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

Introduction

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.

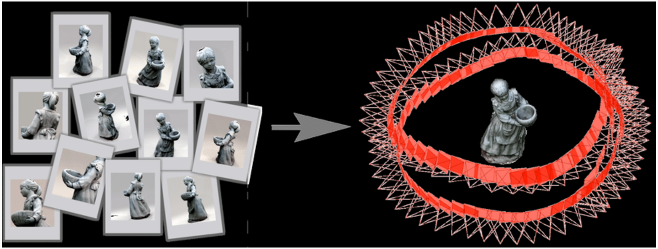


Figure 1: 3D reconstruction

Motivation

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

- 3D reconstruction
- Robotics and automation
- Augmented reality
- Photogrammetry
- Stereo vision

Figure 2: Image with a high distortion

Example

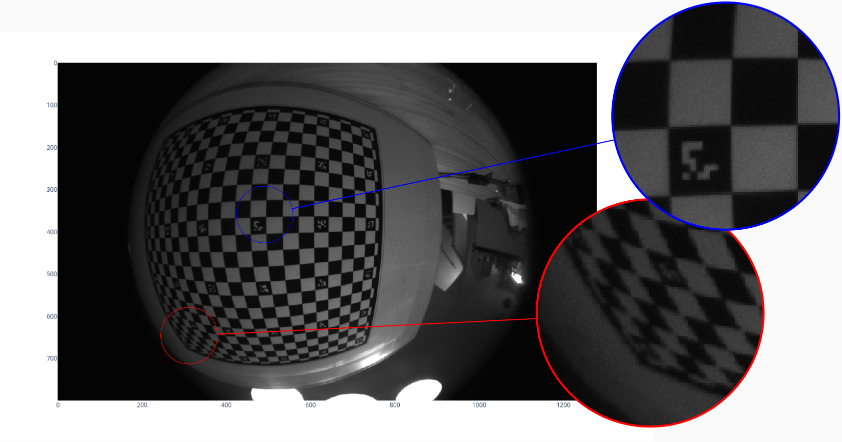


Figure 3: Corners near the center of the image and at the edge

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?

Related work

TartanCalib[1]

Algorithm

1. Run the calibration toolchain
2. Undistort the image
3. Subpixel refinement
4. Repeat

Issues

- Runs complete calibration toolchain each iteration
- Does not use the geometry information
- Not self-contained

[1] Duisterhof et al., 2022.

Models comparison

Supports	Name				
	OpenCV[2]	TartanKalib[3]	Kalibr[4]	OCamCalib[5]	LIBCBDETECT[6]
Unknown board shape	✗	✓	✓	✓	✓
Occlusion	✗	✓	✓	✓	✓
Model-based extra features detection	✗	✓	✗	✗	✗
Geometry-based extra features detection	✗	✗	✗	✗	✗

Table 1: Comparison of feature detectors

[2] Duda and Frese, 2018.
[3] Duisterhof et al., 2022.
[4] Maye, Furgale, and Siegwart, 2013.
[5] Scaramuzza, Martinelli, and Siegwart, 2006.
[6] Geiger et al., 2012.

Approach

Intuition (image)

Can you guess where the missing corners could be?

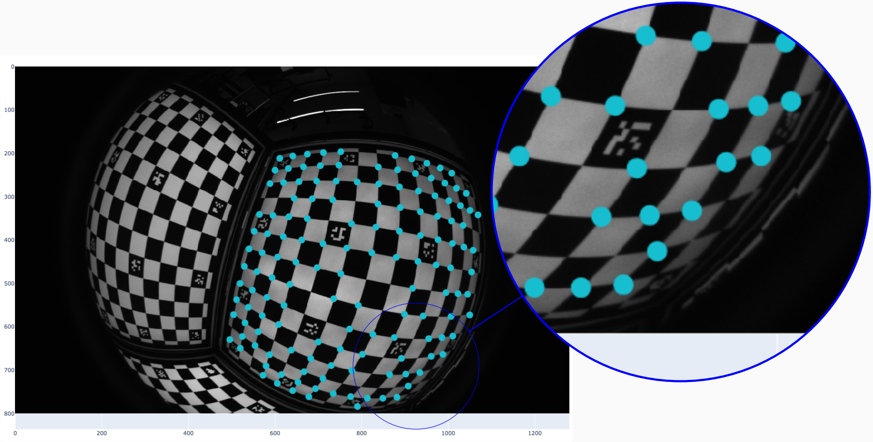
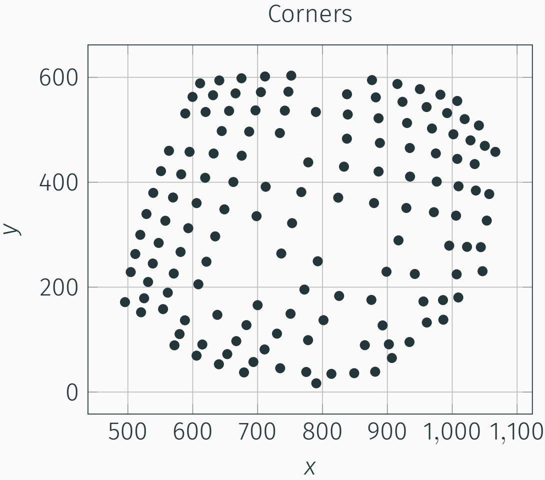


Figure 4: Example board detection

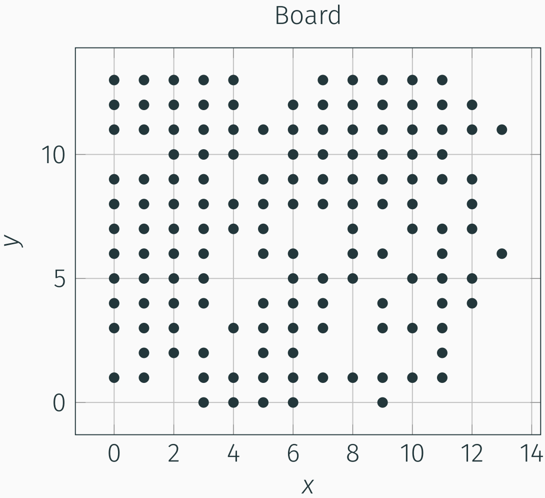
Intuition (corners)

Can you now?



Intuition (board)

What about here?



Use the board’s geometry to find missing corners!

Method

Pipeline overview

1. Feature detection.
2. Camera calibration.
    2.1 Initialize camera parameters using the solver.
    2.2 Refine camera parameters 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.

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 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

$$\alpha\mathbf{u} = Kf_{\boldsymbol{\lambda}}(H\mathbf{x}) \qquad \text{(Projection)}$$

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

where K is the camera matrix, $g_{\boldsymbol{\lambda}}(\cdot)$ is the division distortion model[7], $f(\cdot)$ is the inverse of $g(\cdot)$ and H is the homography matrix, and α, β are non-zero scalars.

Note

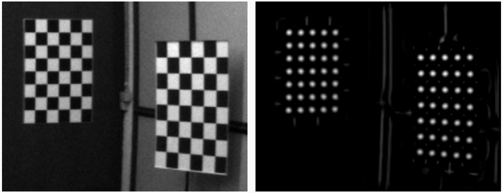
You have printed materials with the details about the camera model.

[7] Fitzgibbon, 2001.

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

Camera calibration 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

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

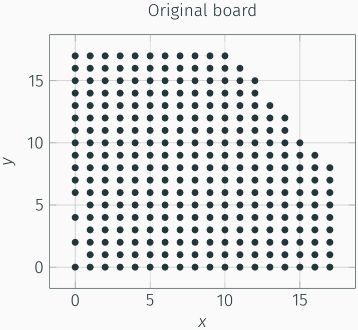
The reprojection error is the distance between the reprojected point and the measured one:

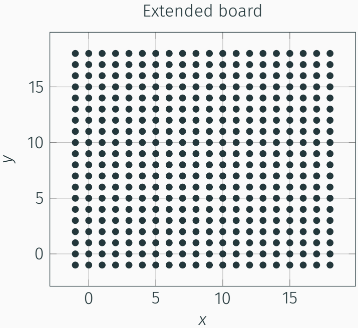
$$L = \sum_{i=1}^{N} \left\| H^{-1} g_{\boldsymbol{\lambda}}(K^{-1}\mathbf{u_i}) - \mathbf{x_i} \right\|^2 ,$$

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

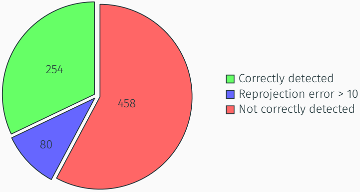
Dataset

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.

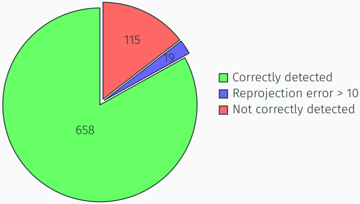
Note

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 the camera calibration obtained via optimization of reprojection error.

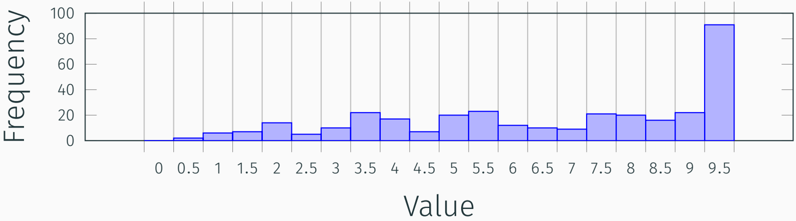


(a) Initial calibration

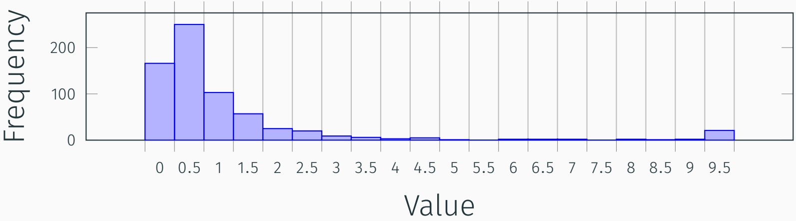


(b) Final calibration

Camera calibration 2



(a) Initial calibration's reprojection error histogram



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

Additional features detection

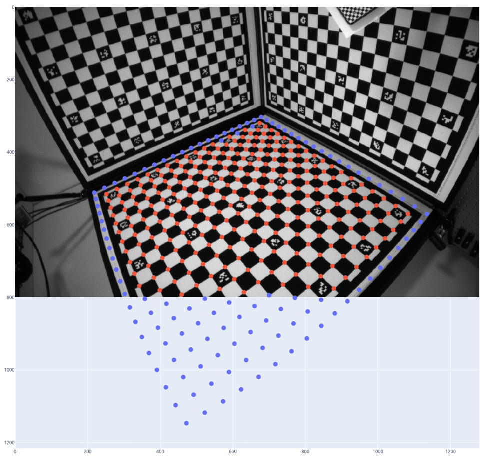
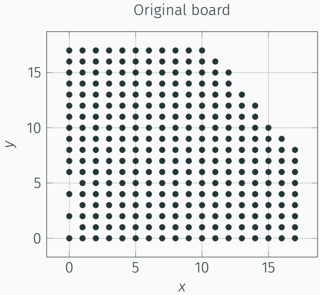
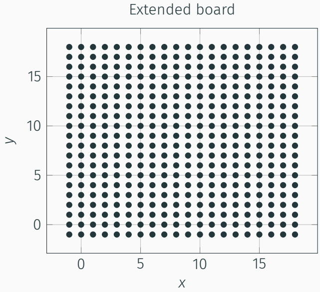
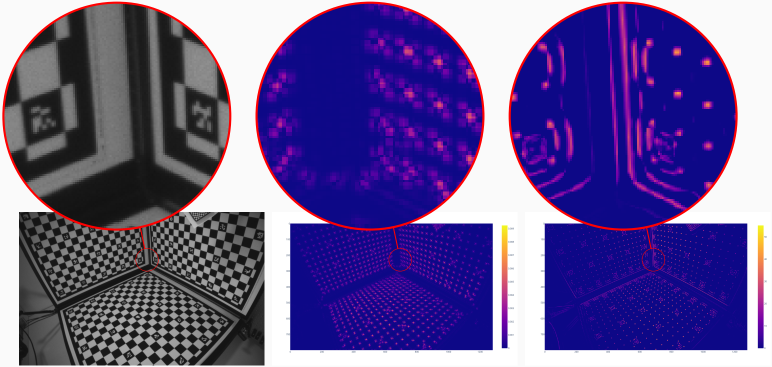


Figure 9: Extended board, new points are marked as blue





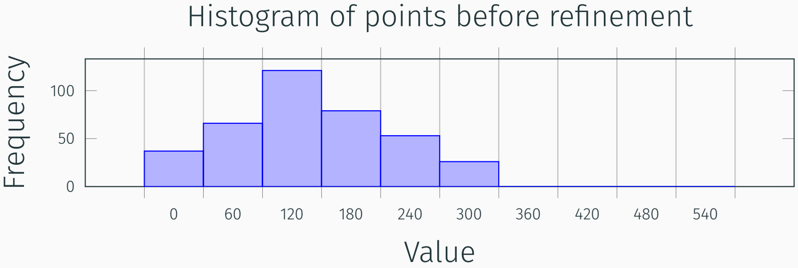
The Hessian approach proved to be more robust, as the alternative gave too many false positives, especially for the edges.

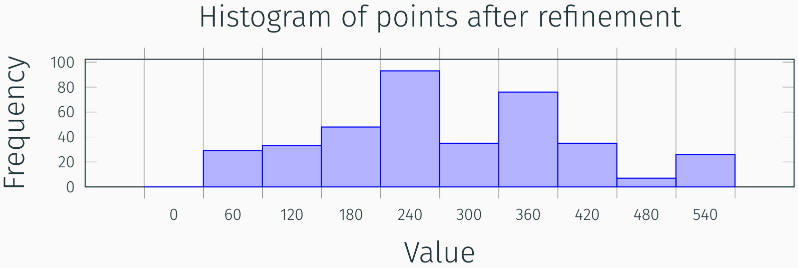


(a) Original image (b) Hessian response (c) Geiger et al., 2012

Evaluation (artificially removed points)

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





Evaluation (artificially removed points) 2

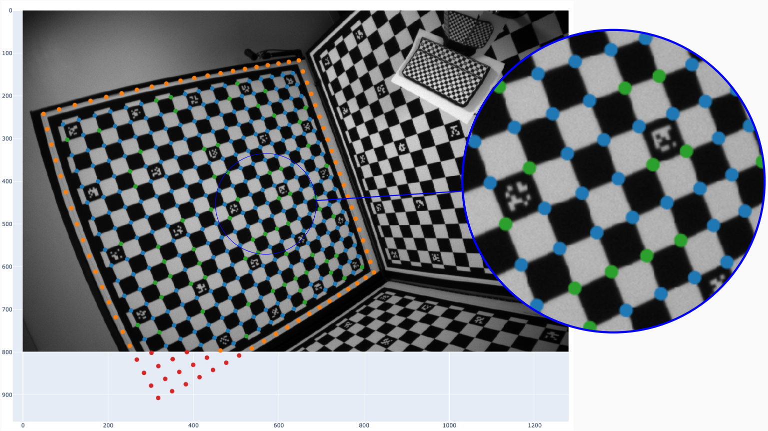
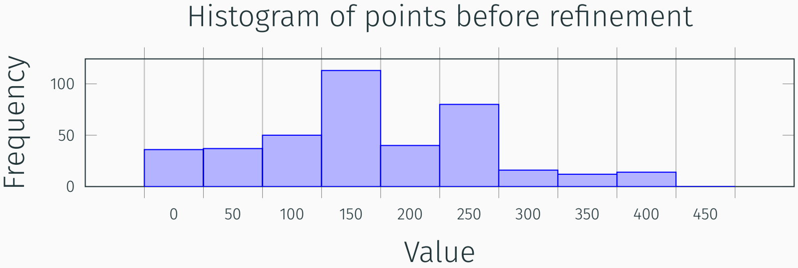
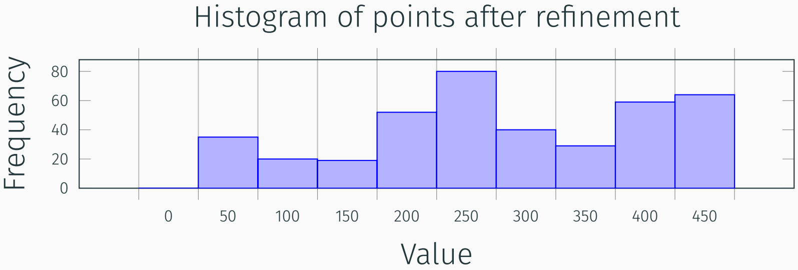


Figure 13: Feature recovery on the board with 80% of the points (unchanged filtered out new corner out of image)

Evaluation (artificial occlusion)

Occlusions pose complications for feature detection.





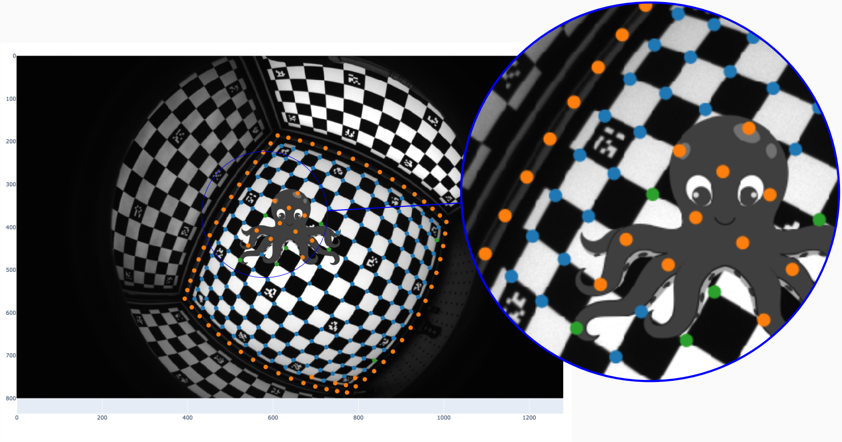


Figure 15: Additional features detection on the board with partial board occlusion (unchanged filtered out new corner out of image)

Evaluation (real data)

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

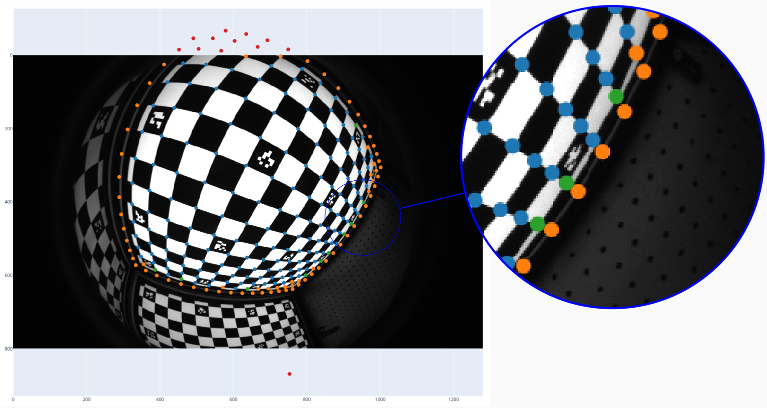


Figure 16: Newly detected points (unchanged filtered out new corner out of image)

Evaluation (real data) 2

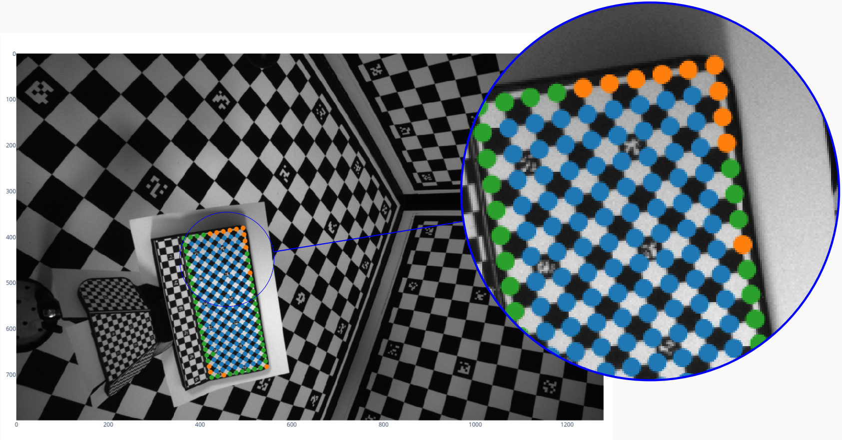


Figure 17: Newly detected points (unchanged filtered out new corner out of image)

Evaluation (real data, false positives)

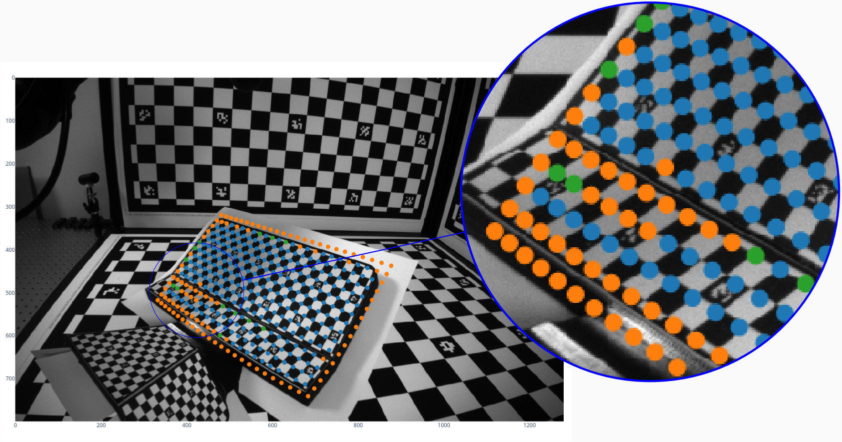


Figure 18: Example of false positives (unchanged filtered out new corner out of image)

Evaluation (real data)

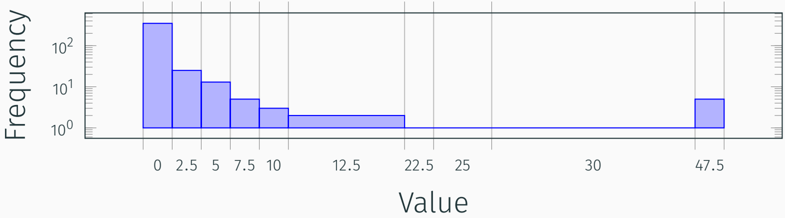


Figure 19: Histogram of the newly recovered features

Comparison

Supports	Name					
	OpenCV[8]	TartanKalib[9]	Kalibr[10]	OCamCalib[11]	LIBCBDETECT[12]	Ours
Unknown board shape	✗	✓	✓	✓	✓	✓
Occlusion	✗	✓	✓	✓	✓	✓
Model-based extra features detection	✗	✓	✗	✗	✗	✓
Geometry-based extra features detection	✗	✗	✗	✗	✗	✓

Table 2: Comparison of feature detectors

[8] Duda and Frese, 2018.
[9] Duisterhof et al., 2022.
[10] Maye, Furgale, and Siegwart, 2013.
[11] Scaramuzza, Martinelli, and Siegwart, 2006.
[12] Geiger et al., 2012.

Conclusions

Answers to the research questions

- **How to find additional features on the calibration board which were not detected by the feature detector?**
  This thesis proposes a model-based approach to improve the detection of calibration board fiducials from calibration imagery taken by wide-angle or fisheye lenses.
- **How to filter out falsely detected features?**
  We proposed a classifier that filters false positives, and shown that it works well on real data.
- **Is there a need for finding additional features on the calibration board? Are all of the points detected?**
  We demonstrated that the proposed method successfully detected additional features on the real data.

Reviewers' comments 1

No comparison was made to classical camera calibration results

The main contribution is the feature detection step.

The algorithm estimates camera parameters, but we don't claim that they are better than state-of-the-art. The user can use those, or pass the refined features into any camera calibration toolchain.

The literature review does not include a review of methods that solve the feature detection step in a different way

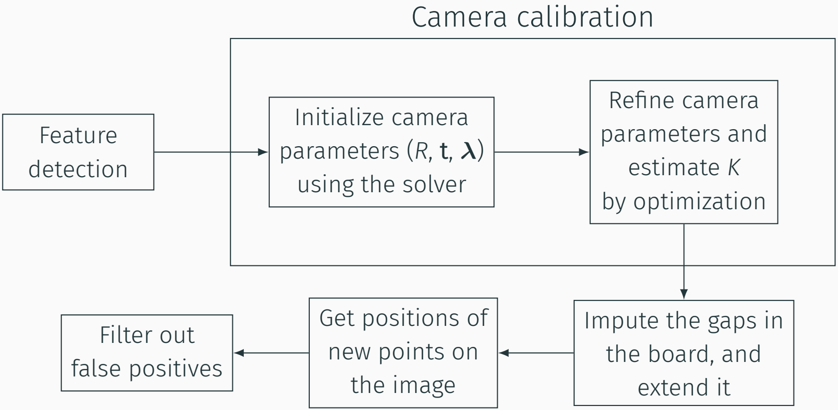
A review of feature detection methods is presented in the paper.

The specific subtask of model-based missed feature detection is not big enough to have a separate review. However, we included TartanKalib[13].

[13] Duisterhof et al., 2022.

Reviewers' comments 3

There is no general block diagram of the algorithm



Reviewers' comments 4

It is not sufficiently described what results are obtained on test sets

The datasets we used did not contain the ground truth camera calibration. Also, the algorithm does not include the learning step, hence there is no reason to use the test set.

Reviewers' comments 4

The questions "How to filter incorrectly defined grid points" and "Is there a need to search for additional grid points" posed in the work are insufficiently disclosed

Addressed in section Conclusions.

A comparison of the results with other feature detection methods has not been made

We have added a feature-wise comparison to the state-of-the-art methods, and numerical comparison to the LIBCBDETECT[14] method.

The TartanKalib[15] method currently supports only AprilGrid boards.

[14] Geiger et al., 2012.
[15] Duisterhof et al., 2022.

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:

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