

Camera Calibration

Leopold Gaube - 34948, IN

June 7, 2022

Contents

1	Introduction	2
2	Task Overview	2
3	Frame Selection	2
4	Intrinsic Camera Matrix	3
5	Undistorting	4
	List of Acronyms	5
	References	5

1 Introduction

When taking images or video with a camera, a 3D scene gets projected onto a 2D image. Naturally, this is accompanied with a loss in information as the distance of an object to the camera is lost. Therefore, a perfect reconstruction of the original 3D scene from a 2D image is impossible. However, ...

Another problem is that some supposed straight lines, e.g. a door, do not appear to be straight in the resulting image. This is due to lens distortions. (would not have been an issue with pinhole camera model) The provided video footage has been captured with a ultrawide lens, thus it suffers from Barrel Distortion.

(radial and tangential distortion)

In order to undistort the images, we first have to retrieve the intrinsic matrix K

- 3D scene projected onto 2D image (loss of information) - distortions - intrinsic and extrinsic camera parameters - easily recognisable patterns, OpenCV uses a simple checkerboard pattern. (printed, flat) - can be used to undistort images/frames

Problem: We usually don't manufacture the sensor ourself How to calculate K and (R, t)

[Ope22] The code and all frames used for this camera calibration task are available in this GitLab repository [Gau22].

2 Task Overview

- multiple images (minimum of 3?) of a known object - best results if taken from various positions - compute camera matrix, distortion parameters?, rotation and translational vectors (extrinsics?) - finally, remove distortion

3 Frame Selection

In theory, we only need three frames to retrieve the intrinsic camera matrix. (why?) In practise, more frames give better results and it is also advisable to use frames with the checkerboard present in different parts of the image. It is possible to manually extract appropriate frames from the calibration video, but we chose to automate this process in order to minimize work for the user for future camera calibrations.

First we have to specify how many calibration frames should be used. For our algorithm we have settled on 15, as this creates good results in a reasonable execution time. Naturally, we do not want to use consecutive frames, but frames that span the entire video. This way, we get varying checkerboard positions (assuming the checkerboard is being moved around in the calibration video).

Some of them may have a lot of motion blur which can be problematic for accurately localizing the checkerboard corners. So instead, for each evenly spaced time step of the video, we consider 50 consecutive candidate frames (corresponding to two seconds of video) from which we only choose the least blurred one. The extend of blur in an image can be calculated using the Variance of Laplacian. [BRC16] With OpenCV it is simple to obtain a focus measure for any greyscale image:

```
focus_score = cv2.Laplacian(frame, cv2.CV_64F).var()
```

The absolute value is of no interest, but the highest value of our 50 candidate frames corresponds to the sharpest one.

4 Intrinsic Camera Matrix

$$K = \begin{pmatrix} f * k_1 & 0 & s_1 \\ 0 & f * k_2 & s_2 \\ 0 & 0 & 1 \end{pmatrix}$$

K is an upper triangular matrix, therefore, the determinant of K is the product of its trace, which is always non-zero because the focal point f, as well as k_1 and k_2 (and theta?) is greater than 0.

K: focal length, mm step size, sheering angle translation –i intrinsic parameters, only 5 DoF not 6! because we dont care about the individual parameters themself, only the resulting matrix which is a combination of these parameters

Our Algorithm retrieve the following intrinsic matrix:

$$K = \begin{pmatrix} f * k_1 & 0 & s_1 \\ 0 & f * k_2 & s_2 \\ 0 & 0 & 1 \end{pmatrix}$$

K has to be calculated only once and can be reused in a different scene assuming the same camera is being used.

5 Undistorting

List of Acronyms

References

- [BRC16] Raghav Bansal, Gaurav Raj, and Tanupriya Choudhury. Blur image detection using laplacian operator and open-cv. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pages 63–67, 2016.
- [Gau22] Gaube. Camera calibration. <https://gitlab.com/gaubeleo/camera-calibration>, 07.06.2022.
- [Ope22] OpenCV. Opencv - about. <https://opencv.org/about/>, 21.04.2022.