Infrastructure-based Multi-Camera Calibration using Radial Projections

Yukai Lin[1], Viktor Larsson[1], Marcel Geppert[1], Zuzana Kukelova[2], Marc Pollefeys[1,3], and Torsten Sattler[4]

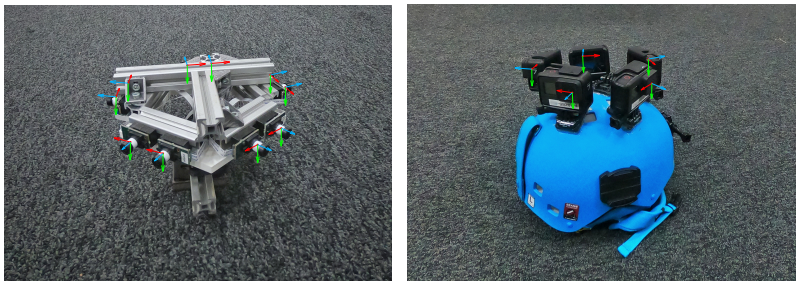
[1] Department of Computer Science, ETH Zrich,
[2] VRG, Faculty of Electrical Engineering, Czech Technical University in Prague
[3] Microsoft Mixed Reality & AI Zurich Lab
[4] Department of Electrical Engineering, Chalmers University of Technology

Abstract. Multi-camera systems are an important sensor platform for intelligent systems such as self-driving cars. Pattern-based calibration techniques can be used to calibrate the intrinsics of the cameras individually. However, extrinsic calibration of systems with little to no visual overlap between the cameras is a challenge. Given the camera intrinsics, infrastucture-based calibration techniques are able to estimate the extrinsics using 3D maps pre-built via SLAM or Structure-from-Motion. In this paper, we propose to fully calibrate a multi-camera system from scratch using an infrastructure-based approach. Assuming that the distortion is mainly radial, we introduce a two-stage approach. We first estimate the camera-rig extrinsics up to a single unknown translation component per camera. Next, we solve for both the intrinsic parameters and the missing translation components. Extensive experiments on multiple indoor and outdoor scenes with multiple multi-camera systems show that our calibration method achieves high accuracy and robustness. In particular, our approach is more robust than the naive approach of first estimating intrinsic parameters and pose per camera before refining the extrinsic parameters of the system. The implementation is available at *https://github.com/youkely/InfrasCal*.

1 Introduction

Being able to perceive the surrounding environment is a crucial ability for any type of autonomous intelligent system, including self-driving cars [11, 34] and robots [36]. Multi-camera systems (see *e.g.* Figure 1) are popular sensors for this task: they are cheap to build and maintain, consume little energy, and provide high-resolution data under a wide range of conditions. Enabling 360° perception around a vehicle [34] using such systems, makes visual localization [2, 7] and SLAM [24] more robust.

Multi-camera systems need to be calibrated before use. This includes calibrating the intrinsic parameters of each camera, *i.e.* focal length, principal point and distortion parameters, as well as the extrinsic parameters between cameras, *i.e.*, the relative poses between them. An accurate and efficient calibration is often



(a) Pentagonal camera rig (b) GoPro helmet

Fig. 1. Multi-camera rigs. Our estimated rig calibrations are overlayed in the figure. (a) Pentagonal camera rig with five stereo pairs. (b) Ski helmet with five GoPro Hero7 Black attached covering 360° panoramic view.

crucial for safe and robust performance. A standard approach to this problem, implemented in calibration toolboxes such as Kalibr [27], is to use a calibration pattern to record data which covers the full field-of-view (FoV) of the cameras. Although this method is powerful in achieving high accuracy, it is computationally expensive and recording a calibration dataset with adequate motion/view coverage is cumbersome, especially for wide-FoV cameras. Moreover, it is incapable of calibrating the camera-rigs with little or even no overlapping fields of view, which is often the case for applications in autonomous vehicles.

Another approach to handle such scenarios are sequence- and infrastructure-based calibration [12, 13]. In both cases, the methods require prior knowledge of the intrinsics before the extrinsics calibration, which still requires a per camera pre-calibration step using calibration patterns.

In this paper, we introduce an infrastructure-based calibration that calibrates both intrinsics and extrinsics in a single pipeline. Our method uses a pre-build map of sparse feature points as a substitute for the calibration patterns. The map is easily built by a Structure-from-Motion pipeline, *e.g.* COLMAP [33]. We calibrate the camera-rigs in a two-stage process. In the first stage, the camera poses are estimated under a radial camera assumption, where the extrinsics are recovered up to an unknown translation along the principal axis. In the second stage, the intrinsics and the remaining translation parameters are jointly estimated in a robust way. We demonstrate the accuracy and robustness through extensive experiments in indoor and outdoor datasets with different multi-camera systems.

The **main contributions** of this paper are: (**1**) We propose an infrastructure-based calibration method for performing multi-camera rig intrinsics and extrinsics calibration in an user-friendly way as we remove the need for pre-calibration for each camera or tedious recording for calibration pattern data. (**2**) In contrast to current methods, we show that it is possible to first (partially) estimate the camera rig's extrinsic parameters before estimating the internal calibration for each camera. (**3**) Our proposed method is experimentally shown to give high-quality camera calibrations in a variety of environments and hardware setups.

2 Related Work

Pattern-Based Calibration. A pattern-based calibration method estimates camera parameters using special calibration patterns such as AprilTags [28] or checkerboards [27, 37, 42]. The patterns are precisely designed so that they can be accurately estimated via camera systems. We note that the pattern-based calibration of multi-camera systems usually requires the camera pairs to have overlapping FoVs, since the pattern must be visible in multiple images to constrain the rig's extrinsic parameters. Some works [19,23,31] calibrate the cameras without assuming the overlapping FoV. Kumar et al. [19] show that the use of an additional mirror can help to create overlap between cameras. Li et al. [23] only require the neighboring cameras to partially observe the calibration patterns at the same time but the observed parts do not necessarily need to overlap. Robinson et al. [31] calibrate the extrinsic parameters for non-overlapping cameras by temporarily adding an additional camera during calibration with an overlapping FoV with both cameras. We note that the use of a calibration pattern board always introduces a certain viewing constraint or extra effort to calibrate the cameras with non-overlapping FoV. Furthermore, the calibration of wide FoV cameras is especially cumbersome. The pattern needs to be close to the camera to cover any significant part of the image but if it is too close, it leads to problems where the pattern is out of focus. Thus, to get accurate calibration results, it is typically necessary to capture a large number of images.

Infrastructure-Based Calibration. Rather than using calibration patterns, infrastructure-based calibration uses natural scene features to estimate camera parameters. Carrera et al. [5] propose a feature-based extrinsic calibration method through a SLAM-based feature matching among the maps for each camera. Heng et al. [12] simplify that approach to rely on a prior high-accuracy map, removing the need for inter-camera feature correspondences and loop closures. Their pipeline first infers camera poses via the P3P method for calibrated cameras, and subsequently, an initial estimate of the camera-rig transformations and rig poses. A final non-linear refinement step optimizes the camera-rig transformations, rig poses and optionally intrinsics.

Our method is most similar to the work of Heng et al. [12] in that we use a prebuilt sparse feature map for calibration. However, their method relies on a known intrinsics input which still requires calibration patterns for intrinsic calibration. Our method does not require a prior intrinsics knowledge and performs complete calibration, both intrinsic and extrinsic, using the sparse map.

Compared to checkerboard-style calibration objects, infrastructure-based methods are able to get significantly more constraints per-image since there are typically more feature points observed which acts as a virtual large calibration pattern. In practice, infrastructure-based calibration provides a much wider application range than pattern-based calibration.

Camera Pose Estimation with Unknown Intrinsic Parameters. Given a sparse set of 2D-3D correspondences between an image and a 3D point cloud (a map), it is possible to recover the camera pose. If the cameras' internal cal-

ibration is known, i.e. the mapping from image pixels to viewing rays, the absolute pose estimation problem becomes minimal with three correspondences. This problem is usually referred to as the Perspective-Three-Points (P3P) problem [9]. In settings where the intrinsic parameters are unknown, the estimation problem becomes more difficult and more correspondences are necessary. Most modern cameras can be modeled as having square pixels (i.e. zero skew and unit aspect ratio). Due to this, most work on camera pose estimation with unknown/partially known calibration has focused on the case of unknown focal length. The minimal problem for this case was originally solved by Bujnak et al. [3]. Since then, there have been several papers improving on the original solver [18, 20, 30, 41, 43]. The case of unknown focal length and principal point was considered by Triggs [39] and later Larsson et al. [21]. When all of the intrinsic parameters are unknown, the Direct-Linear-Transform (DLT) [10] can be applied. Camera pose estimation with unknown radial distortion was first considered by Josephson and Byröd [15]. There have been multiple works improving on this paper in different aspects; faster runtime [17, 20], support for other distortion models [22] and even non-parametric distortion models [4].

Radial Alignment Constraint and the 1D Radial Camera Model. Focal length and radial distortion only scales the images points radially outwards from the principal point (assuming this is the center of distortion). This observation was used by Tsai [40] to derive constraints on the camera pose which are independent of the focal length and distortion parameters. For a 2D-3D correspondence, the idea is to only require that the 3D point projects onto the radial line passing through the 2D image point, and to ignore the radial offset. This constraint is called the Radial Alignment Constraint (RAC). This later gave rise to the 1D radial camera model (see [38]) which re-interprets the camera as projecting 3D points onto radial lines instead of 2D points. Since forward motion also moves the projections radially, it is not possible to estimate the forward translation using these constraints. In practice, the 1D radial camera model turns out to be equivalent to only considering the top two rows of the normal projection matrix. Instead of reprojection error, radial reprojection error measures the distance from 2D point to projected radial line, which is invariant to focal length and radial distortion parameters.

These ideas have also been applied to absolute pose estimation with unknown radial distortion. In Kukelova et al. [17], the authors present a two-stage approach which first estimates the camera pose up to an unknown forward translation using the RAC. In a second step the last translation component is jointly estimated with the focal length and distortion parameters. This was later extended in Larsson et al. [22]. Camposeco et al. [4] applied a similar approach to non-parametric distortion models.

In this paper we take a similar approach as [4, 17, 22]. However, instead of using just one frame, we can leverage multiple frames for the upgrade step since we consider multi-camera systems. We show it is possible to use joint poses of multiple (non-parallel) 1D radial cameras to transform the frames into the camera coordinate system for each single camera.

3 Multi-Camera Calibration

Now we present our framework for calibration of a multi-camera system. Our approach is similar to the infrastructure-based calibration method from Heng et al. [12]. We improve on their approach in the following aspects:

- We leverage state-of-the-art absolute pose solvers [17, 22] to also perform estimation of the camera intrinsic parameters, thus completely removing any need for pre-calibrating each camera independently.
- We present a new robust estimation scheme to initialize the rig extrinsic parameters. Our experiments show that this greatly improves the robustness of the calibration method, especially on datasets with shorter image sequences.
- Finally we show it is possible to first partially estimate the rig extrinsics and pose before recovering the camera intrinsic parameters. This partial extrinsic knowledge allows us to more easily incorporate information from multiple images into the estimation.

Similarly to Heng et al. [12] we assume that we have a sparse map of the environment. The input to our method is then a synchronized image sequence captured by the multi-camera system as it moves around in the mapped environment. The main steps of our pipeline are presented below and detailed in the next sections.

1. **Independent 1D radial pose estimation.** We independently estimate a 1D radial camera pose (see Section 2) for each image using RANSAC [6].
2. **Radial camera rig initialization.** We robustly fit a multi-camera rig with the 1D radial camera model to the estimated individual camera poses.
3. **Radial Bundle Adjustment.** We optimize the partial rig extrinsics and poses by minimizing the radial reprojection error (see Section 2). Here we additionally refine the principal point for each camera.
4. **Forward translation and intrinsic estimation.** Using the partially known extrinsic parameters and poses of the rig we can transform all 2D-3D correspondences into the camera coordinate system (up to the unknown forward translation). This allows us to use the entire image sequence when initializing the intrinsic parameters and the forward translations [4].
5. **Final refinement.** Finally, we perform bundle adjustment over rig poses, rig extrinsic and intrinsic parameters, minimizing the reprojection error over the entire sequence.

The entire calibration pipeline is illustrated in Figure 2.

3.1 The Sparse Map and Input Framesets

One of the inputs to our method is a pre-built sparse feature map, which can be built using a standard Structure-from-Motion pipeline, e.g. COLMAP [33]. It is necessary to build a high-accuracy map in order to produce accurate calibration result. The map can be used as long as there is no large change in the environment. The correct scale of the map can be derived either from a calibrated multi-camera system, e.g. stereo system, or by the user measuring some

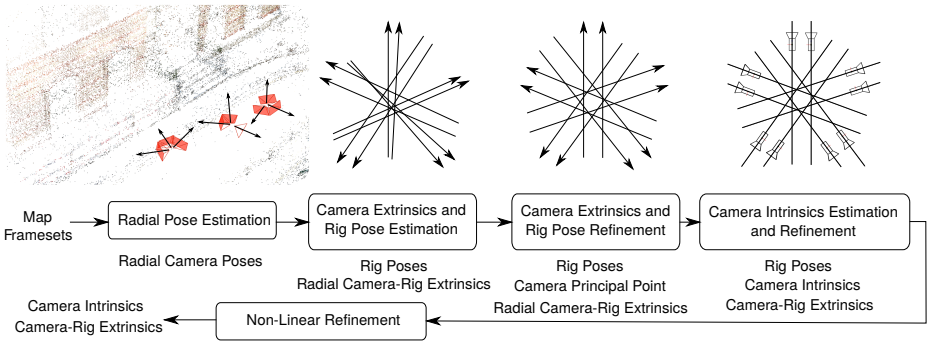


Fig. 2. Illustration of the calibration pipeline. The output of each step is placed below each block. In the first step we independently estimate the 1D radial pose of each camera. Next we robustly fit a rig to the estimated poses. We refine the rig extrinsic parameters and poses by minimizing the radial reprojection errors. Then we upgrade each camera by estimating the last translation component jointly with the internal calibration. Finally we refine all parameters by minimizing the reprojection error.

distances in both the real world and the map and scaling the map accordingly, e.g. by using a checkerboard. In addition, a sequence of synchronized images captured in the map is recorded as the calibration dataset. We define a frameset to be a set of images captured at the same timestamp from all different cameras.

3.2 Initial Camera Pose Estimation

The first step of our pipeline is to independently estimate the pose of each image with respect to the pre-built map. Using the 1D radial camera model allows us to estimate the pose of the camera (up to forward translation) without knowing the camera intrinsic parameters (see Section 2). Similarly to Heng et al. [12], to find 2D-3D correspondences between the query image and the map we use a bag-of-words-based image retrieval against the mapping images, followed by 2D-2D image matching. For local features/descriptors we use upright SIFT [25], but any local feature could be used. Once the putative 2D-3D correspondences are found we use the minimal solver from Kukelova et al. [17] (see Section 2) in RANSAC to estimate the 1D radial camera pose. The principal point for each camera is initialized to the image center, a valid assumption for common cameras, and could be recovered accurately in later steps. Note that this only estimates the orientation and two components of the camera translation. At this stage we filter out any camera poses with too few inliers.

Alternatively we can also use the solvers from [17, 22] which directly solve for the intrinsic parameters. However, estimating the intrinsic parameters from a single image turns out to be significantly less stable. See Section 5.4 for a comparison of the errors in the intrinsic calibration when we perform the intrinsic calibration at this stage of the pipeline.

3.3 Camera Extrinsics and Rig Poses Estimation

In the previous step we estimated the absolute poses for each image independently. Since we used the 1D radial camera model we only recovered the pose up to an unknown forward translation, i.e. we estimated

$$T_{ij} = \begin{bmatrix} R & \begin{pmatrix} t_x \\ t_y \\ ? \end{pmatrix} \end{bmatrix} \ , \tag{1}$$

which transforms from the map coordinate system to the coordinate system of ith camera in the jth frameset. The goal now is to use the initial estimates to recover both the rig extrinsic parameters as well as the rig pose for each frameset in a robust way. In [12], they simplify this problem by assuming that there is at least one frameset where each camera was able to get a pose estimate. In our experiments this assumption was often not satisfied for shorter image sequences, leading to the method completely failing to initialize.

Let P_i be the transform from the rig-centric coordinate system to the ith camera and let Q_j be the transform from the map coordinate system to the rig-centric coordinate system for the jth frameset. A rig-centric coordinate system can be set to any rig-fixed coordinate frame since we only consider the relative extrinsics. In our case, it is set initially to be the same as the first camera with the unknown forward translation being zero. For noise-free measurements we should have

$$T_{ij} = P_i Q_j, \quad (i,j) \in \Omega \ , \tag{2}$$

where Ω is the set of images that were successfully estimated in the previous step, i.e. $(i,j) \in \Omega$ if camera i in frameset j was successfully registered. Since we did not estimate the third translation component of T_{ij}, we restrict ourselves to finding the first two rows of the camera matrices, i.e.

$$\hat{T}_{ij} = \hat{P}_i Q_j, \quad (i,j) \in \Omega \ , \tag{3}$$

where $\hat{T}_{ij}$ denotes the first two rows of T_{ij} and similarly for $\hat{P}_i$. As described in Section 2 we can interpret $\hat{P}_i$ as 1D radial camera poses. If some $\hat{P}_i$ are known, then the rig poses Q_j can be found by solving

$$\begin{bmatrix} \hat{T}_{1j} \\ \hat{T}_{2j} \\ \vdots \end{bmatrix} = \begin{bmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \vdots \end{bmatrix} Q_j \quad \text{where} \quad Q_j = \begin{bmatrix} R & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \ , \tag{4}$$

which has a closed form solution using SVD [35]. Note that this requires that at least two cameras have non-parallel principal axes. We discuss this limitation more in Section 4. In turn, if the rig poses Q_j are known, we can easily recover the rig extrinsic parameters as $\hat{P}_i = \hat{T}_{ij} Q_j^{-1}$.

To robustly fit the rig extrinsics $\hat{P}_i$ and rig poses Q_j to the estimated absolute poses $\hat{T}_{ij}$, we solve the following minimization problem,

$$\min_{\{\hat{P}_i\},\{Q_j\}} \sum_{(i,j)\in\Omega} \rho\left(d\left(\hat{T}_{ij},\ \hat{P}_i Q_j\right)\right) \ , \tag{5}$$

where ρ is a robust loss function and d is a weighted sum of the rotation and translation errors. Since this is a non-convex problem we perform a robust initialization scheme based on a greedy assignment in RANSAC.

In our case we randomly select any frameset with at least two cameras as initialization and assign the corresponding $\hat{P}_i$ using the relative poses from this frameset. Note that this might leave some $\hat{P}_i$ unassigned. We then use these assigned poses to estimate the rig poses Q_j of any other frameset which also contains the already assigned $\hat{P}_i$. We can then iterate between assigning any of the missing $\hat{P}_i$ and estimating new Q_j. This back-and-forth search repeats until all of the rig extrinsics and rig poses are assigned. We repeat the entire process multiple times in a RANSAC-style fashion, keeping track of the best assignment with minimal radial reprojection over all frames. Finally, for the best solution we perform local optimization of (5) using Levenberg-Marquardt.

This approach is similar to the rotation averaging methods in [8, 29] which repeatedly build random minimum spanning trees in the pose-graph and assigns the absolute rotations based on these.

3.4 Camera Extrinsics and Rig Poses Refinement

We further refine the camera rig extrinsics and rig poses by performing bundle adjustment to minimize the radial reprojection error. In this step we also optimize over the principal point for each camera which was initialized to the image center. Let $\boldsymbol{X}_p$ be a 3D point and $\boldsymbol{x}_{ijp}$ its observation in the ith camera in frameset j. Then we optimize

$$\min_{\hat{P}_i, Q_j, \boldsymbol{c}_i} \quad \sum_{i,j,p} \rho\left(\left\| \pi_r\left(\hat{P}_i Q_j \boldsymbol{X}_p,\ \boldsymbol{x}_{ijp} - \boldsymbol{c}_i \right) - (\boldsymbol{x}_{ijp} - \boldsymbol{c}_i) \right\|^2 \right) \ , \tag{6}$$

where ρ is a robust loss function, $\boldsymbol{c}_i$ is the principal point of the ith camera and $\pi_r : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ is the orthogonal projection of the second argument onto the line generated by the first, i.e. $\pi_r(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^T \boldsymbol{v}}{\boldsymbol{u}^T \boldsymbol{u}} \boldsymbol{u}$.

3.5 Camera Upgrading and Refinement

In this step, we estimate the internal calibration as well as the remaining unknown translation component for each camera. By transforming all 2D-3D correspondences into the rig frame, we can leverage data from all framesets.

From the previous step we have estimated the camera rig extrinsics P_i, except for the third component of the translation vector, i.e. $t_{z,i}$. The 3D points mapped into each camera's coordinate system can then be written as

$$\boldsymbol{Z}_{ijp} + t_{z,i} \boldsymbol{e}_z = P_i Q_j \boldsymbol{X}_p \quad \text{where} \quad \boldsymbol{e}_z = \left(0, 0, 1\right)^T . \tag{7}$$

Now we can use the minimal solvers from Kukelova et al. [17] and Larsson et al. [22] for jointly estimating $t_{z,i}$ and the intrinsic parameters. To further remove outlier correspondences, we again use RANSAC to robustly initialize

the parameters. Additionally, we perform non-linear optimization to refine the intrinsics and $t_{z,i}$ by minimizing the reprojection error as

$$\min_{\theta_i, t_z} \sum_{j,p} \rho\left(\left\|\pi_{\theta_i}\left(\boldsymbol{Z}_{ijp} + t_{z,i}\boldsymbol{e}_z\right) - \boldsymbol{x}_{ijp}\right\|^2\right) \quad , \tag{8}$$

where θ_i are the intrinsic parameters and π_{θ_i} denotes the projection into image space. Note that here we use full distortion model instead of pure radial distortion. This is done for each camera individually.

3.6 Final Refinement

In the final step, we optimize all the camera intrinsics, extrinsics and rig poses by minimizing the reprojection error. The optimization problem is

$$\min_{P_i, Q_j, \theta_i} \sum_{i,j,p} \rho\left(\left\|\pi_{\theta_i}(P_i Q_j \boldsymbol{X}_p) - \boldsymbol{x}_{ijp}\right\|^2\right) \quad . \tag{9}$$

Optionally the 3D scene points can be added into optimization problem, in case the scene points are not accurate enough.

4 Implementation

Our implementation is based on the infrastructure-based calibration from the CamOdoCal library [14]. The sparse map is built by COLMAP [33], which uses upright SIFT [25] features and descriptors. For the camera model, pinhole with radial-tangential distortion and pinhole with equidistant distortion [16] are supported to suit different cameras. The optimization is solved with the Levenberg-Marquardt algorithm using the Ceres Library [1] and we use the Cauchy loss with scale parameter 1 as the robust loss function.

Limitations. Note that to robustly fit the rig extrinsics among different framesets requires that at least two cameras in the rig have non-parallel principal axes, otherwise Equation 5 fails to determine the rig pose. However, camera rigs with parallel principal axes, usually stereo camera setups, can be easily calibrated through existing calibration methods. Other cases, *e.g.* two cameras with opposite direction, commonly equipped in mobile phones, can be calibrated by our proposed calibration variant **Inf+RD+RA** described in Section 5.1, which uses pose solvers that can estimate both the poses and intrinsics per frame.

5 Experimental Evaluation

For the experimental evaluation of our method we first consider two different multi-camera systems, one pentagonal camera rig with ten cameras arranged in five stereo pairs (Figure 1a) and a ski helmet with five GoPro Hero7 Black cameras attached (Figure 1b). For the GoPro cameras we record in wide FoV mode, which roughly corresponds to 120° degree horizontal FoV. The cameras on the pentagonal rig have circa 70° horizontal FoV.

Fig. 3. Sample images of the environment. *Left:* Indoor environment in a lab room. *Right:* Outdoor environment on an urban road.

5.1 Evaluation Datasets and Setup

To validate our method we record datasets in both indoor and outdoor environments. See Figure 3 for example images. For each dataset we record a mapping sequence with the GoPro Hero Black 7 in linear mode[5], calibration sequences with both the pentagonal rig and the GoPro helmet, and Aprilgrid sequences to allow for offline calibration and validation. We use the calibration toolbox Kalibr [27] on the Aprilgrid datasets to create a *ground-truth* calibration for comparison. As far as we know there is no competing method that performs infrastructure-based multi-camera calibration with unknown intrinsic parameters. We augment the original pipeline from Heng et al. [12] with radial distortion solvers from Larsson et al. [22] as candidates to join the comparison. In particular, we compare the following approaches:

- **Inf+K.** The infrastructure-based method from Heng et al. [12].
- **Inf+K+RI.** Same as Inf+K but with refinement of the intrinsic parameters during the final bundle adjustment.
- **Inf+RD.** We replace the P3P solver in [12] with the pose solvers from Larsson et al. [22] which also estimate distortion parameters and focal length.
- **Inf+RD+RA.** We add a robust rig averaging similar to Section 3.3.
- **Inf+1DR+RA.** The proposed pipeline as described in Section 3.2-3.6 which delays estimation of the intrinsic parameters using 1D radial cameras.

Note that **Inf+K** and **Inf+K+RI** use the intrinsic parameters from running Kalibr on the Aprilgrid images and join the competition as references. To evaluate the resulting calibrations we robustly align the estimated camera rigs with the camera rigs obtained from Kalibr [27] and compute the difference in the rotations (degrees) and camera centers (centimetres). To evaluate the intrinsic parameters we validate the calibration on the Aprilgrid datasets and report the average reprojection error (pixels).

5.2 Calibration Accuracy and Run-Time on Full Image Sequence

First we aim to evaluate the accuracy of the calibrations by running the methods on the entire calibration sequences. See Figure 4 for a visualization of camera poses recovered in the *Outdoor* dataset. The results can be found in Table 1. We can see that, using infrastructure-based calibration methods, we are able to

[5] Linear mode provides in-camera undistorted images with a reduced FoV.

Table 1. Evaluation of calibration accuracy The errors are with respect to the calibration obtained from the Aprilgrid datasets with Kalibr [27]. Note that **Inf+K** and **Inf+K+RI** use the ground-truth intrinsic parameters as input.

Inf+	**K**	**K+RI**	**RD**	**RD+RA**	**1DR+RA**
GoPro Helmet / Indoor					
Reproj. error (px)	0.283	0.270	0.526	0.412	0.270
Rot. error (degree)	0.193	0.320	0.328	0.319	0.321
Trans. error (cm)	0.780	0.418	0.430	0.435	0.426
GoPro Helmet / Outdoor					
Reproj. error (px)	0.339	0.337	0.337	0.336	0.337
Rot. error (degree)	0.141	0.188	0.187	0.187	0.187
Trans. error (cm)	0.642	0.392	0.385	0.390	0.384
Pentagonal / Indoor					
Reproj. error (px)	0.230	0.281	0.280	0.308	0.282
Rot. error (degree)	0.293	0.548	0.545	0.543	0.543
Trans. error (cm)	1.316	0.366	0.372	0.377	0.376
Pentagonal / Outdoor					
Reproj. error (px)	0.198	0.268	0.268	0.263	0.271
Rot. error (degree)	0.295	0.570	0.566	0.568	0.567
Trans. error (cm)	2.217	0.441	0.419	0.417	0.423

Fig. 4. Experiments in outdoor urban environment. *Left:* The sparse reconstruction from COLMAP [33] with mapping sequence shown in red. *Middle:* The same scene with frames used for calibration in red. *Right:* Aerial view of the scene.

obtain similar quality results as classical Aprilgrid based methods. In this case, the three methods **Inf+RD**, **Inf+RD+RA**, and **Inf+1DR+RA** all had very similar performance. Note also that the ground truth we are comparing to is not necessarily perfect. In practice, we find that with similar datasets recorded at the same time, the extrinsic results differ up to 0.3° and 0.5 centimeters.

For run-time, we run our method on a DELL Laptop equipped with 16 GB RAM, an i7-9750H CPU and a GTX1050 GPU. A comparison of the processing time of each pipeline is shown in Table 2. Our method **Inf+1DR+RA** takes a similar amount of time while removing the need for pre-calibration required by **Inf+K+RI**, and runs much faster than the pattern-based method **Kalibr**.

5.3 Evaluation of Robustness on Shorter Image Sequences

In the previous section we saw that if we have enough data we are able to achieve high-quality calibration results. In this section we instead evaluate the robustness of the method when input data is more limited. For many applications this is an important scenario since you might want to find the camera calibration as quickly

Table 2. Run-Time Comparison. Table lists the average runtime (in minutes) for different methods on calibration sequences with 500 framesets. The runtime for Inf+K+RI and Inf+1DR+RA consists of indoor/outdoor cases.

Runtime (min)	**Inf+K+RI**	**Inf+1DR+RA**	**Kalibr**
GoPro Helmet	9.5 / 11.3	10.9 / 12.2	24.5
Pentagonal	7.0 / 9.6	11.0 / 15.4	113.0

Table 3. Comparison of robustness for shorter image sequences. Table shows the percentage of sequences which we were able to estimate a complete frameset and the percentage of sequences of sequences that were accurately calibrated. A good calibration is defined in Section 5.3.

	Inf+	**RD**	**RD+RA**	**1DR+RA**
GoPro Helmet / Indoor	Complete	54.5	**98.3**	**98.3**
	Good	44.9	75.6	**79.0**
GoPro Helmet / Outdoor	Complete	67.6	97.7	**98.3**
	Good	38.1	45.5	**48.3**
Pentagonal / Indoor	Complete	31.9	68.4	**69.0**
	Good	23.0	43.1	**44.4**
Pentagonal / Outdoor	Complete	28.8	79.2	**80.5**
	Good	21.1	38.3	**41.5**

as possible to enable other tasks which depend on knowing the camera calibration. To perform the experiment we select multiple sub-sequences and try to calibrate from these. For each sequence we select 10 framesets which approximately differ by one second (the datasets were captured at normal walking speed). Table 3 shows the percentage of frames where the calibration-methods were able to calibrate the complete rig, as well as the percentage of sequences which gave good calibrations (defined as rotation error below 1 degree and translation below 1 cm for indoor and 2 cm for outdoor). The total number of sequences were 313 (penta) and 173 (GoPro). Table 3 shows the superior robustness of our approach.

5.4 Evaluation of Initial Estimates

In this section we evaluate the effect of delaying the estimation of the intrinsic parameters on the quality of the initial estimates, i.e. before running bundle adjustment. Similar to the evaluation for robustness in Section 5.3, we run the different methods on multiple sub-sequences and evaluate the extrinsics error of the initial estimates. A qualitative example of the extrinsics is shown in Figure 5 (Left) and it is obvious that the extrinsics estimate for **Inf+1DR+RA** is much better and almost close to the final result. Figure 5 (Right) shows the distribution of the extrinsics error for both methods, where **Inf+1DR+RA** outperforms **Inf+RD+RA** especially in position error. However, as shown in Table 3 most of these initial errors can be recovered in the final refinement.

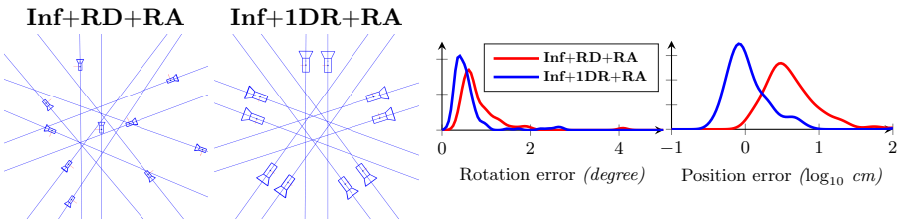


Fig. 5. *Left:* Qualitative example of rig initializations before final refinement. *Right:* Distribution of rotation and translation errors before final refinement.

(a) (b) (c) (d)

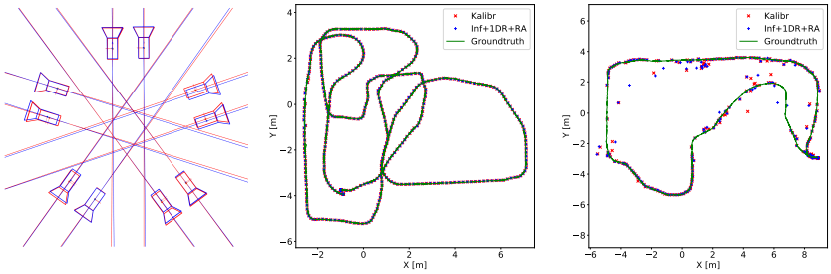
Fig. 6. Results on RobotCar datasets. The extrinsics for out method(blue) and groundtruth(red) are plotted in (a). To validate the intrinsics, the raw image (b) is undistorted using our calibrated results (c) and groundtruth parameters (d).

5.5 Evaluation on RobotCar Dataset

In addition to the experiments mentioned above, we evaluate our calibration method on the public benchmark RobotCar Dataset [26]. We select a short sequence of 30 seconds from the 2014/12/16 datasets (frame No.500 to frame No.900) recorded in the morning to calibrate the three Grasshopper2 cameras pointing left, back and right respectively. The map and calibration groundtruth is obtained from the RobotCar Seasons Dataset [32]. The calibration takes only 3 minutes on a normal PC and the extrinsic results are shown in Figure 6(a). The position error is 1.04cm and rotation error is 0.213°. To validate the intrinsic parameters, we compare the results directly from undistorting the raw image Figure 6(b). Figure 6(c) and Figure 6(d) are the undistorted image for our method and the groundtruth respectively. Although this benchmark is designed for visual localization and place recognition algorithms under changing conditions, we show our method robustly and accurately estimates the camera calibration parameters even with real vehicle vision data in urban roads.

5.6 Application: Robot Localization in a Garden

Finally we evaluate our proposed framework in a real robotics application, namely localization in an outdoor environment. We attach the pentagonal rig to a small robot which autonomously navigates in a garden. We record several datasets of the robot driving around in the garden. From one of the recordings we build a map using the calibration obtained from Aprilgrid calibration



(a) Extrinsics (b) Easy case (c) Hard case

Fig. 7. Results in gardening datasets. The extrinsics are plotted in (a). (b) shows the localization trajectory of an easy dataset and (c) a hard one. The Kalibr results are indicated by red and our method by blue.

with Kalibr. We then calibrate the camera rig using one of the other datasets and evaluate localization performance on the rest of the datasets. The position of the robot is tracked with a TopCon laser tracker yielding accurate position used as groundtruth. The plot of Kalibr extrinsics and results from our results shown in Figure 7(a) confirms the high accuracy of our calibration method. In Figure 7(b) and Figure 7(c) we plot the localized trajectory of two different localization datasets using the calibration results of Kalibr and our method. The median position errors for the two sequences are 3.56 cm and 9.22 cm using results from the proposed method, and 3.77 cm and 9.67 cm using calibration with Kalibr. Using a calibration estimated from the map we are able to achieve slightly higher accuracy for localization compared to the pattern-based approach.

6 Conclusions

We have proposed a method for complete calibration, both intrinsic and extrinsic, of multi-camera systems. Due to the use of natural scene features, our calibration method can be used in any arbitrary indoor and outdoor environments without the aid of other calibration patterns or setups. The extensive experiments and real case application demonstrate the high accuracy, efficiency and robustness of our proposed calibration method. Given the practical usefulness of our approach, we expect it to have large impact in the robotics and autonomous vehicle community.

Acknowledgement This work was supported by the Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots), the Chalmers AI Research Centre (CHAIR) (VisLocLearn), OP VVV project Research Center for Informatics No. CZ.02.1.01/0.0/0.0/16_019/0000765, and EU Horizon 2020 research and innovation program under grant No. 688007 (TrimBot2020). Viktor Larsson was supported by an ETH Zurich Postdoctoral Fellowship.

References

1. Agarwal, S., Mierle, K., et al.: Ceres solver, 2013. URL http://ceres-solver. org (2018)
2. Arth, C., Wagner, D., Klopschitz, M., Irschara, A., Schmalstieg, D.: Wide area localization on mobile phones. In: 2009 8th ieee international symposium on mixed and augmented reality. pp. 73–82. IEEE (2009)
3. Bujnak, M., Kukelova, Z., Pajdla, T.: A general solution to the p4p problem for camera with unknown focal length. In: Computer Vision and Pattern Recognition (CVPR) (2008)
4. Camposeco, F., Sattler, T., Pollefeys, M.: Non-parametric structure-based calibration of radially symmetric cameras. In: International Conference on Computer Vision (ICCV) (2015)
5. Carrera, G., Angeli, A., Davison, A.J.: Slam-based automatic extrinsic calibration of a multi-camera rig. In: International Conference on Robotics and Automation (ICRA) (2011)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
7. Geppert, M., Liu, P., Cui, Z., Pollefeys, M., Sattler, T.: Efficient 2d-3d matching for multi-camera visual localization. In: International Conference on Robotics and Automation (ICRA) (2019)
8. Govindu, V.M.: Robustness in motion averaging. In: Asian Conference on Computer Vision (ACCV) (2006)
9. Haralick, B.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. International journal of computer vision **13**(3), 331–356 (1994)
10. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
11. Heng, L., Choi, B., Cui, Z., Geppert, M., Hu, S., Kuan, B., Liu, P., Nguyen, R., Yeo, Y.C., Geiger, A., et al.: Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In: International Conference on Robotics and Automation (ICRA) (2019)
12. Heng, L., Furgale, P., Pollefeys, M.: Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig. Journal of Field Robotics **32**(5), 775–802 (2015)
13. Heng, L., Lee, G.H., Pollefeys, M.: Self-calibration and visual slam with a multi-camera system on a micro aerial vehicle. Autonomous robots **39**(3), 259–277 (2015)
14. Heng, L., Li, B., Pollefeys, M.: Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In: International Conference on Intelligent Robots and Systems (IROS) (2013)
15. Josephson, K., Byrod, M.: Pose estimation with radial distortion and unknown focal length. In: Computer Vision and Pattern Recognition (CVPR) (2009)
16. Kannala, J., Brandt, S.S.: A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. Trans. Pattern Analysis and Machine Intelligence (PAMI) **28**(8), 1335–1340 (2006)
17. Kukelova, Z., Bujnak, M., Pajdla, T.: Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In: International Conference on Computer Vision (ICCV) (2013)

18. Kukelova, Z., Heller, J., Fitzgibbon, A.: Efficient intersection of three quadrics and applications in computer vision. In: Computer Vision and Pattern Recognition (CVPR) (2016)
19. Kumar, R.K., Ilie, A., Frahm, J.M., Pollefeys, M.: Simple calibration of non-overlapping cameras with a mirror. In: Computer Vision and Pattern Recognition (CVPR) (2008)
20. Larsson, V., Kukelova, Z., Zheng, Y.: Making minimal solvers for absolute pose estimation compact and robust. In: International Conference on Computer Vision (ICCV) (2017)
21. Larsson, V., Kukelova, Z., Zheng, Y.: Camera pose estimation with unknown principal point. In: Computer Vision and Pattern Recognition (CVPR) (2018)
22. Larsson, V., Sattler, T., Kukelova, Z., Pollefeys, M.: Revisiting radial distortion absolute pose. In: International Conference on Computer Vision (ICCV) (2019)
23. Li, B., Heng, L., Koser, K., Pollefeys, M.: A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In: International Conference on Intelligent Robots and Systems (IROS) (2013)
24. Liu, P., Geppert, M., Heng, L., Sattler, T., Geiger, A., Pollefeys, M.: Towards robust visual odometry with a multi-camera system. In: International Conference on Intelligent Robots and Systems (IROS) (2018)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) **60**(2), 91–110 (2004)
26. Maddern, W., Pascoe, G., Gadd, M., Barnes, D., Yeomans, B., Newman, P.: Real-time kinematic ground truth for the oxford robotcar dataset. arXiv preprint arXiv: 2002.10152 (2020), `https://arxiv.org/pdf/2002.10152`
27. Maye, J., Furgale, P., Siegwart, R.: Self-supervised calibration for robotic systems. In: 2013 IEEE Intelligent Vehicles Symposium (IV). pp. 473–480. IEEE (2013)
28. Olson, E.: Apriltag: A robust and flexible visual fiducial system. In: International Conference on Robotics and Automation (ICRA) (2011)
29. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. In: Scandinavian Conference on Image Analysis (SCIA) (2011)
30. Penate-Sanchez, A., Andrade-Cetto, J., Moreno-Noguer, F.: Exhaustive linearization for robust camera pose and focal length estimation. Trans. Pattern Analysis and Machine Intelligence (PAMI) **35**(10), 2387–2400 (2013)
31. Robinson, A., Persson, M., Felsberg, M.: Robust accurate extrinsic calibration of static non-overlapping cameras. In: International Conference on Computer Analysis of Images and Patterns (CAIP) (2017)
32. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6dof outdoor visual localization in changing conditions. In: Computer Vision and Pattern Recognition (CVPR) (2018)
33. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Computer Vision and Pattern Recognition (CVPR) (2016)
34. Schwesinger, U., Bürki, M., Timpner, J., Rottmann, S., Wolf, L., Paz, L.M., Grimmett, H., Posner, I., Newman, P., Häne, C., et al.: Automated valet parking and charging for e-mobility. In: Intelligent Vehicles Symposium (IV). IEEE (2016)
35. Sorkine-Hornung, O., Rabinovich, M.: Least-squares rigid motion using svd. Computing **1**(1) (2017)
36. Strisciuglio, N., Tylecek, R., Blaich, M., Petkov, N., Biber, P., Hemming, J., van Henten, E., Sattler, T., Pollefeys, M., Gevers, T., et al.: Trimbot2020: an outdoor robot for automatic gardening. In: ISR 2018; 50th International Symposium on Robotics. pp. 1–6. VDE (2018)

37. Sturm, P.F., Maybank, S.J.: On plane-based camera calibration: A general algorithm, singularities, applications. In: Computer Vision and Pattern Recognition (CVPR) (1999)
38. Thirthala, S., Pollefeys, M.: Radial multi-focal tensors. International Journal of Computer Vision (IJCV) **96**(2), 195–211 (2012)
39. Triggs, B.: Camera pose and calibration from 4 or 5 known 3d points. In: International Conference on Computer Vision (ICCV) (1999)
40. Tsai, R.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal on Robotics and Automation **3**(4), 323–344 (1987)
41. Wu, C.: P3. 5p: Pose estimation with unknown focal length. In: Computer Vision and Pattern Recognition (CVPR) (2015)
42. Zhang, Q., Pless, R.: Extrinsic calibration of a camera and laser range finder (improves camera calibration). In: International Conference on Intelligent Robots and Systems (IROS) (2004)
43. Zheng, Y., Sugimoto, S., Sato, I., Okutomi, M.: A general and simple method for camera pose and focal length determination. In: Computer Vision and Pattern Recognition (CVPR) (2014)