Assignment 7

This exercise is part of the course assignment. **Deadline for the assignment 30.11.2022 at 23:59**

The topic of this assignment is camera calibration. For this assignment you should return

- The files main7.m and camcalibDLT.m. The file should contain your name and student number (of both students if you work in pairs).
- Your answers to the questions in the analysis part. At the end of the course, you should return a single pdf containing the answers to all questions of the assignments. The report should contain also your name and student number (of both students if you work in pairs).

Coding part (5pt)

- First, run demo7.p to get a taste of this task.
- Let $P = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix}$ the camera projection matrix Let $\mathbf{X} = [X, Y, Z]^\top$ a 3D point and $\mathbf{x} = [x, y]^\top$ the corresponding image point, then the following equation holds

$$\begin{bmatrix} 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \\ X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \mathbf{0} \tag{1}$$

  Since P has 11 degrees of freedom (make sure you understand why!) and from each 3D-2D correspondence we get 2 equations, we need at least 6 2D-3D correspondences to perform camera calibration. This way from (1) we obtain the homogeneous system $A\mathbf{p} = \mathbf{0}$ which can be solved using homogeneous least-squares (see exercise 3). Your task is to implement the function P=camcalibDLT(X, x) which takes as input a matrix $X \in \mathbb{R}^{n\times 3}$ containing the coordinates of the 3D points, a matrix $\mathbf{x} \in \mathbb{R}^{n\times 2}$ containing the coordinates of the 2D points and compute the projection matrix P. You can assume $n \geq 6$. Particularly you should

  - Assemble the homogeneous system using (1)
  - Solve the system using homogeneous least squares (see exercise 3 of this week exercises for hints)

- reshape the vector $\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$ into the projection matrix P.

- When you are done, open the script main7.m. In the first cell we are reading two images and loading the 3D coordinates of 8 points (matrix X) as well as the 2D coordinates of the corresponding 2D points in the 2 images (matrices x_1 and x_2).

- In the second cell we are estimating the projection matrix for each image. If you have done the previous task correctly, this cell should run without error.

- In the third cell we want to project the 3D points to the images to visually evaluate how good our estimate was. Complete this cell, particularly
  - from the 8×3 matrix X create the 8×4 matrix X_{hom} containing the homogeneous coordinates of the 3D points
  - Project the 3D points to the two images using the projection matrices P_1 and P_2 you estimated before. Store the results into the matrices $x1proj$ and $x2proj$.
  - Convert $x1proj$ and $x2proj$ to Cartesian coordinates. The final arrays should have size 8×2.

- In the last cell we are extracting $K, R, veect$ from P. Go through the codes rq.m and dissect_P.m, you will need to answer some question in the analysis part.

Analysis part (5pt)

Answer the following questions in your report

- Derive equation (1). (2pt)
  Hint: Start from the camera equation

$$\lambda \tilde{\mathbf{x}} = P\tilde{\mathbf{X}}, \tag{2}$$

  where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{X}}$ are now homogeneous coordinates. Take the cross product $\tilde{\mathbf{x}}\times$ on the left of both sides of the equation.

- The projection matrix P is obtained as $P = K[R\ \mathbf{t}]$, where K is the upper triangular matrix containing the instrinct paramters and $R, \mathbf{t}$ are the rotation matrix and translation vector of the camera (shortly referred to as camera pose). The function `dissect_P` takes as input the projection matrix P and computes the $K, R, \mathbf{t}$. The first step to extract K and R is to compute the so-called RQ-decomposition of $P[1:3, 1:3]$, which decomposes the matrix into the product of an upper triangular matrix and unitary matrix (similarly to QR, but the other way around). **Note!** be careful to the notation, in RQ-decomposition, R is the upper triangular matrix and Q is the unitary matrix, but in computer vision R is the rotation (and thus unitary) matrix and K is the upper triangular matrix.
  - In general a unitary matrix Q can have $\det(Q) = \pm 1$, but at the end of the script `rq.m` we impose the determinant of the unitary matrix to be positive. Why? (1pt)
  - At line 8 of `dissect_P` we impose $K = K/K(3,3)$. Why do we do that and why are we allowed to do that? (1pt)
  - What are we doing at lines 10-14 of `dissect_P` and why are we doing it? (1pt)