Corner Localization and Camera Calibration from Imaged Lattices

Andrii Stadnik [1] James Pritts [2]

2023

[1]Faculty of Computer Science, Ukrainian Catholic University, Lviv, Ukraine
[2]Czech Institute of Informatics, Robotics and Cybernetics at Czech Technical University in Prague, Czech

Structure of the presentation

Introduction

Background

Approach

Experiments

Conclusions

Introduction

Camera calibration

Camera calibration is a necessary step for mapping from the pixel position to the real-world 3D position.

Note

Here, we will use the term *camera calibration* to refer to the geometric camera calibration.

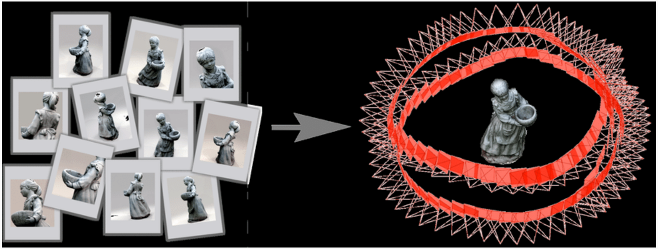


Figure 1: 3D reconstruction

Motivation

Accurate camera calibration is required for many applications, such as:

- 3D reconstruction
- Robotics and automation
- Augmented reality
- Photogrammetry
- Stereo vision
- And many more...

Challenges

Camera calibration is a challenging task, especially for highly distorted images.

The challenges include:

- Robust keypoint detection
- Accurate distortion modeling
- Stability of the calibration

Example

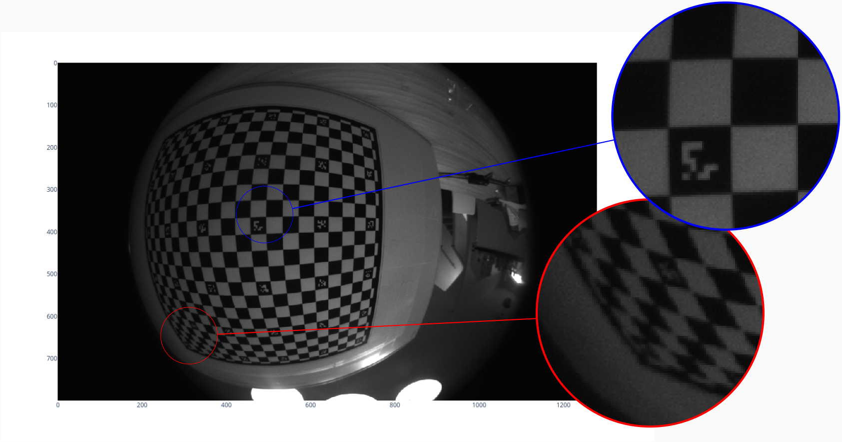


Figure 2: Example of corners near the center of the image and at the edge

Research objective

Improve the detection of calibration board fiducials from calibration imagery taken by wide-angle or fisheye lenses.

For that, we formulate the set of research questions:

- How to find additional features on the calibration board which were not detected by the feature detector?
- How to filter out falsely detected features?
- Is there a need for finding additional features on the calibration board? Are all of the points detected?

Background

Notation

The column vectors will be denoted by bold lowercase letters (e.g. $\mathbf{u} = \begin{pmatrix} u, v, 1 \end{pmatrix}^T$), matrices will be denoted by bold uppercase letters (e.g. $H = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$). We will use homogeneous coordinates to simplify the equations.

Camera model

Camera parameters can be divided into 3 parts:

- Extrinsic parameters
- Distortion parameters
- Intrinsic parameters

They define a camera model, which projects a homogeneous 3D scene point $\mathbf{x} = \left(x, y, z, 1\right)^T$ into the homogeneous image point $\mathbf{u} = \left(u, v, 1\right)^T$.

Definition of the camera model

Definition

$$\alpha \mathbf{u} = K f_{\boldsymbol{\lambda}}(H\mathbf{X}) \qquad \text{(Projection)}$$

$$\alpha \mathbf{X} = H^{-1} g_{\boldsymbol{\lambda}}(K^{-1}\mathbf{u}). \qquad \text{(Back projection)}$$

where K is the camera matrix, $g_{\boldsymbol{\lambda}}(\cdot)$ is the division distortion model (Fitzgibbon, 2001), $f(\cdot)$ is the inverse of $g(\cdot)$ and H is the homography matrix, and α is a non-zero scalar.

Note

Slides with the detailed explanation can be demonstrated upon request during the Q&A session.

Approach

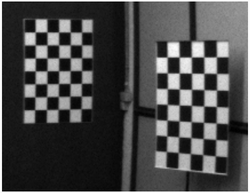
Pipeline overview

1. Feature detection.
2. Camera calibration.
   2.1 Initialize camera parameters $(R, \mathbf{t}, \boldsymbol{\lambda})$ using the solver.
   2.2 Refine camera parameters and estimate K by optimization.
3. Impute the gaps in the board, and extend it.
4. Get positions of new points on the image.
5. Filter out false positives.

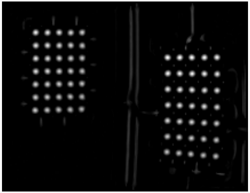
Feature detection

Use the approach, proposed by Geiger et al., 2012:

1. Compute the corner likelihood map by convolving the image with two $n \times n$ prototypes.
2. Additional filtering based on the number of the zero-crossings and non-maximum suppression.
3. Subpixel refinement of the detected corners.
4. Board's structure refinement.



(a) Input image



(b) Corner likelihood

1. Initialize camera parameters $(R, \mathbf{t}, \boldsymbol{\lambda})$ using the method, proposed by Scaramuzza, Martinelli, and Siegwart, 2006.

Overview

The author proposes the multistage solver for the projection equation, assuming that the camera matrix is known.

Camera calibration 2

1. Refine the values of R, $\mathbf{t}$, $\boldsymbol{\lambda}$, and estimate K by minimizing the reprojection error between the board and the back-projected corners.

Reprojection error

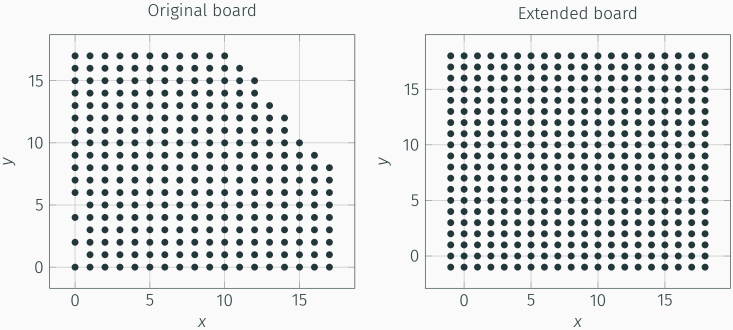
The reprojection error is the distance between the reprojected point and the measured one:

$$L = \sum_{i=1}^{N} \left\| H^{-1} g_{\boldsymbol{\lambda}}(K^{-1}\mathbf{u_i}) - \mathbf{x_i} \right\|^2 ,$$

where $\boldsymbol{\lambda}$ are the division distortion model parameters, $\mathbf{x_i}$ and $\mathbf{u_i}$ are the coordinates of the i-th corner in the 3D scene coordinates, and the respective feature on the image.

Additional features detection

To find the probable positions of the previously undetected corners, we impute the gaps in the board, and extend it by 1 row and column from each side:



Binary classification

- Compute the corner likelihood map for the image.
- Use the ROC curve, and pick the threshold which maximizes the G-mean.

We tested the approach of Geiger et al., 2012, and, alternatively, the Hessian responses for the image, as proposed by Chen and Zhang, 2005.

Experiments

Metrics

The paper's main contribution is finding additional calibration boards' features, which then can be used as an input to any other camera calibration algorithm.

- Number of recovered artificially removed points
- Number of recovered points under occlusion
- Number of recovered points on original images

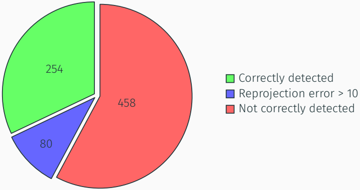
Dataset

OV (Lochman et al., 2021) is a dataset of approximately 1400 images. It was collected using eight stereo cameras. As a calibration pattern, the checkerboard pattern with 9×6 tags of 22 mm size was used.

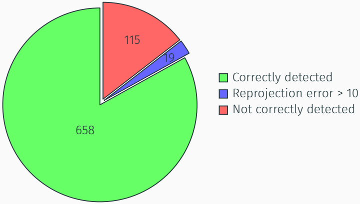
Note

Much more data was collected. However, the initial feature detection supports only checkerboard patterns for now. Other than that, the pipeline works with any pattern.

Compare the initial camera calibration with camera calibration obtained via optimization of reprojection error.

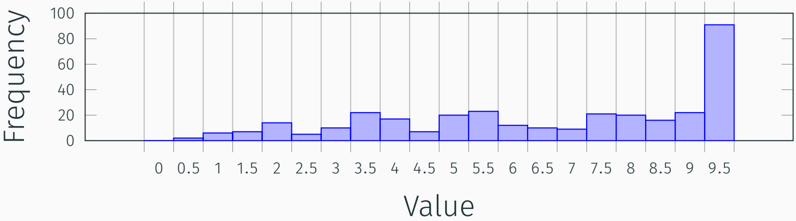


(a) Initial calibration

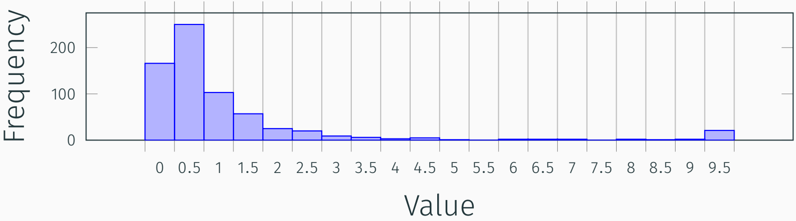


(b) Final calibration

Camera calibration 2



(a) Initial calibration's reprojection error histogram



(b) Final calibration's reprojection error histogram (in px.)

Additional features detection

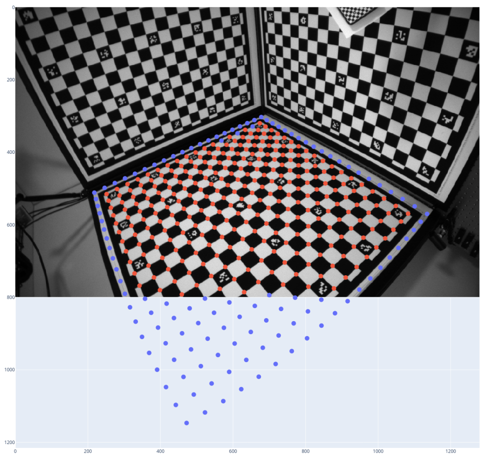
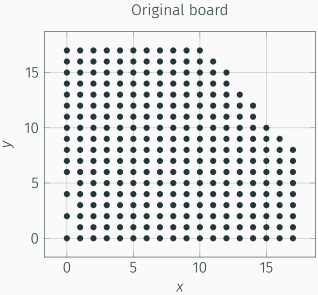
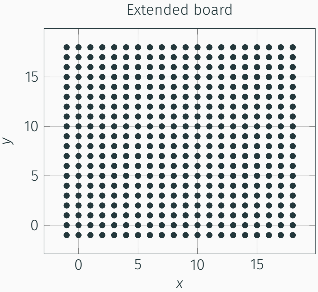


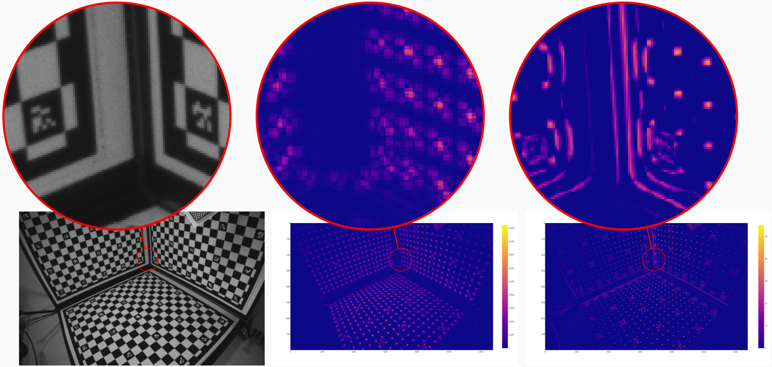
Figure 7: Extended board, new points are marked as blue





Classification

The Hessian approach proved to be more robust, as the other gave too many false positives, especially for the edges.

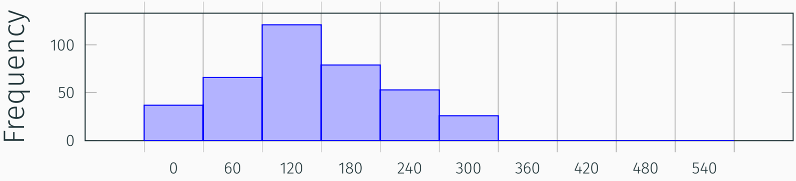


(a) Original image (b) Hessian response (c) Geiger et al., 2012

Evaluation (artificially removed points)

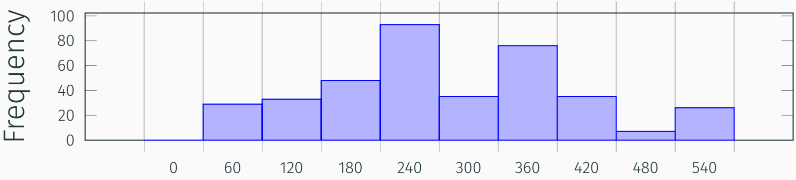
We removed 20% of the points from the original board and then tried to recover them.

Histogram of points before refinement



Value

Histogram of points after refinement



Evaluation (artificially removed points) 2

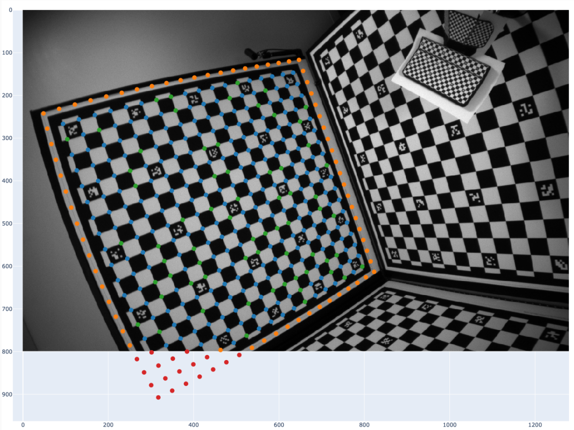
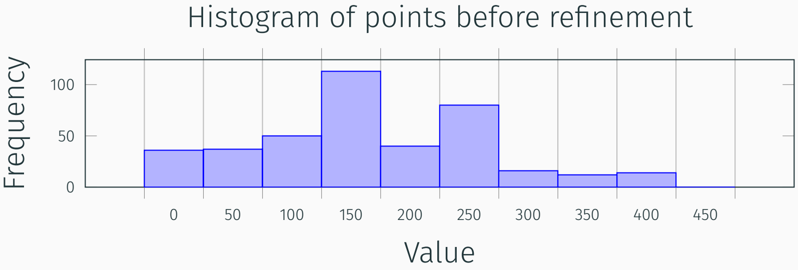


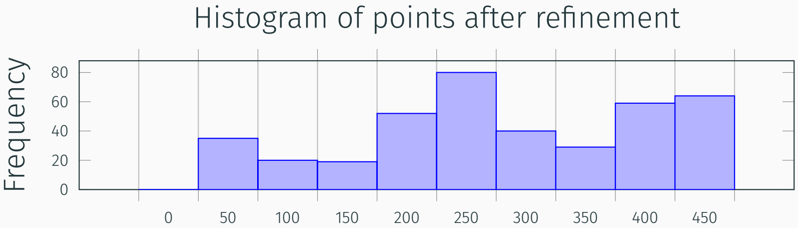
Figure 11: Recovered pruned points (unchanged filtered out new corner out of image)

Figure 12: Feature refinement on the board with 80% of the points

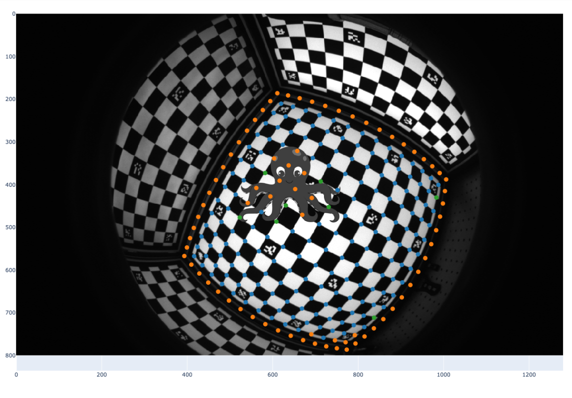
Evaluation (artificial occlusion)

Occlusions pose additional complications for feature detection.





Evaluation (artificial occlusion) 2

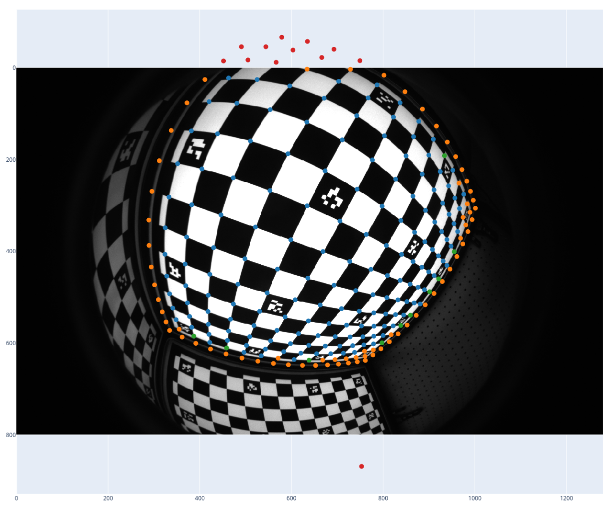


(a) Recovered pruned points (unchanged filtered out new corner out of image)

Figure 14: Feature refinement on the board with partial board occlusion

Evaluation (real data)

Lastly, we recovered the points that were not detected by the initial feature detector.



Evaluation (real data, false positives) 1

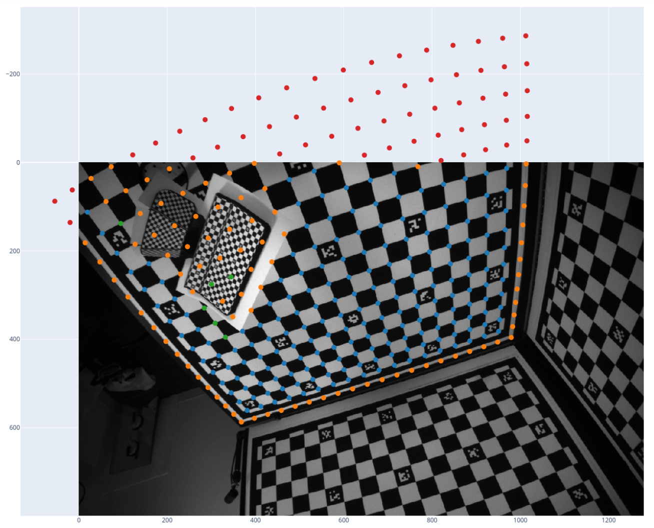
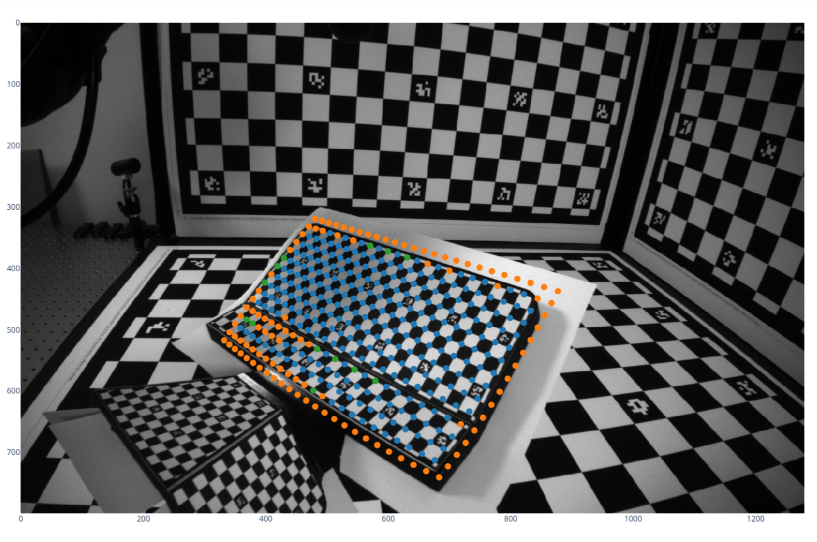


Figure 16: Occlusion on the board with distinct features

Evaluation (real data, false positives) 2



Evaluation (real data)

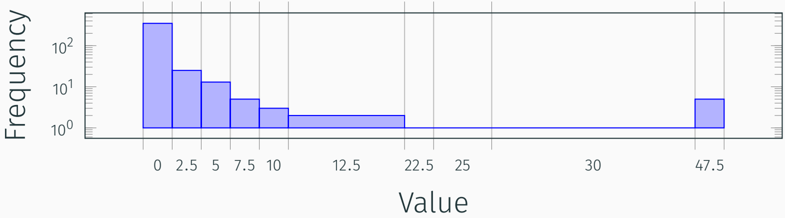


Figure 17: Histogram of the newly recovered features

Conclusions

Reviewers' comments

No comparison was made to classical camera calibration results and no discussion why results differ.

The main contribution is the feature detection step. The user can use the found camera parameters, or pass the obtained features to any camera calibration toolchain.

Q&A

Extrinsic parameters

The extrinsic parameters represent a rigid transformation from a 3-D world coordinate system to the 3-D camera's coordinate system.

Definition

$$\widehat{\mathbf{x}} = R\left(x, y, z\right)^T + \mathbf{t} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \left(x, y, z, 1\right)^T,$$

where $\left(x, y, z\right)^T$ is a 3D scene point, R is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector.

Extrinsic parameters

When working with the coplanar scene points, we can simplify the projection by assuming that the scene plane is located at $Z = 0$. In this case, the projection of the point becomes:

Definition for coplanar points

$$\alpha \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}}_{H} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Distortion model

The distortion of the image is caused by the lens not being perfectly planar. We used the division distortion model Fitzgibbon, 2001 which maps a point from a retinal plane to the ray direction in the camera coordinate system.

Definition

$$g(\mathbf{u}) = \Big(u, v, \psi(r(\mathbf{u}))\Big)^T, \psi(r) = 1 + \sum_{n=1}^{N} \lambda_n r^{2n},$$

where $\mathbf{u} = \Big(u, v, 1\Big)^T$ is a point in the retinal plane , $r(\mathbf{u}) = \sqrt{u^2 + v^2}$ is the radial distance from the principal point and λ_n are the distortion coefficients.

Intrinsic parameters

Represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates.

Definition

$$K = \begin{bmatrix} \alpha_x & \alpha_x \cot\theta & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & k & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}.$$

For a typical camera, $\theta = \pi/2$ and $\alpha_x = \alpha_y$ Hartley and Zisserman, 2004:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}.$$