

Assignment 2

Due: 10/17/17 (by 5 pm)

Notes:

- Be sure to re-read the “Policy on Academic Integrity” on the course syllabus.
- This assignment is to be done individually. General help is fine (e.g., “When was that covered?”, “What does that mean again?”, “How do I run Matlab?”, etc.), but the problems are to be worked on, and the answers conceived, on your own.
- Any updates or corrections will be posted on e-campus, so check there occasionally.
- Be aware of the late policy in the course syllabus.
- Non-running code or large errors in the outputs will be severely penalized.

Instructions:

1. Download the ‘checkerboard.png’ image for assignment 2 from e-campus.
2. Images should be processed so that the pixel intensities are in the range $[0, 255]$.
3. Implement your solution and turn in your code via turn-it-in on e-campus.
4. Turn in the sorted 100 strongest corners in descending order.
5. Place comments explaining the computations.
6. Include a README file including your name and e-mail address as well a few paragraphs explaining how to compile and/or execute your program.
7. Turn in any necessary file for executing your programs (e.g., Makefiles).

-
1. **Harris corner detector.** Implement the Harris corner detector. The detector must accept the following parameters:
 - Size of window (input).
 - Parameter for corner response k (input).
 - Corner response threshold τ (input).
 - The input image (input).
 - A list of pixel coordinates sorted by corner response in descending order (output).

Note: Do not implement a non-maximum suppression.

Validation. Download the 'checkerboard.png' image from e-campus and detect Harris corners using your implementation. Use a window of size 5, $k = 0.04$, $\tau = 10000$. Every corner must have a record in the list of the form $x \ y \ r$, where x is the horizontal coordinate, y is the vertical coordinate, and r is the corner response. For the submission of the 100 strongest corners, create a file where each corner only has a single space in between its entries ($x \ y \ r$) and a row per corner. For example, your file should look like this:

```
64    64    3.7157e9
64    65    3.7157e9
64   128    3.7157e9
64   129    3.7157e9
64   192    3.7157e9
...
```

If possible, visualize the detected corners by drawing a circle around their centers.