PeAR WPI Teaching

AutoCalib

Table of Contents:

1. Due Date

11:59:59 PM, February 15, 2025. This homework is to be done individually and NO LATE DAYS ARE ALLOWED.

2. Introduction

Estimating parameters of the camera like the focal length, distortion coefficients and principle point is called **Camera Calibration**. It is one of the most time consuming and important part of any computer vision research involving 3D geometry. An automatic way to perform efficient and robust camera calibration would be wonderful. One such method was presented Zhengyou Zhang of Microsoft in this paper and is regarded as one of the hallmark papers in camera calibration. Recall that the camera calibration matrix K is given as follows

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

and radial distortion parameters are denoted by k_1 and k_2 respectively. Your task is to estimate $f_x, f_y, c_x, c_y, k_1, k_2$.

3. Data

The Zhang's paper relies on a calibration target (checkerboard in our case) to estimate camera intrinsic parameters. The calibration target used can be found in the file `checkerboardPattern.pdf` Link. This was printed on an A4 paper and the size of each square was 21.5mm. Note that the Y axis has odd number of squares and X axis has even number of squares. It is a general practice to neglect the outer squares (the row and columns of squares on the edges of the checkerboard). For example, if you have a 5×8 grid, you'll only consider the inner 3×6 grid for computation. **Again, this is not necessary but a common practice.** Thirteen images taken from a Google Pixel XL phone with focus locked can be downloaded from here which you will use to calibrate.

4. Initial Parameter Estimation

We are trying to get a good initial estimate of the parameters so that we can feed it into the non-linear optimizer. We will define the parameters we are using in the code next.

x denotes the image points, X denotes the world points (points on the checkerboard), $k = [k_1, k_2]$ denotes the radial distortion parameters, K denotes the camera calibration matrix, R and t represent the rotation matrix and the translation of the camera in the world frame.

4.1. Solving for approximate K or camera intrinsic matrix

Refer to Section 3.1 in this paper for a solution of parameters in K. Use `cv2.findChessboardCorners` function in OpenCV to find the corners of the Checker board with appropriate parameters here.

4.2. Estimate approximate R and t or camera extrinsics

Refer to Section 3.1 in this paper for details on how to estimate R and t. Note that the author mentions a method to convert a normal matrix to a rotation matrix in Appendix C, this can be neglected most of the times.

4.3. Approximate Distortion k

Because we assumed that the camera has minimal distortion we can assume that $k = [0, 0]^T$ for a good initial estimate.

5. Non-linear Geometric Error Minimization

We have the initial estimates of K, R, t, k, now we want to minimize the geometric error defined as given below

$$\sum_{i=1}^{N} \sum_{j=1}^{M} ||x_{i,j} - \hat{x}_{i,j}\left(K, R_i, t_i, X_j, k\right)||$$

Formally, the optimization problem is as follows:

$$\text{argmin}_{f_x, f_y, c_x, c_y, k_1, k_2} \sum_{i=1}^{N} \sum_{j=1}^{M} ||x_{i,j} - \hat{x}_{i,j}\left(K, R_i, t_i, X_j, k\right)||$$

Here $x_{i,j}$ and $\hat{x}_{i,j}$ are an inhomogeneous representation. Feel free to use `scipy.optimize` to minimize the loss function described above. Refer to Section 3.3 in this paper for a detailed explanation of the distortion model, you'll need this part for the minimization function.

6. Submission Guidelines

If your submission does not comply with the following guidelines, you'll be given ZERO credit.

6.1. File tree and naming

Your submission on ELMS/Canvas must be a `zip` file, following the naming convention `YourDirectoryID_hw1.zip`. If you email ID is `abc@wpi.edu`, then your `DirectoryID` is `abc`. For our example, the submission file should be named `abc_hw1.zip`. The file **must have the following directory structure** because we'll be autograding assignments. The file to run for your project should be called `Wrapper.py`. You can have any helper functions in sub-folders as you wish, be sure to index them using relative paths and if you have command line arguments for your Wrapper codes, make sure to have default values too. Please provide detailed instructions on how to run your code in `README.md` file. Please **DO NOT** include data in your submission.

```
YourDirectoryID_hw1.zip
|   README.md
```

```
|   Your Code files
|               ├── Any subfolders you want along with files
|   Wrapper.py
└──     Report.pdf
```

6.2. Report

For each section of the homework, explain briefly what you did, and describe any interesting problems you encountered and/or solutions you implemented. You must include the following details in your writeup:

- Your report **MUST** be typeset in LaTeX in the IEEE Tran format provided to you in the `Draft` folder and should of a conference quality paper.
- Present images of checkerboard after rectification and reprojection of corners on rectified image.
- Present the value of re-projection error.
- Clearly, specify the K matrix and k vector in the report.

7. Collaboration Policy

NOTE: You are **STRONGLY** encouraged to discuss the ideas with your peers. Treat the class as a big group/family and enjoy the learning experience.

However, the code should be your own, and should be the result of you exercising your own understanding of it. If you reference anyone else's code in writing your project, you must properly cite it in your code (in comments) and your writeup. For the full honor code refer to the RBE/CS549 Spring 2025 website.

8. Acknowledgements

This fun homework was inspired by a similar homework in University of Maryland's CMSC733 (Classical and Deep Learning Approaches for Geometric Computer Vision).