

# BabelCalib: A Universal Approach to Calibrating Central Cameras

## Supplemental Material

### A. Aspect Ratio Solver

The epipolar constraint  $\mathbf{u}^\top \mathbf{F} \mathbf{x} = 0$  is invariant to a projective change of coordinates. This implies that the aspect ratio cannot be recovered using only the constraints available from the radial fundamental matrix (e.g., see (13)). Additional constraints can be obtained by enforcing the concurrence of back-projected corners with their corresponding scene points using (1). The constraints given by (1) generate polynomial equations of the unknown aspect ratio, remaining unknowns of  $\mathbf{H}$ , and the unknown division model parameters  $\{a, h_{31}, h_{32}, h_{33}, \lambda_1, \dots, \lambda_n\}$ . The formulation introduces  $N + 1$  unknowns, but the formulation is invariant to the scale of  $\mathbf{H}$ . The minimal solution requires  $N/2 + 2$  image-to-target correspondences.

Let  $\mathbf{u}'$  be the projection center-subtracted image point  $\mathbf{u}' = \mathbf{T}(-\mathbf{e})\mathbf{u}$ . Then we can rewrite (1) as following

$$\begin{pmatrix} u'/f \\ v'/f \\ \psi(r(\text{diag}(a^{-1}, 1, 1)\mathbf{u}')) \end{pmatrix} \times \underbrace{\text{diag}(a, 1, 1)}_{\hat{\mathbf{H}}} \mathbf{H} \mathbf{x} = \mathbf{0}. \quad (21)$$

Solving (10) and (11) gives the radial fundamental matrix  $\mathbf{F}$  and projection center  $\mathbf{e}$ , respectively. We use the relation  $\mathbf{F} \sim [\mathbf{e}]_{\times} (a\mathbf{h}_1^\top \mathbf{x}, \mathbf{h}_2^\top \mathbf{x}, 0)^\top$  to determine the first two rows of the matrix  $\hat{\mathbf{H}}$ , up to scale,  $\hat{\mathbf{h}}_1^\top = (f_{21}, f_{22}, f_{23})$  and  $\hat{\mathbf{h}}_2^\top = -(f_{11}, f_{12}, f_{13})$ .

Rewriting (21) in terms of the unknowns gives

$$\begin{pmatrix} u' \\ v' \\ f \left( 1 + \sum_{n=1}^N \lambda_n r_0(a)^{2n} \right) \end{pmatrix} \times \begin{pmatrix} x' \\ y' \\ \hat{\mathbf{h}}_3^\top \mathbf{x} \end{pmatrix} = \mathbf{0}, \quad (22)$$

where  $r_0(a) = \sqrt{(\frac{u'}{a})^2 + (v')^2}$ ,  $x' = \hat{\mathbf{h}}_1^\top \mathbf{x}$ ,  $y' = \hat{\mathbf{h}}_2^\top \mathbf{x}$ . After reparameterizing with  $b = a^2$ , it can be seen that for  $N = 1$ , (22) is linear in the unknowns and for  $N = 2$ , a polynomial system of degree three must be solved.

We tested the solvers for the cases  $N = 1$  and  $N = 2$  on synthetic data and found that they are sensitive to noise. With a sufficiently good guess on aspect ratio, the solver in (17) performs better than the aspect ratio solver (see Fig. C.1). We opted to sample over aspect ratio rather than introduce these solvers. We leave the incorporation of minimal solvers for unknown aspect ratio for future work.

### B. Recovering Radial-Projection Functions for User-Selected Camera Models

The detected corners  $\mathbf{u}$  can be back-projected to rays  $\gamma g(\mathbf{u}) = \gamma (u, v, \psi(r(\mathbf{u})))^\top$ ,  $\gamma > 0$  in the direction of the board fiducials in the camera's coordinate system. The polar angle of the ray determines how the points of the ray are projected, so any point of the ray can be used (equivalently, any  $\gamma > 0$  in (1) can be used). The distances to the optical axis  $R$  and principal plane  $Z$  are computed from the unit vector concurrent with the ray,  $\frac{g(\mathbf{u})}{\|g(\mathbf{u})\|}$ . With the radii and depths recovered, linear least squares can be used to recover the unknowns of the radial projection functions listed in Tables 1, with the exception of the Double Sphere (DS) model. This demonstrates the ease with which BabelCalib is extended.

**Division Model to Kannala-Brandt Regression** The experiments of Sec. 5 confirm that the Kannala-Brandt (KB) model is the most flexible and accurate over the largest range of lenses, which is consistent with the results of [41]. We also found that the Kannala-Brandt model is also effective for catadioptric rigs (see Table 5).

The number of failed calibrations of OpenCV and Kalibr reported in Table 5 suggests that Kannala-Brandt is one of the hardest camera models to initialize. Directly computing a model proposal for Kannala-Brandt is difficult. The displacement of the projection from the optical center is proportional to a polynomial function of  $\text{atan2}$ . The trigonometric function  $\text{atan2}$  is not easily eliminated since the unknown depth  $Z$  is unique to each fiducial on the chessboard. Thus the problem cannot be solved as a polynomial system. BabelCalib initializes Kannala-Brandt by linearly regressing its parameters against the estimated division model, which is formulated in Sec. 3. We derive the linear system here to show how easy the model-to-model regression is once the back-projection function is known.

We back-project the corners  $\mathbf{u}_i = (u_i, v_i, 1)^\top$  to ray directions  $\mathbf{x}'_i = (x'_i, y'_i, z'_i)^\top$  where  $\mathbf{x}'_i = g(\mathbf{K}^{-1}\mathbf{u}_i)$ . The radii and depths are computed as  $R_i = \sqrt{x'^2_i + y'^2_i}$  and  $Z_i = z'_i$ , respectively. The polar angles  $\omega_i = \text{atan2}(R_i, Z_i)$  are determined, and the unknown coefficients of the polynomial  $r_i = \omega_i + \sum_{n=1}^4 k_n \omega_i^{2n+1}$  can be determined linearly,

$$\begin{bmatrix} \omega_i^3 & \dots & \omega_i^9 \\ \omega_{i+1}^3 & \dots & \omega_{i+1}^9 \\ \vdots & & \vdots \end{bmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix} = \begin{pmatrix} r_i - \omega_i \\ r_{i+1} - \omega_{i+1} \\ \vdots \end{pmatrix}. \quad (23)$$

Solving (23) fully specifies the Kannala-Brandt projection model since the center of projection  $\mathbf{e}$  and the focal length  $f$  are known from estimation of the back-projection model — from (11) and (17), respectively.

### C. Algorithm

To summarize the proposed RANSAC-based calibration framework detailed in Sec. 4, we provide Algorithm 1 with all the crucial steps of the BabelCalib framework. The details on the pose optimization are omitted in the algorithm.

---

#### Algorithm 1: BabelCalib Camera Calibration

---

**Input:** 2D-3D point correspondences  $\mathbf{u}_{ijk} \leftrightarrow \mathbf{X}_{ij}$   
**Parameters:** Radial projection model  $\phi_\theta$   
**Output:**  $\Theta^* = \{\theta^*, \mathbf{K}^*, \mathbf{R}_{jk}^*, \mathbf{t}_{jk}^*\}$   
 $\mathcal{J}_0^* \leftarrow \infty, \mathcal{J}^* \leftarrow \infty$   
**repeat**  
  Sample image  $k'$  and plane  $j'$   
  Sample  $\{\mathbf{u}_{ij'k'} \leftrightarrow \mathbf{X}_{ij'}\}_{i=1}^N$   
  Compute  $\mathbf{e}$  and  $\mathbf{F}$  from  $\{\mathbf{u}_{ij'k'} \leftrightarrow \mathbf{X}_{ij'}\}_{i=1}^N$  with (15)  
  Correct corners  $\mathbf{u}_{ij'k'}$  to  $\mathbf{u}_{ij'k'}^*$  with (14)  
  Compute  $\{\mathbf{R}_{j'k'}, t_x, t_y\}$  from  $\mathbf{F}$  with (13)  
  Sample aspect ratio  $a$  from  $[0.5, 2]$   
  Compute  $\{f, \lambda_1, \dots, \lambda_N, t_z\}$  with (17)  
   $\mathbf{K} \leftarrow \mathbf{T}(\mathbf{e}) \text{diag}(af, f, 1), \mathbf{t}_{j'k'} \leftarrow (t_x, t_y, t_z)^\top$   
  **if**  $\phi_\theta^{-1}$  is not the division model **then**  
    Regress  $\theta$  of  $\phi_\theta$  with (18)  
  **end**  
  **for**  $(k, j) \neq (k', j')$  **do**  
    Sample image  $k$  and plane  $j$   
    Back-project the corners  $\mathbf{x}'_{ijk} \leftarrow g(\mathbf{K}^{-1} \mathbf{u}_{ijk})$   
    Compute  $\{\mathbf{R}_{jk}, \mathbf{t}_{jk}\}$  from  $\{\mathbf{x}'_{ijk}\}_i$  with [29]  
  **end**  
   $\Theta_0 \leftarrow \{\theta, \mathbf{K}, \mathbf{R}_{jk}, \mathbf{t}_{jk}\}$   
  Compute loss  $\mathcal{J}_0 \leftarrow \mathcal{J}(\Theta_0)$  with (19)  
  **if**  $\mathcal{J}_0 < \mathcal{J}_0^*$  **then**  
     $\mathcal{J}_0^* \leftarrow \mathcal{J}_0, \mathcal{I}_0^* \leftarrow \mathcal{I}_0$   
    Optimize  $\Theta_{LO} \leftarrow \text{argmin}_\Theta \mathcal{J}(\Theta)$  with (19)  
    Compute loss  $\mathcal{J}_{LO} \leftarrow \mathcal{J}(\Theta_{LO})$  with (19)  
    **if**  $\mathcal{J}_{LO} < \mathcal{J}^*$  **then**  
       $\mathcal{J}^* \leftarrow \mathcal{J}_{LO}, \Theta^* \leftarrow \Theta_{LO}$   
    **end**  
  **end**  
**until**  $T$  iterations;  
Compute inlier ratio  $\mathcal{I}^* \leftarrow \mathcal{I}(\Theta^*)$  with (20)

---

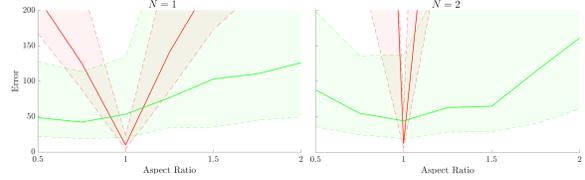


Figure C.1: **Noise sensitivity.** RMS reprojection error on synthetic images with added 1 px noise for (left) the solvers for the division model complexity  $N = 1$ , and (right) the solvers for the complexity  $N = 2$ . Red curves correspond to the linear solvers from (17) that do not solve for the aspect ratio, and green curves correspond to the aspect ratio solvers from (22). Solid curves are the median errors and the shaded plots are the corresponding interquartile ranges.

### D. Calibration with Limited Data

Calibration accuracy is dependent on good feature coverage. However, fast calibration from few images is also im-

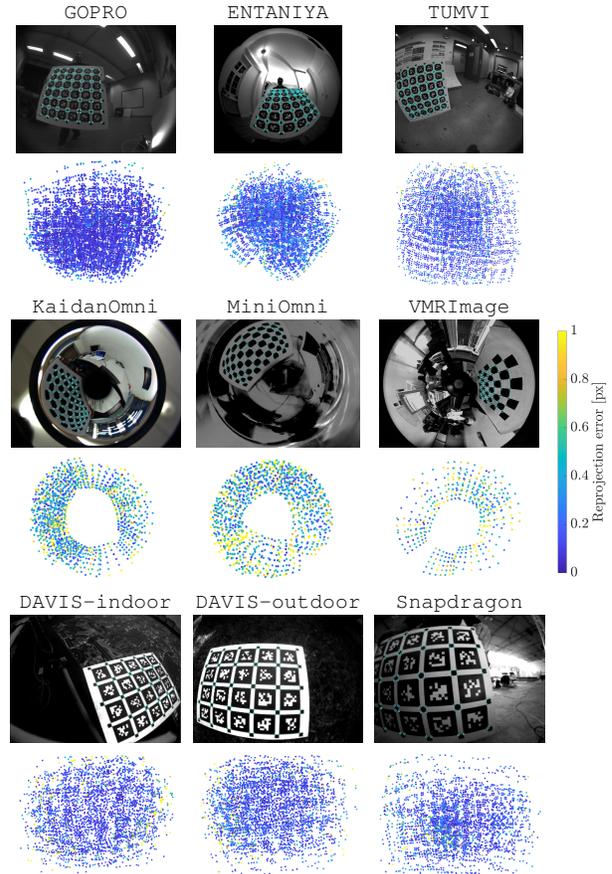


Figure C.2: **Kalibr, OCamCalib and UZH data and calibration results.** (rows 1,3,5) example images with detected (red crosses) and reprojected (cyan circles) corners; (rows 2,4,6) all corners from the camera subset color-coded corresponding to their reprojection errors.

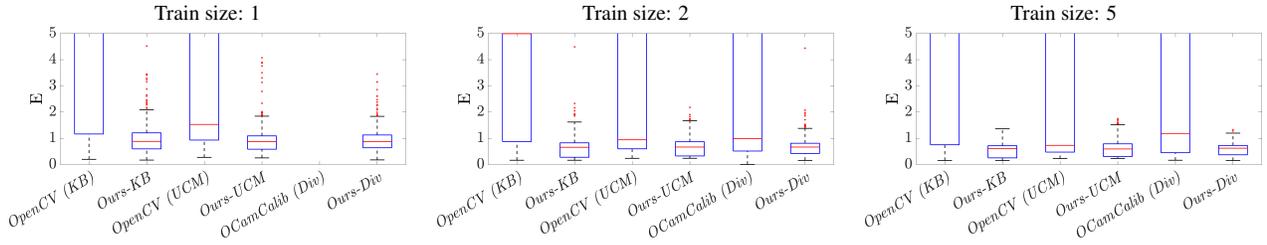


Figure C.3: **Calibration results for limited train data.** Cross-validation with corresponding train size is performed, and the reported errors are normalized in accordance with image resolution  $1000 \times 1000$  px. E denotes the RMS weighted reprojection error. OCamCalib-DIV requires at least two images for calibration.

portant for users e.g. for lens inventory purposes. We compared the performance of calibration methods on a limited amount of training data starting from a single image. We randomly drew the subsamples of sizes 1, 2, and 5 images from the training data (10 times for each train size) and evaluated the calibrations on the hold-out test data. The Fig. C.3 reports the distributions of the robust test errors (19). To remove the effect of the image resolution for different cameras, the errors are normalized in accordance with the resolution  $1000 \times 1000$  px. There are several limitations found in current frameworks. OCamCalib-DIV requires at least two images for calibration so they don't have the results for the train size 1. Also, as was mentioned in previous section, this toolbox requires all points to be visible from all views which is a big limitation. All other methods would occasionally fail for particular subsets of images. For such failure cases we set the test error to be the maximum error. The proposed calibration method never fails and provides the best calibration starting from a single image already.