Learning-Based Distortion Correction and Feature Detection for High Precision and Robust Camera Calibration

Yesheng Zhang, Xu Zhao, *Member, IEEE*, and Dahong Qian, *Senior Member, IEEE*

***Abstract*—Camera calibration is a crucial technique which significantly influences the performance of many robotic systems. Robustness and high precision have always been the pursuit of diverse calibration methods. State-of-the-art calibration techniques, however, still suffer from inexact corner detection, radial lens distortion and unstable parameter estimation. Therefore, in this paper, we improve the precision and robustness of calibration by widening these bottlenecks. In particular, effective distortion correction is performed by a learning-based method. Then, accurate sub-pixel feature location is achieved by the combination of robust learning detection, exact refinement and complete post-processing. To obtain stable parameter estimation, an image-level RANSAC-based calibration procedure is proposed. Ultimately, we assemble these methods into a novel and practical calibration framework. Compared with state-of-art methods, experiment results on both real and synthetic datasets under noise, bad lighting and distortion manifest the better robustness and higher precision of the proposed framework.**

***Index Terms*—Calibration and identification, deep learning for visual perception, visual learning.**

I. Introduction

CAMERA calibration is crucial for many robotic applications [1]–[3]. Especially, in some industrial and medical applications [4], precision and robustness of camera calibration have significant impact on the overall performance.

The most widely-used camera calibration toolboxes [5], [6] are built based on Zhang's technique [7]. Usually chessboard images are captured to calculate camera parameters according to the established feature correspondences between 3D world and 2D images. This pipeline is flexible and easy to implement. Building a precise and robust calibration system, however, is still a challenging problem, mainly due to the following issues:

1) *Inexact detection:* Sub-pixel feature localization is hard to achieve, especially in scenario with noise and bad illumination.
2) *Radial distortion:* Severe radial lens distortion may result in calibration failure.
3) *Unstable estimation:* Purely algebraic optimization of reprojection error leads to sub-optimal and unstable calibration results.

For the first issue, the calibration task naturally demands high precision feature detection, e.g. sub-pixel level accuracy. Although the chessboard corner feature is relatively simple for detection, the accuracy of hand-crafted detectors [8]–[10] needs to be enhanced as various noise can easily change the original feature pattern. As the rapid advance of deep learning, the convolution neural network (CNN) is introduced [11]–[13] to detect chessboard corners. Based on massive data training and data augmentation, the CNN can learn better feature representation of chessboard corner than manual methods. Thus, the learning detectors [14], [15] are much more robust to noise than hand-crafted ones. However, as the CNN is not sensitive to feature location [16], sub-pixel corner coordinates are difficult to be derived directly from the CNN. Therefore, recent work [17] decouples the corner location and the learning feature by refining the peak of CNN's output heatmap to get sub-pixel location. We follow this learning heatmap refinement pipeline, but propose a novel detection method with better combination between feature learning and sub-pixel refinement. Specifically, our network is trained to generate standard Gaussian distribution for each corner. Thus our refinement is performed by Gaussian surface fitting algorithm to get accurate sub-pixel location. Moreover, thanks to the known distributions of detected corners, we tackle the wrong detection problem in [15] by distribution-aware outlier rejection. Considering the chessboard pattern's peculiarity, the collineation post-processing is applied to recover lost corners after rejection and obtain higher accuracy. Our detection pipeline is tightly coupled and attains decent results against various noise in experiments. Fig. 1.

The second issue is about lens distortion, which is a non-trivial problem in camera calibration, especially radial distortion [18]. Classical methods [7], [19], [20] estimate camera and distortion parameters simultaneously by iterative optimization based on detection results. However, ambiguity is introduced in this way

Manuscript received 23 February 2022; accepted 28 June 2022. Date of publication 20 July 2022; date of current version 4 August 2022. This letter was recommended for publication by Associate Editor H. Zhang and Editor L. Pallottino upon evaluation of the reviewers' comments. This work was supported in part by NSFC under Grants 62176156 and 61673269, and in part by the Shanghai Engineering Research Center of Intelligent Control and Management. *(Corresponding author: Xu Zhao; Dahong Qian.)*

Yesheng Zhang and Dahong Qian are with the School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Institute of Medical Robotics, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: preacher@sjtu.edu.cn; dahong.qian@sjtu.edu.cn).

Xu Zhao is with the The Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Institute of Medical Robotics, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhaoxu@sjtu.edu.cn).

The code is publicly available at https://github.com/Easonyesheng/CCS.

Digital Object Identifier 10.1109/LRA.2022.3192610

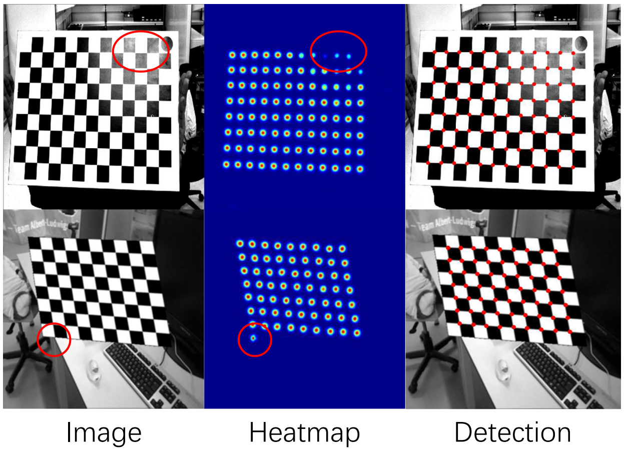

Fig. 1. Examples of detection outliers and detection results. The real image (up) and its native heatmap are related to lost corners and the synthetic ones (bottom) are fake corners. Both of them can be rejected according to abnormal distribution and achieve accurate detection after our post-processing.

as parameters are tangled together leading to failure under severe distortion. Moreover, this paradigm relies on detection accuracy, but distortion is detrimental to detection. On the other hand, the distortion is related to the curvature of the straight lines. This feature can be learned by deep network and utilized to correct distortion according to some fisheye image correction work [21]–[23]. As chessboard images naturally own many straight lines, we apply a compact network to infer the correction parameters from distorted images with a practical distortion model. As the detection suffers from radial distortion and our collineation post-processing assumes no distortion, so the distortion correction is performed first in our framework.

The third issue is caused by purely algebraic optimization aimed at re-projection error minimization, which may lead to unreasonable calibration results. The re-projection error is the residual between the 2D detection results and 3D projections through camera model. When the detection is accurate, the re-projection error minimization can obtain accurate camera model. However, the detection results always contain noise. Thus, the re-projection error may not directly relate to the accuracy of camera model. Based on our synthetic dataset, we conduct experiments to explore the relationship between re-projection error and camera model accuracy under noisy detection (Sec. IV-C). We find that the small re-projection error may not ensure the high accuracy of calibration, but the re-projection error consistency of all images can give a cue of precise camera model. Therefore, we propose an efficient image-level RANSAC algorithm based on re-projection error consistency to search for the optimal camera model and can improve the robustness of parameter estimation.

In sum, the critical components of a calibration system, distortion correction, feature detection and parameter estimation, are reforged and integrated as a novel and efficient calibration framework (Fig. 2). The contributions can be summarized as follows.

1) A novel learning-based camera calibration framework is proposed including radial distortion correction, sub-pixel feature detection and stable parameter estimation.
2) Our proposed feature detection method well combines the robustness of learning method with the precision of specially designed refinement and post-processing.
3) Massive experiment results manifest the high precision and robustness of our framework compared with state-of-art methods.

II. Related Work

Chessboard Corner Detection: Firstly, corner detectors such as [8], [9], [24] are adopted, but they are sensitive to noise. The widely-used detection function in OpenCV [5] refines coordinates according to gray distribution constraints. In [10], corner coordinates are refined based on the chessboard structure. While these methods achieve sub-pixel precision, heavily relying on hand-crafted features leads to the lack of robustness. Recently, some detection algorithms based on learning features are proposed [11]–[13]. In this methods, CNNs are trained on synthetic chessboard images to output the location of chessboard corners in image. Learning feature representation makes their methods robust against noise, but they are trapped in pixel level accuracy. Schroter *et al.* [15] propose a learning-based general point detection method. This method achieves sub-pixel accuracy, yet provides false negative results owning to global detection. Kang *et al.* [14] tackle this problem by parsing global context, but its method provide inaccurate results under difficult scenes as the sparse detection is achieved by non-maximum suppression. Chen *et al.* [17] propose a more accurate method by fitting CNN's response map. This pipeline is close to ours, but our learning heatmap method gets more tight connection between detection and refinement. Besides, we avoid unreliable detection results in [15] through the distribution-aware outlier rejection. Specific techniques named collineation post-processing considering the peculiarity of chessboard pattern are proposed to achieve higher precision.

Radial Distortion Correction: Classical algorithms integrate distortion in camera model and solve it by non-linear optimization techniques [7], [19], [20] using the detection results. Although this method works well under slight radial distortion, they may end up with a bad solution when the distortion is severe because of the inaccurate detection and parameter tangle. On the other hand, straight lines wrapped by distortion are appropriate features for CNNs to learn distortion parameters which is proved in some related fisheye image distortion correction work [21], [22], [25]. As chessboard images contain sufficient straight line features, we adopt a compact CNN to regress parameters of a specially selected distortion correction model. Different with previous work [25] with similar compact network, we adopt a more flexible distortion model which can choose suitable number of parameters based on different distortion levels.

Parameter Estimation: Widely-used Zhang's technique [7] first solves the initial guess of parameters based on the correspondence between the real world and image. Then these parameters are refined by minimizing the re-projection error. However, re-projection minimization can not ensure an accurate camera model as detection always contains noise. Besides, purely algebraic optimization is unstable leading to

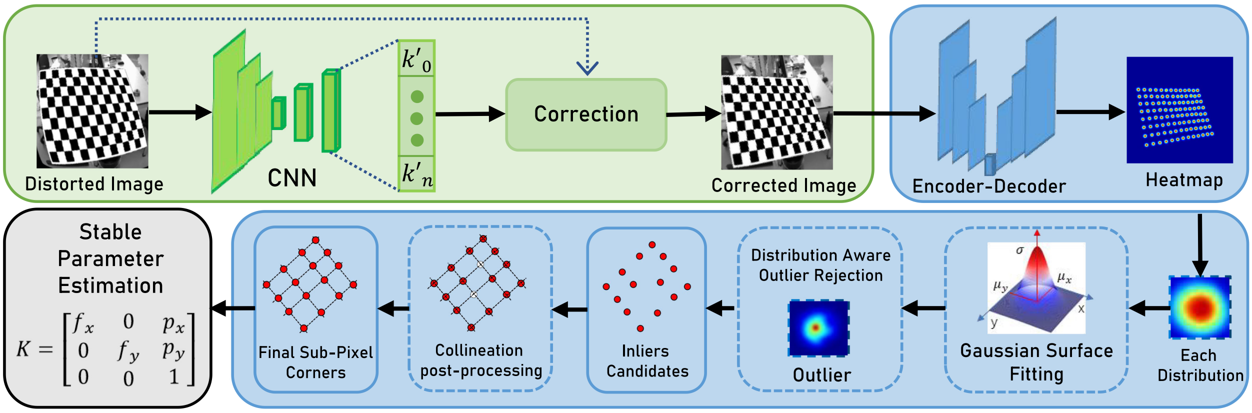

Fig. 2. Our framework includes three main parts. First, radial distortion is removed by correction model inferred from distorted images by our network (green part). The chessboard corner detection part consists of heatmap learning and sub-pixel refinement with outlier rejection and collineation post-processing (blue part). After precise sub-pixel corner coordinates are obtained, stable parameter estimation through a simplified RANSAC procedure is performed to achieve accurate camera parameters (gray part).

suboptimal calibration results. RANdom SAmple Consensus (RANSAC) [26] algorithm has been introduced into calibration to enhance the estimation stability. Y Lv *et al.* [27] propose a RANSAC-based method to evaluate camera intrinsic parameters and eliminate unreliable chessboard images depending on the distance between the circular point and the image of absolute conic. RANSAC-based algorithm is also adopted in [28] to exclude the detected feature points with overlarge noise. We also adopt a simplified RANSAC-based procedure to improve the robustness of parameter estimation. Different from previous work, as outlier rejection already gets involved in detection, we propose an efficient image-level RANSAC algorithm. Based on our experimental findings, this algorithm is aimed at searching for the camera model with both small overall error and best error consistency.

III. The Proposed Calibration Framework

In this section, we introduce the proposed camera calibration framework. The camera calibration task which we focus on is to estimate camera parameters (K, the intrinsic matrix includes focal length and principle points.) from a set of chessboard images (N images) ($\{I^i_{dist}\}^N_{i=0}$) which may contain distortion. It can be formulated as:

$$K = \mathcal{P}\left(\{I^i_{dist}\}^N_{i=0}\right) \tag{1}$$

where $\mathcal{P}$ is the calibration procedure. Based on the three challenges in calibration (Sec. I), we divide this task into three sub-tasks and propose our solutions for each of them.

A. Radial Distortion Correction

As simultaneous parameter estimation and distortion correction not only increase calibration effort but also reduce precision, our framework performs distortion correction first. Considering that chessboard images contain sufficient straight line features which are helpful for CNN to learn the distortion [21], [23], we adopt a CNN-based, 8 layers encoder with 3 regression layers to regress correction model parameters from images. Besides, we apply a more practical distortion model than previous work: the radial model ($r_d = r_c(k_0 + k_1 r_c^2 + \ldots)$), which is symmetric and flexible [18]. Its symmetric property maintains consistency between the distortion and correction. Thus we can generate distorted images (I_{dist}) to train this network, which outputs parameters of another radial model with higher order ($r_c = r_d(k'_0 + k'_1 r_d + k'_2 r_d^2 + \ldots)$) for correction. This model's flexibility allows us to choose appropriate number of output parameters according to the specific distortion:

$$\{k'_i\}^M_{i=0} = \mathcal{F}\left(I_{dist}, \Theta_{\mathcal{F}}\right) \tag{2}$$

where M is the correction parameter number and $\mathcal{F}$ is our network with parameter $\Theta_{\mathcal{F}}$. Then, distortion is corrected:

$$I_{corr} = \mathcal{C}\left(I_{dist}, \{k'_i\}^M_{i=0}\right) \tag{3}$$

where $\mathcal{C}$ is the correction function utilizing bilinear interpolation. In practice, to train the network, we generate massive distorted chessboard image by randomly sampling up to three parameters in distortion model. Some image examples can be seen in Fig. 5. As the distortion model and correction model have different parameters, only robust sampling grid loss is adopted [23]: $L_{grid} = \frac{1}{N}\sum_i^N ||p^i_{dst} - p^i_{cor}||_1$, where p_{dst} represent the location of distorted grid points and p_{cor} represent the corrected ones.

B. Chessboard Corner Detection

We propose heatmap learning detection with specially designed refinement and post-processing to obtain sub-pixel chessboard corner.

Heatmap learning detection: The ground truth heatmap (Y) is designed to represent each corner as a 2-dimensional Gaussian distribution ($\mathcal{G}$) centered at the labelled sub-pixel coordinate. Then we use these heatmaps as supervision to train the classical UNet [29] ($\mathcal{U}$) with the L2 detection loss: $L_{detect} = \iint_{R^2} \|\hat{Y}(x,y) - Y(x,y)\|^2 dxdy$. Therefore, the image is transformed to a heatmap:

$$\hat{Y} = \mathcal{U}\left(I_{corr}, \Theta_{\mathcal{D}}\right) \tag{4}$$

where $\Theta_{\mathcal{D}}$ is the network parameters.

Sub-pixel refinement with outlier rejection: The Non-Maximum-Suppression (NMS) is first applied on heatmap to obtain distributions:

$$\{\hat{\mathcal{G}}_i\}_{i=0}^{J} = NMS(\hat{Y}) \tag{5}$$

where J is the chessboard corner number. Then, for each distribution, we find the center $\boldsymbol{\mu}$ and the variance $\boldsymbol{\sigma}^2$ by Gaussian surface fitting algorithm:

$$\{\boldsymbol{\mu}, \boldsymbol{\sigma}^2\} = \arg\min_{\boldsymbol{\mu}, \boldsymbol{\sigma}^2} \left\| \mathcal{G}\left(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) - \hat{\mathcal{G}} \right\|_2^2 \tag{6}$$

which can be solved using points (p_i) around the distribution peak:

$$p_i = \hat{Y}(x_i, y_i) = e^{-\frac{(x_i-\mu_x)^2}{2\sigma_x^2} - \frac{(y_i-\mu_y)^2}{2\sigma_y^2}} \tag{7}$$

Then we have:

$$p_i \times \ln p_i = \begin{bmatrix} p_i & p_i\, x_i & p_i\, y_i & p_i\, x_i^2 & p_i\, y_i^2 \end{bmatrix}$$
$$\begin{bmatrix} -\frac{\mu_x^2}{2\sigma_x^2} - \frac{\mu_y^2}{2\sigma_y^2} & \frac{\mu_x}{\sigma_x^2} & \frac{\mu_y}{\sigma_y^2} & -\frac{1}{2\sigma_x^2} & -\frac{1}{2\sigma_y^2} \end{bmatrix}^T$$

which can be expressed as:

$$a_i = b_i \cdot c_i^T \tag{8}$$

For N points, we can stack formulations as:

$$A = BC^T \tag{9}$$

Then we can solve the matrix C^T which contains $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ by the SVD decomposition. The $\boldsymbol{\mu} = (\mu_x, \mu_y)$ represents the corner's sub-pixel coordinate. As our network is trained to generate standard Gaussian distribution, abnormal distribution corresponds to unreliable detection. The distribution aware outlier rejection (OR) is to eliminate wrong detections according to $\boldsymbol{\sigma}^2$ compared with the variance of training data:

$$\|\boldsymbol{\sigma}^2 - \boldsymbol{\sigma}_{train}^2\|_1 > threshold \tag{10}$$

if the (10) is statisfied, the distribution $\hat{\mathcal{G}}$ will be treated as outlier:

$$\{\boldsymbol{\mu_i'}\}_{i=0}^{L} = OR\left(\{\boldsymbol{\mu_i}, \boldsymbol{\sigma_i^2}\}_{i=0}^{J}\right) \tag{11}$$

where L is the inlier number and $L \leqq J$.

Collineation post-processing After inliers are selected, collineation post-processing (CP) is proposed not only to refine the sub-pixel coordinates but also to recover some lost corners. However, before collineation post-processing, we need to sort unordered corners after outlier rejection. Therefore we can get sets of corners belong to each line. We sort corners based on OpenCV [5] and [30]. After corners are sorted, we take sets of corners to fit lines. Due to the distortion being removed firstly by our framework, the final corner coordinates are calculated by intersecting these lines (Fig. 3):

$$U = \{\hat{\boldsymbol{\mu}}_{\boldsymbol{i}}\}_{i=0}^{J} = CP\left(\{\boldsymbol{\mu_i'}\}_{i=0}^{L}\right) \tag{12}$$

where U is the final sub-pixel corners set of a image.

In sum, the proposed corner detection method can be described as:

$$U = \mathcal{D}\left(I_{corr}, \Theta_{\mathcal{D}}\right) \tag{13}$$

where $\mathcal{D}$ is the overall detection method.

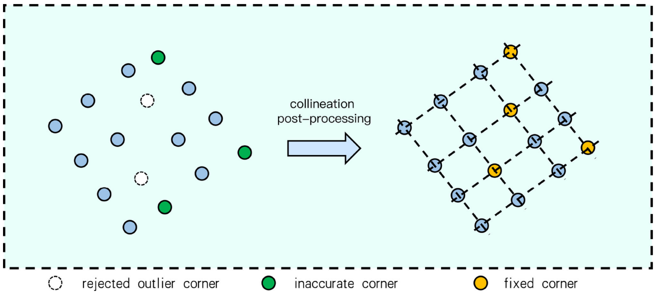

Fig. 3. The proposed collineation post-processing which can recover lost corners due to outlier rejection and refine inaccurate corners based on the chessboard corner collineation after distortion correction.

C. Parameter Estimation

Traditional calibration objective function (re-projection error minimization) assumes the 2D detection is accurate, which conflicts with the reality that detection always contains noise. Thus the re-projection error minimization under noisy detection is to search a camera model which minimizes most of points' re-projection error. As the detection noise pattern is unknown, the camera model may not be accurate although the re-projection error is small. Hence, we explore the relationship between calibration precision and re-projection error (Sec. IV-C). Experiment results indicate that small re-projection errors may not guarantee accurate camera models. However, the camera model's re-projection error distribution in all images can help us to distinguish accurate camera models with small and similar re-projection errors. As images in the same calibration set often contains similar noise (they are taken in the same scene), accurate camera model exhibits small and close re-projection errors in these images. Based on this experimental finding, we propose an effective parameter estimation method which improves Zhang's technique [7] by the simplified RANSAC procedure. This method can be described as follows:

1) Choose some of the images randomly to estimate parameters based on Zhang's technique.
2) Calculate the re-projection errors of all images and determine the inliers whose re-projection errors are less than the threshold.
3) Output the parameters if the inliers number exceeds threshold or iteration times are big enough (output the best model who has the most inliers); otherwise repeat the above steps.

With appropriate threshold, we can obtain accurate camera model with both small overall re-projection error and best consistency of all images' re-projection errors through this RANSAC-based estimation. Experiment results prove its effectiveness as well. The parameter estimation procedure ($\mathcal{E}$) can be expressed as:

$$K = \mathcal{E}\left(\{U_i\}_{i=0}^{N}\right) \tag{14}$$

The accurate camera model (K) can be achieved under the input of corner sets.

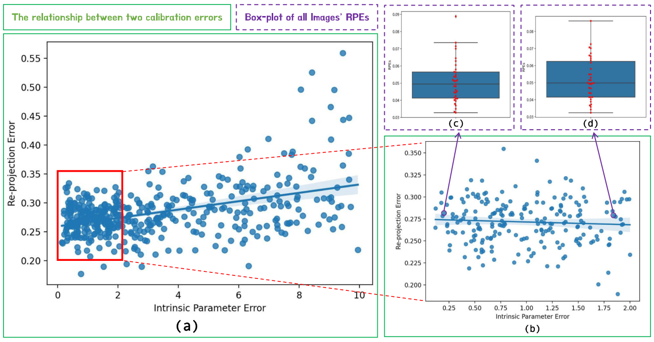

Fig. 4. Calibration experiment with noisy detection to show the relationship between re-projection error and calibration accuracy. (a) The regress plot of intrinsic parameter error (E_{IP}) and RMS re-projection error (RPE) in 2 K calibration trails. (b) Figure (a) is zoomed in on the red rectangular area where RPE can not represent calibration precision. (c) The box-plot of single image re-projection error (SI-RPE) distribution in calibration image set of an accurate camera model. (d) Another SI-RPE distribution box-plot of an inaccurate camera model with similar RPE as the accurate model.

D. Framework Integration

In sum, the three sub-module described above is integrated into a novel calibration framework (Fig. 2):

$$K = \mathcal{E}\left(\left\{\mathcal{D}\left(\mathcal{C}\left(I^i_{dist}, \mathcal{F}\left(I^i_{dist}, \Theta_{\mathcal{F}}\right)\right), \Theta_{\mathcal{D}}\right)\right\}^N_{i=0}\right) \tag{15}$$

which takes a set of chessboard images as input and outputs accurate and stable camera parameters.

IV. Experimental Results

The performance of our camera calibration framework is evaluated on both synthetic and real data. We also construct experiments to demonstrate the accuracy of our feature detection part.

A. Dataset and Metric

To train our networks, we generate massive chessboard images (image size: 480×480, chessboard size: $5 \times 6 \sim 12 \times 9$) with ground truth corner heatmaps and camera parameters (focal length: $100 \leq f_x, f_y \leq 300$ in pixel, principal points: $120 \leq p_x, p_y \leq 360$ in pixel and random extrinsic parameters). Moreover, noise, bad lighting, distortion and fake background using TUM dataset [31] are applied as data augmentation. Specifically, the image distortion level is decided by parameters k_0, k_1 and k_2. The example synthetic images can be seen in Fig. 5 (left). We use the metrics related to focal length (FL) and principal points (PP) in intrinsic matrix which can be defined as:

$$E_{FL} = \|FL_{GT} - \hat{FL}\|_1 \tag{16}$$

$$E_{PP} = \|PP_{GT} - \hat{PP}\|_1 \tag{17}$$

$$E_{IP} = \frac{1}{2}(E_{FL} + E_{PP}) \tag{18}$$

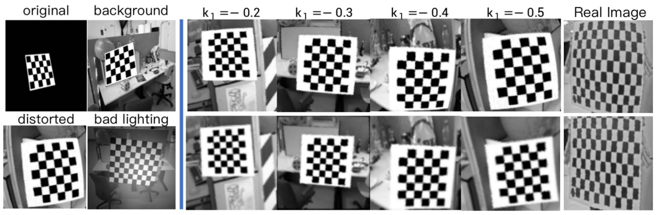

Fig. 5. **Left:** Examples of synthetic images. **Right:** Examples of images under different degrees of distortion($k_0 = 1, -0.5 \leq k_1 \leq -0.2, k_2 = -0.02$) and real distorted image (top) along with the corrected images (bottom) acquired by our method.

The RMS re-projection error (RPE) is used too.

$$RPE = \frac{1}{\sqrt{N \times M}} \sum_i^N \sum_j^M \|p_{ij} - s_i K[R_i|t_i]P_{ij}\|_2 \tag{19}$$

where p_{ij} are the 2D points and P_{ij} are corresponding 3D points. M is the number of chessboard corners. N is the number of images.

B. Implementation Details

Training Settings: **a)** The distortion correction model is trained with 40 K distorted images sized 480×480 and batched 4. The Adam [32] optimizer is used. The base learning rate is set as 0.01 with weight decay of $1e^{-8}$ and 200 epochs are used. **b)** The corner detection model is trained with 60 K, 480×480 chessboard images. Optimization is performed by Adam as well with batch size set as 4 and 800 epochs are used. The base learning rate is set as 0.001 with weight decay of $1e^{-8}$.

Parameter Estimation Settings: The outlier RPE threshold is set as $0.06 \sim 0.08$ in our experiments. The inlier image number threshold can be set as $\frac{3}{4} \times N$, where N is the image number of calibration set.

C. Calibration Accuracy Exploration

Traditional calibration objective function, the re-projection error minimization, may not directly reflect the calibration precision under noisy detection. To explore the relationship between calibration accuracy and re-projection error under noisy detection, we conduct experiment with 2K calibration trails with our detection results. Specifically, 5 sets of chessboard images (40 images per calibration set with synthetic noise), ground truth camera parameters and our detection results on them are collected. Our detection is with corner error in $0 \sim 2$ pixels because of the noise applied on images. Each trail we randomly take $3 \sim 20$ images from one of the image sets to perform calibration based on Zhang's method [7] and get the RMS re-projection error (**RPE**, (19)). The intrinsic parameter error (***E*$_{IP}$**, (18)) of each calibrated camera model is calculated as the true calibration accuracy. In order to better view the re-projection error distribution in different images, we use the calibrated camera model of each trail to calculate the single image re-projection errors (**SI-RPE**) of all the images in the calibration set. The experiment results are summarised as regress

TABLE I
COMPARISON OF CAMERA CALIBRATION WITH DIFFERENT IMAGES

Calibration Methods			Noise		Bad Lighting		Distortion I		Distortion II	
Distortion Correction	Feature Detection	Parameter Estimation	E_{IP}	RPE	E_{IP}	RPE	E_{IP}	RPE	E_{IP}	RPE
Opt.	Hand-crafted	Std.	1.52	0.53	1.93	0.47	2.71	0.64	5.35	0.57
Opt.	Learning Resp. [17]	Std.	0.95	0.54	1.03	0.63	1.73	0.94	3.13	1.40
Learning [23]	Learning Resp. [17]	IAC-RAN. [27]	0.79	0.46	0.81	**0.35**	1.16	1.02	2.69	0.97
Ours	Learning Detc. [14]	IAC-RAN. [27]	0.85	0.38	0.92	0.41	0.81	0.64	1.55	0.73
Ours	Learning Resp. [17]	Ours	0.73	0.41	0.77	0.36	0.69	**0.43**	1.41	0.86
Learning [23]	Ours	Ours	0.68	0.23	0.74	0.39	0.67	0.65	2.57	0.76
Ours	Ours	Ours	**0.68**	**0.23**	**0.74**	0.39	**0.60**	0.55	**1.12**	**0.53**

plots of two calibration errors and box-plots of calibrated camera model's SI-RPE distribution in all images of the calibration set. (Fig. 4) It can be seen that smaller RPE corresponds to more accurate camera model in general (when the $E_{IP} = 0 \sim 10$, Fig. 4(a)). However, when the $E_{IP} = 0 \sim 2$, the RPE does not reflect the calibration accuracy (Fig. 4(b)). It shows RPE minimization can not yield stable and accurate result, but we find that the distribution of SI-RPE can guide us to find more accurate camera model. Even with the similar RPEs, more accurate model exhibits better consistency of re-projection errors in different images. (Examples can be seen in Fig. 4(c), (d)) This can be explained as images of the same set share the noise pattern (correspond to images taken in the same scene in real life), thus the accurate camera model tends to output small and similar re-projection error in every image. This finding motivates the proposed RANSAC-based calibration procedure, which can screen out accurate camera model with both small overall RPE and good consistency of SI-RPE.

D. Calibration Performance

To extensively evaluate our framework, we conduct calibration experiments on four different image configurations: noise, bad lighting, distortion and real data. As our framework divides calibration into three part, we collect state-of-art methods belong to these parts and organize them into different calibration frameworks with different combinations.

1) The distortion correction part. For conventional solution, we choose the optimization-based method [7] (**Opt.**). The learning-based effective state-of-art method [23] (**Learning**) is taken for comparison as well.

2) The feature detection part includes a hand-crafted feature-based method [5] (**Hand-crafted**) and two recent learning feature-based methods [14], [17] (**Learning Resp.** and **Learning Detc.**).

3) The parameter estimation part. As our work is based on Zhang's method [7], we take standard Zhang's method [5] (**Std.**) and another image-level RANSAC method [27] (**IAC-RAN.**) based on absolute conic is taken for comparison.

For better visualization, we summarize calibration results of some representative combinations in Table. I. All of these results are average results of 50 independent calibration trails on different images (40 images per trail) and camera parameters.

Calibration under noise and bad lighting: The first experiment is to demonstrate the robustness and accuracy of our framework in terms of environmental noise like low sensor resolution and uneven illumination. We apply 3×3 (on 20 calibration trails), 5×5 (20 trails) and 7×7 (10 trails) Gaussian kernels to blur images for noise simulation. The uneven brightness is simulated by specular lighting model [33] with random size and center on each calibration trail. The average results are summarized in the 'Noise' and 'Bad Lighting' columns of Table. I. Notice that since the distortion is absent here, the last two rows of Table. I show the same results. It can be seen that the detection part matters in calibration and higher detection precision (which can be seen in Table. III) corresponds to higher calibration accuracy. Moreover, the RANSAC algorithms do decrease the RPE and E_{IP}. The IAC-RAN. method chooses outliers based on initial guess of Zhang's method which works well under slight noise but the initial guess is unreliable under difficult condition. Our RANSAC procedure excludes outliers based on re-projection error which is stable against different conditions. The related results (lower E_{IP} using our PE part) can be seen in the third and fifth rows of Table. I. In general, our framework achieves best results among all the combinations as our detection part is more accurate and both point-level and image-level outliers are exclude in our framework.

Calibration under distortion: We also conduct calibration experiments on distorted images, and we set two different distortion levels by randomly setting 3 parameters. The first distortion level's parameters are: $k_0 = 1, -0.2 \leq k_1 \leq -0.35, -0.1 \leq k_3 \leq 0$. The second distortion level is more severe, whose parameters are: $0.8 \leq k_0 \leq 1.2, -0.35 \leq k_1 \leq -0.5, -0.3 \leq k_3 \leq -0.1$. In practice, our network outputs 5 parameters for correction and the learning method [23] is trained by our synthetic dataset. The average results are shown in the 'Distortion I' and 'Distortion II' columns of Table. I. Learning-based distortion correction methods perform better than traditional methods according to the results. Our method gets comparable results to state-of-art method [23] under slight distortion ('Distortion I' column). However, under severe distortion, our method outperforms others by a noteworthy margin as shown in the last two rows of 'Distortion II' column. This confirms that our framework maintains high precision under different distortion levels. On the other hand, our distortion correction method works well with two learning-based detection methods which demonstrates the correction capability of our method.

Calibration on real data: To evaluate the performance of our system under realistic conditions, we perform calibration on a HIKROBOT MV-CA016-10GM camera with resolution of 1440×1080 (resized to 480×480) by a 12×8 chessboard.

TABLE II
COMPARISON OF CALIBRATION ON REAL DATA

	Ours	Type I	Type II
f_x	780.26	784.56	781.37
f_y	1042.61	1049.93	1046.82
p_x	745.43	755.46	742.13
p_y	538.24	536.47	535.66
RPE	**0.37**	0.56	0.44
STD	**0.03**	0.10	0.06

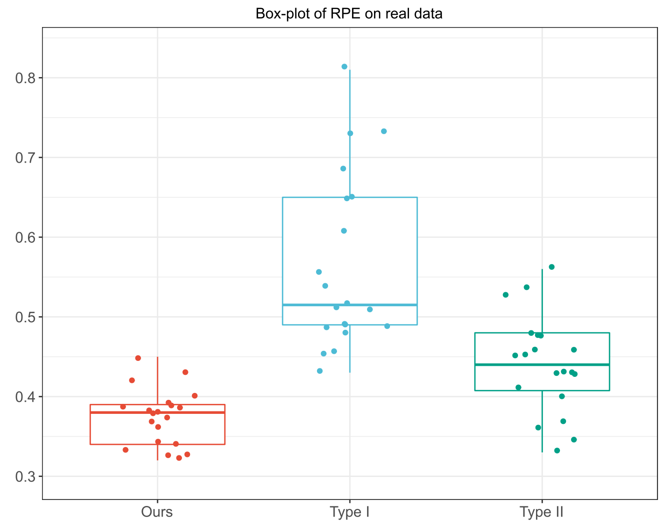

Fig. 6. The box-plot of calibration RPEs of 20 calibration trails on real data. The width of box indicates the dispersion of the data. It can be seen that our method get lower and more tight RPE distribution which prove the precision and stability of our calibration framework.

These images exhibit slight distortion and uneven illumination. We repeat 20 times of calibration with different combinations of chessboard poses and get the average results of intrinsic parameters, RPE and the standard deviation(STD) of RPE. Two types of calibration framework composed by state-of-art methods are selected for comparison. The **Type I** framework includes handcrafted feature-based detection, traditional distortion correction method and the standard estimation. We implement it based on OpenCV [5]. The **Type II** framework contains Learning Resp. detection, learning-based distortion correction [23] and the IAC-RAN. estimation. Table. II shows the results. We can observe that the three frameworks produce similar results and ours gets the lowest RPE and STD. For better visualization, we draw the box-plot of three methods' RPE (Fig. 6) in which we can seen the stability of each method and our RPE values are more stable than others.

E. Corner Detection Accuracy

As calibration benefits from precise chessboard corner coordinates, the accuracy of our corner detection method is tested on both synthetic and real data in this part. Compared with both feature-based [5], [10] and learning-based [14], [17] corner detection methods, we conduct experiments on synthetic images with different configurations including noise (5 × 5 Gaussian blur), bad lighting (random center and size) and distortion ($k_0 = 1, k_1 = -0.4, k_2 = -0.1$). As the corner detection on distorted images is not suitable for our framework where distortion correction is performed at first, we evaluate the detection accuracy on corrected images through our correction method (See Sec. III-A for details.). This evaluation can demonstrate the detection robustness against noise brought by the interpolation in image correction. Each configuration above contains 2 K chessboard images.

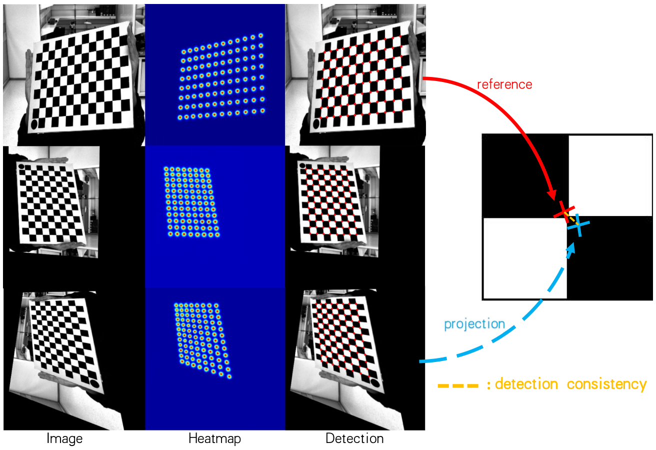

Fig. 7. The illustration of detection consistency under homography transformations. The random transformations are applied on original image first. Then the detection results on these transformed images are projected on the original image. The distance between the original detection (red cross) and detection projection (blue cross) is taken as detection consistency (dotted yellow line). This metric is used to evaluate the detection performance on real images.

In our experiments, detection performance is also evaluated on real data. Due to the lack of ground truth coordinates in real images, we evaluate the consistency of detection through known homography transformation as the performance demonstration. Specifically, we collect 69 real chessboard images. For each real image, we apply 20 random homography transformations on it. Then the corner detection are performed on these images. The detection results on original image are taken as reference, while the results of transformed ones are projected to original image through the known homography. The distances between projections and reference are calculated as the detection consistency which can be seen in Fig. 7.

Moreover, in order to demonstrate that the post-processing is helpful for detection, our detection pipeline with post-processing discarded (**Ours w/o OR & PP**) are evaluated. As the lost corners recovery is rely on the post-processing, the outlier rejection part is turned off as well. The results are shown in Table. III. It can be seen that our approach results in comparable precision under noise and more accurate detection under bad lighting and distortion rectification. These results are consistent with the calibration experiments and proves the precision of our method. The better detection consistency on real data also proves our method's better performance. On the other hand, the proposed outlier rejection and post-processing are in particular advantageous in order to detect under difficult situation, which can deliver up to 34% accuracy improvement.

TABLE III
CORNER DETECTION ACCURACY UNDER DIFFERENT IMAGES

Corner Detection Accuracy (pixels)				
Methods	Image Configuration			
	Noise	Bad Lighting	Correction[1]	Real[2]
OpenCV [5]	2.23	2.43	2.94	1.27
libcb [10]	2.66	2.72	1.98	0.94
Kang et al. [14]	1.02	0.97	1.53	0.72
Chen et al. [17]	**0.93**	0.76	1.21	0.79
Ours w/o OR & PP	1.08	0.73	1.17	0.63
Ours w/ OR & PP	0.96	**0.51**	**0.89**	**0.41**

[1] The distorted images are corrected by our method.

[2] The accuracy is quantified by detection consistency under known homography transformation. See Sec. IV-E for details.

V. CONCLUSION

In this paper, the accuracy and robustness of camera calibration are improved from three aspects: distortion correction, corner detection and parameter estimation. Specifically, the distortion correction is performed by the learning-based method. Accurate feature locations are achieved by the combination of learning-based detection, specially designed refinement and complete post-processing. Moreover, we obtain stable parameter estimation by a RANSAC procedure. Finally, these proposed methods are integrated into a novel calibration framework. This framework surpasses other state-of-art methods by a noteworthy margin in terms of calibration precision on both synthetic and real dataset. Extensive experiments prove the robustness of this framework against noise, bad lighting and radial distortion as well. Besides, our corner detection method is evaluated where decent results manifest the accuracy and contribution of this part to our framework.

REFERENCES

[1] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2017.

[2] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[3] P. F. Martins, H. Costelha, L. C. Bento, and C. Neves, "Monocular camera calibration for autonomous driving – A comparative study," in *Proc. IEEE Inter. Conf. Auton. Robot Syst. Competitions*, 2020, pp. 306–311.

[4] J. Barreto, J. Roquette, P. Sturm, and F. Fonseca, "Automatic camera calibration applied to medical endoscopy," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 1–10.

[5] G. Bradski, "The Open CV library," *Dr Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–123, 2000.

[6] J.-Y. Bouguet, *Camera Calibration Toolbox for Matlab*, Pasadena, CA and Geneva, Switzerland: CaltechDATA, May 2022.

[7] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[8] L. Lucchese and S. Mitra, "Using saddle points for subpixel feature detection in camera calibration targets," in *Proc. Asia-Pacific Conf. Circuits Syst.*, 2002, vol. 2, pp. 191–195.

[9] S. Placht et al., "Rochade: Robust checkerboard advanced detection for camera calibration," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 766–779.

[10] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 3936–3943.

[11] S. Donné, J. De Vylder, B. Goossens, and W. Philips, "MATE: Machine learning for adaptive calibration template detection," *Sensors*, vol. 16, pp. 1858–17, Nov. 2016.

[12] B. Chen, C. Xiong, and Q. Zhang, "CCDN: Checkerboard corner detection network for robust camera calibration," in *Proc. Intell. Robot. Appl.*, 2018, pp. 324–334.

[13] H. Wu and Y. Wan, "A highly accurate and robust deep checkerboard corner detector," *Electron. Lett.*, vol. 57, no. 8, pp. 317–320, Mar. 2021.

[14] J. Kang, H. Yoon, S. Lee, and S. Lee, "Sparse checkerboard corner detection from global perspective," in *Proc. IEEE Int. Conf. Signal Image Process. Appl.*, 2021, pp. 12–17.

[15] J. Schroeter, T. Tuytelaars, K. Sidorov, and D. Marshall, "Learning multi-instance sub-pixel point localization," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 669–686.

[16] A. R. Kosiorek, S. Sabour, Y. W. Teh, and G. E. Hinton, "Stacked capsule autoencoders," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2020, pp. 15512–15522.

[17] B. Chen, Y. Liu, and C. Xiong, "Automatic checkerboard detection for robust camera calibration," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.

[18] Z. Tang, R. Grompone von Gioi, P. Monasse, and J.-M. Morel, "A precision analysis of camera distortion models," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2694–2704, Sep. 2020.

[19] J. Salvi et al., *An Approach to Coded Structured Light to Obtain Three Dimensional Information*. Girona, Spain: Universitat de Girona, 1998.

[20] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Autom.*, vol. 3, no. 4, pp. 323–344, Aug. 1987.

[21] Z. Xue, N. Xue, G.-S. Xia, and W. Shen, "Learning to calibrate straight lines for fisheye image rectification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1643–1651.

[22] X. Yin, X. Wang, J. Yu, M. Zhang, P. Fua, and D. Tao, "Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 469–484.

[23] H. Zhao, Y. Shi, X. Tong, X. Ying, and H. Zha, "A simple yet effective pipeline for radial distortion correction," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 878–882.

[24] C. Harris and M. Stephens, "A combined corner and Edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 10–5244.

[25] J. Rong, S. Huang, Z. Shang, and X. Ying, "Radial lens distortion correction using convolutional neural networks trained with synthesized images," in *Proc. Asian Conf. Comput. Vis.*, 2017, pp. 35–49.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] Y. Lv, J. Feng, Z. Li, W. Liu, and J. Cao, "A new robust 2D camera calibration method using ransac," *Optik*, vol. 126, no. 24, pp. 4910–4915, Dec. 2015.

[28] F. Zhou, Y. Cui, Y. Wang, L. Liu, and H. Gao, "Accurate and robust estimation of camera parameters using ransac," *Opt. Lasers Eng.*, vol. 51, no. 3, pp. 197–212, Mar. 2013.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. image Comput. Comput.- Assist. Intervention*, 2015, pp. 234–241.

[30] F. Jin and X. Wang, "An autonomous camera calibration system based on the theory of minimum convex hull," in *Proc. Int. Conf. Instrum. Meas., Comput., Commun. Control*, 2015, pp. 857–860.

[31] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D slam systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[33] B. T. Phong, "Illumination for computer generated pictures," *Commun. ACM*, vol. 18, pp. 311–317, 1975.