Camera Calibration and Epipolar Geometry Using OpenCV

CHINMAY AMRUTKAR
chinmay.amrutkar@asu.edu

1. Exploring Epipolar Geometry

1.1. Overview and Objective

The goal of this project is to explore fundamental concepts in computer vision related to multi-view geometry, specifically through the practical implementation of two key tasks: **camera calibration** and **epipolar geometry estimation**. Camera calibration allows us to recover the intrinsic properties of a camera — including focal length, optical center, and distortion coefficients — by observing a known 3D structure (a planar checkerboard) from multiple viewpoints.

Building upon the calibration, we then estimate the **fundamental matrix** between a pair of images captured from different perspectives of a static scene. The fundamental matrix describes the geometric relationship between corresponding points in the two images and allows us to visualize epipolar lines and compute the essential matrix, which encodes the relative camera pose (rotation and translation) under the pinhole camera model.

This pipeline is foundational to many 3D vision tasks including stereo reconstruction, structure-from-motion (SfM), and visual SLAM (Simultaneous Localization and Mapping).

1.2. Camera Calibration Using Checkerboard

Camera calibration is a fundamental process in computer vision that allows us to determine the internal characteristics of a camera, known as the intrinsic parameters. These parameters include the focal length, the optical center (also called the principal point), and the distortion coefficients that model lens imperfections. Accurate calibration is essential for applications such as 3D reconstruction, augmented reality, robotics, and any system that requires an understanding of the spatial relationship between the camera and the observed scene.

To perform camera calibration, we used a printed checkerboard pattern with 8×6 inner corners, and each square having a side length of 25mm. The checkerboard was fixed to a flat surface to avoid movement between shots. We then captured a total of 21 images using a mobile phone camera from a variety of angles and distances.

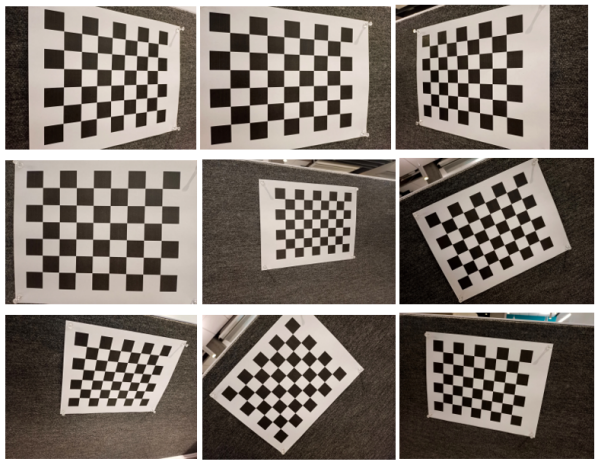

Figure 1. Sample images from the 21-shot dataset used for calibration. The checkerboard is viewed under varying angles and distances.

We used OpenCV's calibration functions to estimate the camera parameters. The process begins by detecting the 2D image coordinates of the checkerboard corners using `cv2.findChessboardCorners()`. These detected points are refined using sub-pixel accuracy methods with `cv2.cornerSubPix()`. The corresponding 3D real-world coordinates are generated using the known size and layout of the checkerboard. These 2D-3D correspondences are used to compute the intrinsic matrix $\mathbf{K}$, distortion coefficients, and pose (rotation and translation vectors).

```
import cv2
import numpy as np
import glob

checkerboard = (8, 6)
objp = np.zeros((checkerboard[0]*
↪  checkerboard[1], 3), np.float32)
objp[:, :2] = np.mgrid[0:8,
↪  0:6].T.reshape(-1, 2) * 25

objpoints, imgpoints = [], []
```

```
images =
↪  glob.glob('calib_images/*.jpg')

for fname in images:
    img = cv2.imread(fname)
    gray = cv2.cvtColor(img,
    ↪  cv2.COLOR_BGR2GRAY)
    ret, corners =
    ↪  cv2.findChessboardCorners(gray,
    ↪  checkerboard, None)
    if ret:
        objpoints.append(objp)
        corners2 =
        ↪  cv2.cornerSubPix(gray,
        ↪  corners, (11,11), (-1,-1),
        ↪  criteria)
        imgpoints.append(corners2)

ret, K, dist, rvecs, tvecs =
↪  cv2.calibrateCamera(objpoints,
↪  imgpoints, gray.shape[::-1], None,
↪  None)
```

$$\mathbf{K} = \begin{bmatrix} 3137.18 & 0 & 2080.91 \\ 0 & 3142.22 & 1545.15 \\ 0 & 0 & 1 \end{bmatrix}$$

The distortion coefficients are:

$$[0.1966, -0.7646, -0.0014, 0.0001, 0.8190]$$

These results confirm that the intrinsic parameters reflect a strong estimation of focal length and principal point, and the distortion coefficients capture both radial and tangential distortion. The mean reprojection error was approximately **0.2574**, indicating good alignment.

1.3. Estimating the Fundamental Matrix Between Two Views

To study epipolar geometry, we captured two images of a book from different perspectives using the same camera. We used the ORB feature detector and descriptor extractor to find keypoints and match them across images. Using these correspondences, we estimated the fundamental matrix $\mathbf{F}$ using RANSAC to reduce the influence of outliers.

```
orb = cv2.ORB_create()
kp1, des1 = orb.detectAndCompute(img1,
↪  None)
kp2, des2 = orb.detectAndCompute(img2,
↪  None)
matches =
↪  cv2.BFMatcher(cv2.NORM_HAMMING,
↪  crossCheck=True).match(des1, des2)

```

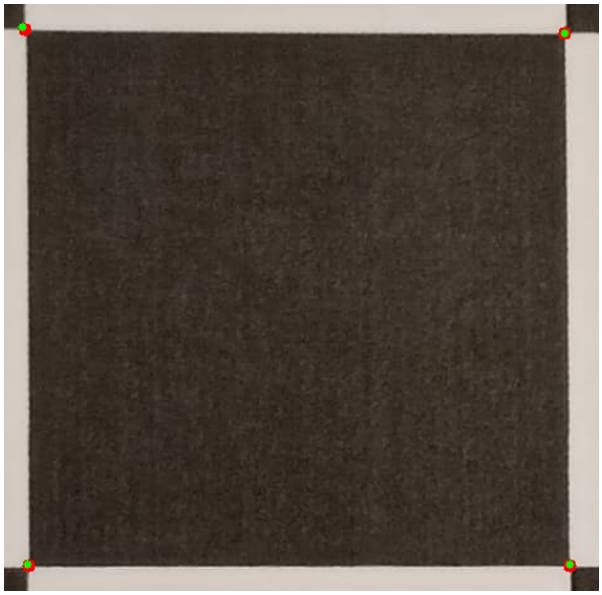

Figure 2. Overlay of detected (red) and reprojected (green) checkerboard corners.

```
pts1 = np.float32([kp1[m.queryIdx].pt
↪  for m in matches]).reshape(-1, 1,
↪  2)
pts2 = np.float32([kp2[m.trainIdx].pt
↪  for m in matches]).reshape(-1, 1,
↪  2)
F, mask = cv2.findFundamentalMat(pts1,
↪  pts2, cv2.FM_RANSAC)
E = K.T @ F @ K
```

The estimated matrices are:

$$\mathbf{F} = \begin{bmatrix} -1.17 \times 10^{-8} & 5.52 \times 10^{-8} & 1.40 \times 10^{-6} \\ 3.65 \times 10^{-7} & 6.54 \times 10^{-9} & -2.58 \times 10^{-3} \\ -4.72 \times 10^{-4} & 1.58 \times 10^{-3} & 1.00 \end{bmatrix}$$

$$\mathbf{E} = \begin{bmatrix} -0.1156 & 0.5439 & 0.1951 \\ 3.6006 & 0.0645 & -5.6819 \\ 0.2117 & 5.3499 & -0.2099 \end{bmatrix}$$

The epipoles were located at:

- Image 1: $[7032.91,\ 1472.13,\ 1.0]$
- Image 2: $[-28640.70,\ 372.33,\ 1.0]$

Both lie outside the image boundary, indicating lateral (side-to-side) camera motion.

1.4. Epipolar Line Visualization and Interpretation

As shown in Fig. 3, each colored epipolar line in the left image corresponds to a point in the right image. For instance, the first yellow line in image 1 passes through the

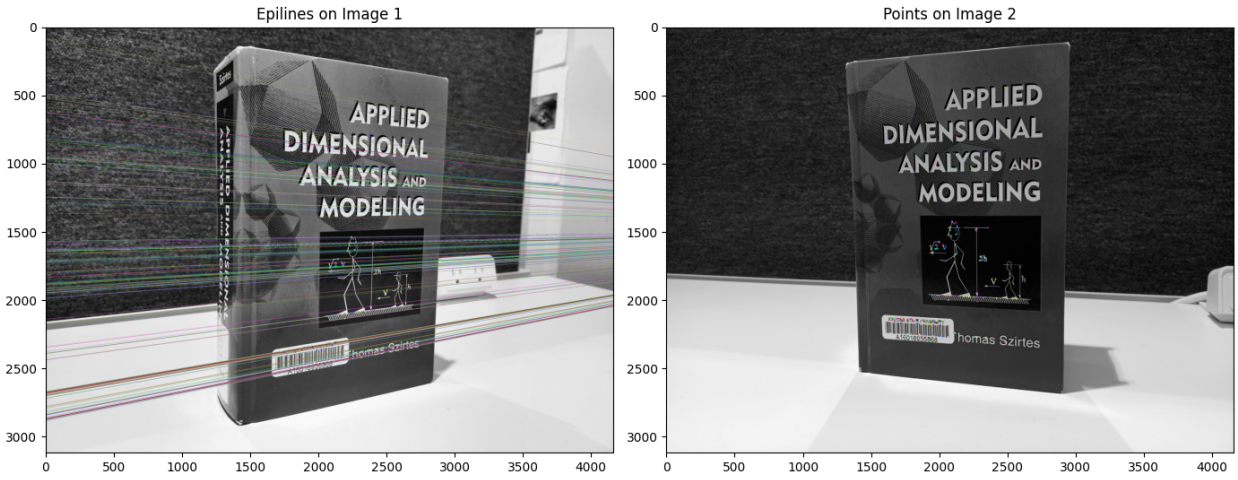

Figure 3. Epipolar lines (colored) in one image correspond to matching points in the other view.

letter "I" in the word APPLIED, and the yellow dot in image 2 lies near the same word — indicating a correct correspondence. These visualizations validate the accuracy of **F**.

1.5. Conclusion and Reflections

This project successfully demonstrated the use of OpenCV for modeling both the intrinsic and extrinsic properties of a camera system. Through the calibration of a mobile phone camera using a printed checkerboard, we obtained an accurate estimate of the intrinsic matrix and distortion coefficients, confirmed by a low reprojection error of 0.2574.

We then explored the geometry between two views of a scene using the estimated intrinsic matrix to compute both the fundamental and essential matrices. Epipolar lines were visualized to show how matched feature points correspond geometrically between views. The location of the epipoles and the shape of the lines were consistent with the physical setup of the cameras and scene.

This hands-on implementation reinforces the theoretical understanding of projective geometry and showcases the power of robust computer vision pipelines built upon well-established algorithms like RANSAC, ORB, and matrix decomposition. The skills demonstrated here serve as a strong foundation for more advanced tasks such as 3D reconstruction and motion estimation.

1.6. Acknowledgments

List of resources used:

- https://markhedleyjones.com/projects/calibration-checkerboard-collection
- https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html
- https://docs.opencv.org/3.4/da/de9/tutorial_py_epipolar_geometry.html