

Department of Computer Science
The University of Hong Kong
COMP 7502
Assignment 2

The purpose of this assignment is to get you familiar with Gaussian smoothing, the Harris corner detection algorithm and camera calibration. You are required to finish the missing code in the template provided. Modify ONLY the file Assignment2.java and submit your finished Assignment2.java file using Moodle.

Please be reminded of the policy regarding plagiarism:

<https://intranet.cs.hku.hk/csintranet/contents/general/shared/plagiar.jsp>

Please note the following:

1. ONLY hand-in your modified version of Assignment2.java. Do not modify any other files.
2. You should implement additional helper methods if necessary, e.g. to implement the convolution for Task 1 and 2.
3. The test image contain a pattern with two perpendicular planes which will be used to generate a large number of 2D-3D point correspondences. Manually marked corners are provided and a planar calibration has been implemented in the code proved to you. The planar calibration is employed to generate the large number of 2D-3D coordinates.
4. Eight manually marked 2D - 3D point correspondences are provided if you select 'load test image 1 from web' or 'load test image 2 from web'. You may view the details of the correspondence points by moving the mouse over the magenta circle.
5. If you prefer to work without an internet connection, you may select 'open image...' and open the image 'testImage1.jpg' which is available on moodle. Manually marked 2D-3D correspondences are only available for testImage1.jpg. Do not open any other images.
6. A large number of 2D - 3D correspondences will be generated once your Harris corner detection code works. You should finish task 2 before working on task 3.

You are required to complete the following tasks in Assignment2.java:

public void gaussianSmooth(final byte[] f, int width, int height, double sigma)

Implement 2D Gaussian smoothing (5 Marks)

public ArrayList<double[]> detectCorners(final byte[] f, int width, int height, double sigma, double threshold)

Implement the Harris Corner Detection Algorithm (5 Marks)

private Matrix performCalibration(ArrayList<double[]> points2d, ArrayList<double[]> points3d)

Perform camera calibration and return the 3x4 camera projection matrix from the provided correspondences. (5 Marks)