum()` and so on – with two exceptions. You may use only two OpenCV functions in this section of your code, if you find them helpful – `cv2.GaussianBlur()` and `cv2.Sobel()`. If you use any other OpenCV, scikit or any other functions external to Python and numpy, you will receive zero points for this part of the assignment.

Your submission will consist of a Word or pdf file and a .py file (no .ipynb files) containing your code, as well as four image files (the combined annotated images, similar to the one above). Do NOT put all of your files into a zip file; attach them separately. Your Word file should contain:

- your complete Python code (pasted in as plain text, no screenshots or dark mode);
- the console output of your program as described above, pasted in as plain text;
- the four combined annotated images from your program, sized large enough to see the detail; and
- a brief discussion of the differences in your corner points and the OpenCV keypoints. Are they the same? If they are different, how different are they? What could explain the differences, if any?