CS410 Computer Vision

Assignment 1 - Camera Calibration

Lecturer: Dr. John McDonald
Submission deadline: 10 November 2017

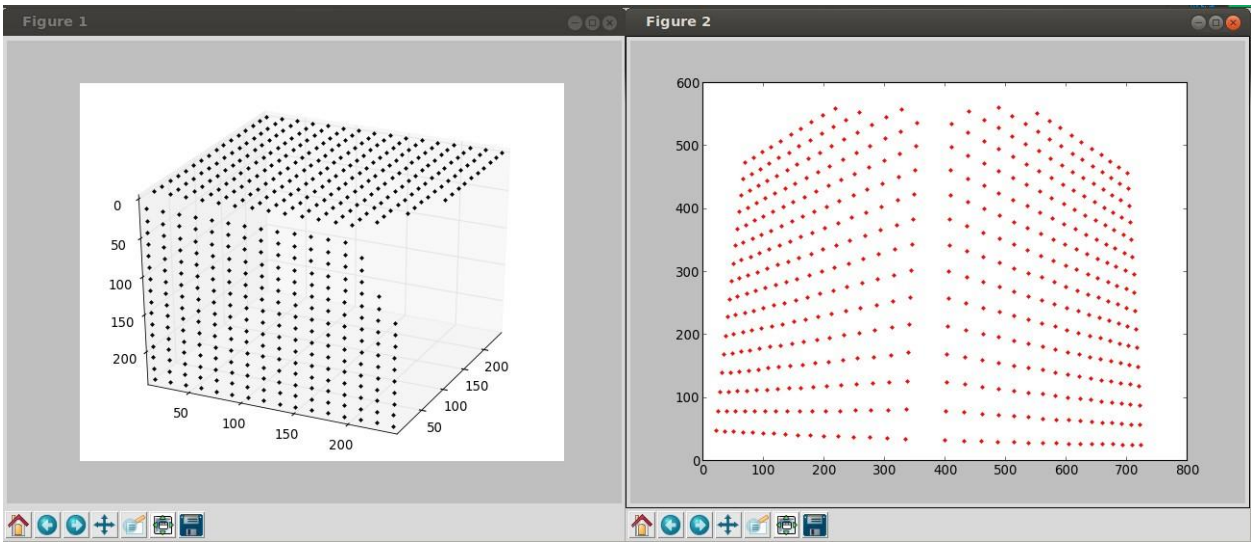
0. Introduction

In this assignment you will be required to write a number of python functions to calibrate a camera based on real-world data. The data includes a set of point correspondences consisting of a set of 3D input points and their 2D projections into image space. The objective of the assignment is to write a set of python functions which given these correspondences computes the associated 3×4 camera matrix $\mathbf{P}$.

Don't worry if you are new to python; it's a very easy language to pick-up, and we will only be using some basic elements of the language. In order to get you started you will find some links on the last slide of the section 2 lecture notes. The most useful of these is the online tutorial at: https://scipy-lectures.github.io/
In particular I would recommend you take some time to go through the first two chapter, providing and intro to python, the third chapter, providing an overview of using numpy, and possibly the fourth chapter on plotting data.

1. Camera calibration

Download the `data.txt` file and the from the course webpage. This file contains a series of world point to image point correspondences from a real-world 3D camera calibration target. The figure below shows plots of the 3D points and 2D image projections, respectively.



The `data` matrix is a N`x`5 matrix where each row defines an individual correspondence. The first three elements of each row define the (X,Y,Z) world coordinates of the point. The fourth and fifth element of the row define the (x,y) image coordinates of the point. In order to get you started with working with the data file, you will also find a LoadCalib.py file on the moodle website that gives some simple code for loading and visualising the data.

Given the above data, write a function in python that takes this matrix of correspondences as input, and returns the perspective projection matrix of the camera as output. The function should have the following prototype

```
P = calibrateCamera3D(data)
```

Provide two further functions

```
visualiseCameraCalibration3D(data, P)
evaluateCameraCalibration3D(data, P)
```

which should both take as input the original calibration data and the camera matrix P. The first function should render a single 2D plot showing (i) the measured 2D image point (i.e. as shown on the right in the above figure), and, (ii) the reprojection of the 3D calibration points as computed by P. The second function should print the mean, variance, minimum, and maximum distances in pixels between the measured and reprojected image feature locations.

2. Submission

You should submit your final code as a single python file via the associated link on the moodle course page. The file should contain definitions for the three functions defined above **and** driver code to demonstrate the operation of the three functions.