CS-E4850 Computer Vision
Exercise Round 9

The following instructions are for the Matlab version. The instructions for the Python version are in Github. Matlab is available on Aalto computers and also for students' own computers via `https://download.aalto.fi`.

The problems should be solved before the exercise session and solutions returned via the MyCourses page. Upload two files: (1) a PDF file containing your written answers to both problems, (2) a ZIP file containing the source codes.

Get the example `m`-files by downloading `Exercise09.zip` from the MyCourses page.

Exercise 1. Camera calibration. (Matlab exercise)
In this exercise you will need to implement the direct linear transform (DLT) method for camera calibration. The algorithm is described on slide 22 of Lecture 8. It is also presented in the book by Hartley & Zisserman (Section 7.1 in the second edition).

The calibration object is a bookshelf whose dimensions are known. That is, width of a shelf is 758 mm, depth is 295 mm, and height between shelves is 360 mm.
Proceed as follows:

a) Run the Matlab example script `camcal_example.m` from `Exercise09.zip`.

b) The corners of the bookshelf are already manually localized from the given two images and visualized by the script. See the comments in the source code.

c) Implement the missing function `camcalibDLT.m`.

d) Calibrate both cameras and check the results visually by uncommenting the relevant lines from the example code.

e) Report the estimated intrinsic camera calibration matrices for both cameras (i.e. variables $K1, K2$).

Tasks continue on the next page...

Exercise 2. Triangulation. (Matlab exercise)

In this exercise you will need to implement the linear triangulation method described on slides 31 and 32 of Lecture 8. (The method is also presented in the book by Hartley & Zisserman.) You must again find a least-squares solution to a system of linear equations. In a similar manner as in exercise problem 1, it can be computed by solving the eigenvectors and eigenvalues of a real symmetric matrix (see Matlab built-in function `eig`).

As illustrated by the example script, the points that will be triangulated are the corner points of the picture on the cover of the course book. As a result of the triangulation we will get the coordinates of these corner points in the world coordinate frame. By computing the distances between the points, we can measure the width and height of the picture in millimeters.

Proceed as follows:

a) Run the Matlab example script `triangulation_example.m` from `Exercise09.zip`.

b) The corners of the picture on the book cover are already manually localized from the given two images and visualized by the script. See the comments in the source code.

c) Implement the missing function `trianglin.m`.

d) Triangulate the three given point correspondences. Here you will need also the estimated camera projection matrices $(P1, P2)$ from exercise problem 1. You can use the same projection matrices here because the two images are the same as in the calibration problem.

e) Report the estimated width and height of the picture by computing the distances between triangulated points.