Assignment 27/03/2019

Camera calibration with Tsai algorithm

Introduction

The goal of this practical session is to implement the algorithm proposed by Tsai to calibrate a camera. The principle is to have as input a set $\{(\mathbf{X_i}, \mathbf{x_i})\}_{i=0\ldots n-1}$ of 3D points X_i expressed in world coordinates and their 2D projection x_i expressed in pixel coordinates. From this input, the projection matrix P can be computed, from which the internal parameters can be extracted in the matrix K. Let us recall the equation:

$$P = \arg\min_P \|x_i \times (PX_i)\|^2, \tag{1}$$

which can be solved by simply stacking the equations

$$x_i \times (PX_i) = 0 \tag{2}$$

as a system $Ap = 0$ with $p = \begin{pmatrix} P_1^\top & P_2^\top & P_3^\top & P_4^\top \end{pmatrix}^\top$ and the P_j are the columns of P.

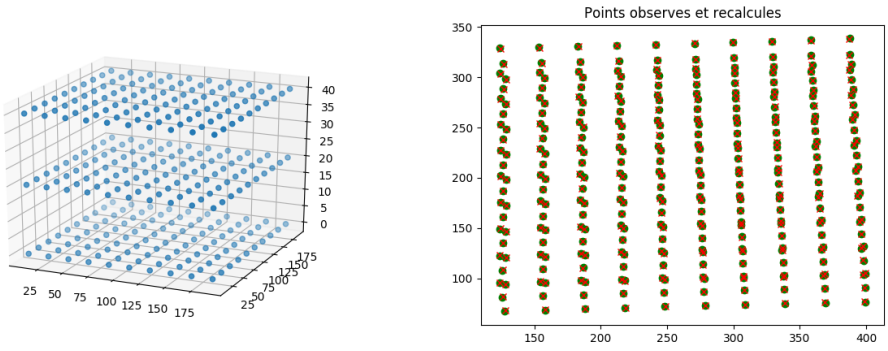

Figure 1: Left: observed 3D points. Right: their observed image in green and their projection through estimated matrix P as red crosses.

Question 1

The equation (2) results in only two linearly independent equations on P. Write these equations.

Question 2

Once P is computed, we have to extract K from it. This is a decomposition RQ of the 3×3 left part of P. Unfortunately, only QR decomposition is proposed by `numpy`. Using the anti-identity matrix J

of the course, and observing that $J = J^\top = J^{-1}$, show that from the QR decomposition of a matrix JM^TJ we can extract the RQ decomposition of M. For that, explain the effect of multiplying a matrix by J from the left or from the right.

Question 3

Implementation in Python:

1. Load data points of file `data.txt` using function `numpy.loadtxt`. Each line is of the form

$$X \quad Y \quad Z \quad x \quad y$$

2. Separate into two matrices containing one the 3D points and the other the 2D points, all in homogeneous coordinates (appending an extra coefficient of 1).
3. Plot in 3D the points of the calibration rig using `mpl_toolkits.mplot3d.Axes3D.scatter`
4. Build the matrix A from the data.
5. Write as a system $Ax = b$ assuming the last unknown to be actually 1. Solve the system in the least square sense with `numpy.linalg.lstsq`.
6. Perform the RQ decomposition (see Question 2) to recover matrix K using function `numpy.linalg.qr`.
7. Show in a 2D plot the observed projected points in the file and the projections of 3D points through the estimation of P.

Guidelines

Each student should submit a single zip file that will contain all the required code as well as a PDF document with the responses to the questions. For queries contact monasse@imagine.enpc.fr.