Optional Mini-Assignment 08

This assignment is optional. If you choose not to attempt it or do poorly on it, it will be left blank in the gradebook.

Your mission, should you choose to accept it, is to implement the Harris corner detection algorithm using numpy and some "low level" OpenCV functions. Recall from class that the corner response function at each pixel is given by

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

where λ_1 and λ_2 are the eigenvalues of the second moment matrix. The second moment matrix is defined at every pixel, and is given by

$$M(i,j) = \sum_{k,l} \begin{bmatrix} G_x(i+k, j+l)^2 & G_x(i+k, j+l)G_y(i+k, j+l) \\ G_x(i+k, j+l)G_y(i+k, j+l) & G_y(i+k, j+l)^2 \end{bmatrix}$$

where G_x and G_y are the image gradients in the x and y directions (i.e. outputs of Sobel operations). You may use `cv.Sobel()` for computing `Gx` and `Gy`.

A useful way to think about this is that, rather than having a 2x2 matrix at every pixel, we can instead keep track of all the $M[0,0]$ values as a 2D array, all the $M[0,1]$ values as a 2D array, and all the $M[1,1]$ values as a 2D array (we don't need $M[1,0]$ because it equals $M[0,1]$). This is the recommended approach (see the following diagram):

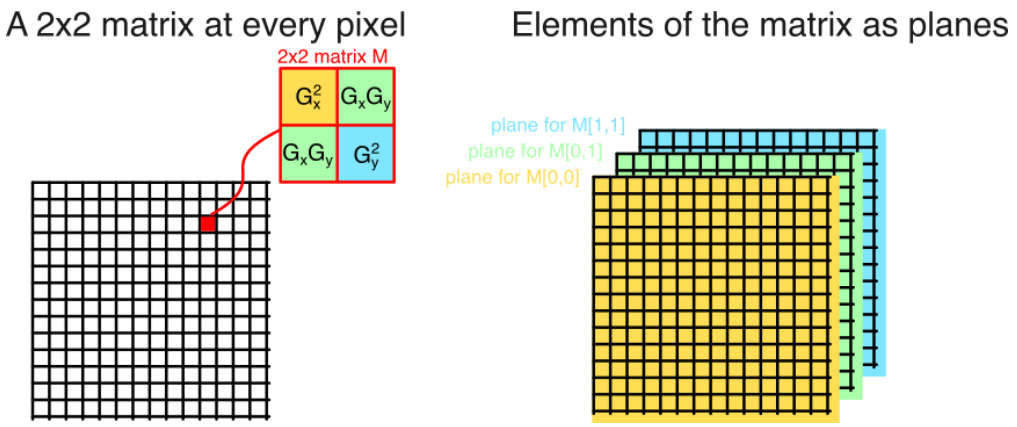

Figure 1: Rather than a "2x2 matrix at each pixel", think about having a separate plane for the elements of the 2x2 matrix.

The following identities about sums and products of eigenvalues will also be useful:

- $\lambda_1 + \lambda_2 = \text{trace}(M)$

- $\lambda_1 \lambda_2 = \det(M)$

...and, you can easily look up what the trace and determinant of a 2x2 matrix are in terms of its elements. In other words, you can compute R using simple multiplies and sums of the three yellow/green/blue image planes in the diagram above.

One last hint: the $\sum_{k,l}$ in the definition of $M(i,j)$ is a sum over a window of pixels and may be implemented using a simple box filter!

What you need to do: Fill in the missing code in `harris_corners.py`. When you run the file, it will display an interactive window with OpenCV corners on the left and yours on the right. If you do everything correctly, they should match.

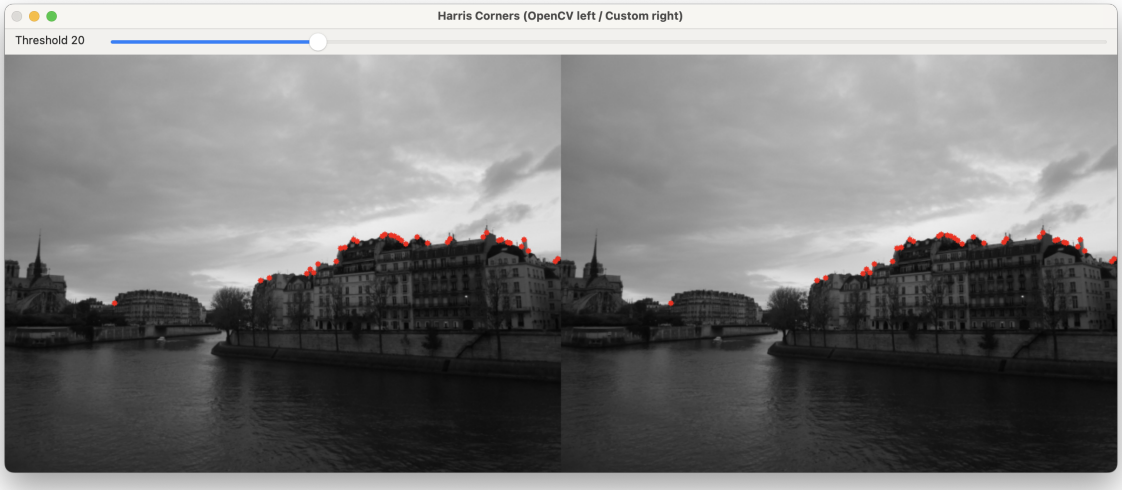

Figure 2: Screenshot of a working solution

Collaboration and Generative AI disclosure

Did you collaborate with anyone? Did you use any Generative AI tools? Briefly explain what you did in the `collaboration-disclosure.txt` file.

Reflection on learning objectives

This is optional to disclose, but it helps us improve the course if you can give feedback.

- what did you take away from this assignment?
- what did you spend the most time on?

- about how much time did you spend total?
- what was easier or harder than expected?
- how could class time have better prepared you for this assignment?

Give your reflections in the `reflection.txt` file if you choose to give us feedback.

Submitting

Use `make submission.zip` to generate a zip file that you can upload to Gradescope. You can upload as many times as you like before the deadline. To account for late penalties, contact the instructor or grader directly if you plan on uploading any further revisions after the deadline. (This is to prevent us from having to grade your work twice. Let us know so that we don't start grading files that will be replaced later).