

Non-Parametric Structure-Based Calibration of Radially Symmetric Cameras

Federico Camposeco, Torsten Sattler, Marc Pollefeys
Department of Computer Science, ETH Zürich, Switzerland

{federico.camposeco, torsten.sattler, marc.pollefeys}@inf.ethz.ch

Abstract

We propose a novel two-step method for estimating the intrinsic and extrinsic calibration of any radially symmetric camera, including non-central systems. The first step consists of estimating the camera pose, given a Structure from Motion (SfM) model, up to the translation along the optical axis. As a second step, we obtain the calibration by finding the translation of the camera center using an ordering constraint. The method makes use of the 1D radial camera model, which allows us to effectively handle any radially symmetric camera, including non-central ones. Using this ordering constraint, we show that we are able to calibrate several different (central and non-central) Wide Field of View (WFOV) cameras, including fisheye, hypercatadioptric and spherical catadioptric cameras, as well as pinhole cameras, using a single image or jointly solving for several views.

1. Introduction

Recently, cameras with a wide field of view (WFOV) such as fisheye and omnidirectional cameras are starting to become more and more popular. Due to their ability to observe a large portion of the scene, using WFOV cameras is advantageous for 3D computer vision tasks such as the precise camera tracking that is done as part of visual navigation for robots and autonomous vehicles. At the same time, action cameras with WFOV such as the GoPro Hero are widely used. Similarly, camera mounts for mobile devices that enable them to take panoramic images, *e.g.*, using a catadioptric lens, are becoming more frequent. As a result, more and more WFOV images are becoming available on photo sharing websites such as Flickr and Picasa. In the case of Structure-from-Motion (SfM) from photo community collections, these photos could be particularly helpful to strengthen the overall reconstruction as they provide constraints to many other cameras. However, they are typically discarded in practice due to the challenge of automatically calibrating these cameras.

In this paper, we present a novel method to automatically calibrate WFOV cameras from 2D-3D matches established

between features extracted in their images and 3D points in a SfM reconstruction using image-based localization methods. Given a partial reconstruction obtained from regular images, our method can thus be used to calibrate and then insert WFOV into a SfM model to strongly link together different parts of the scene which are all visible in these photos and thus improve the quality of the reconstruction.

Our method is based on the 1D radial camera model [15–17], which can be used to describe any type of camera with radial distortion, including pinhole, fisheye, and non-central cameras such as catadioptric lenses, as long as the center of distortion is known. We combine this model with a non-parametric intrinsic calibration to obtain an extremely powerful calibration method that is capable of calibrating a wide range of camera types. Our method consists of two steps. In the first step, the extrinsic calibration is computed from the 2D-3D matches up to the position of the camera center (or camera centers in the case of non-central cameras) along the optical axis. This can be done efficiently using a linear 7-point solver inside a RANSAC loop [5]. Given the partial extrinsic calibration, we employ a novel ordering constraint on the opening angle of the viewing rays corresponding to the 2D features to estimate the remaining extrinsic parameter in the case of a central camera. Since we are considering radially symmetric cameras, fixing the camera center directly provides the intrinsic calibration as it provides a mapping from image positions to viewing rays. A slightly generalized version of this constraint enables us to obtain a more accurate calibration from multiple photos taken with the same camera. The constraint can be further generalized to also handle non-central cameras, both when only a single or multiple photos are available. In addition to its generality, our novel ordering constraint enables us to formulate the second part of the calibration process as a convex optimization program. We experimentally demonstrate the accuracy of our calibration pipeline for a wide range of cameras. Additionally, we make the source code of our method available [1].

The remainder of the paper is structured as follows. Sec. 2 discusses related work. Sec. 3 reviews the 1D radial camera model. Sec. 4 introduces our novel ordering constraint and derives our calibration method for the case of central cameras. The extension to non-central cameras

is then provided in Sec. 5. Finally, Sec. 6 experimentally evaluates our approach on both synthetic and real data.

2. Related Work

Recently, there has been some interest in non-parametric calibration of radially symmetric cameras. For instance, in [8, 11] a non-iterative, non-parametric method for calibration of fisheye cameras is proposed. While they claim that they can do without, the method is only tested using a calibration pattern since their auto-calibration needs several correspondences and is very sensitive to noise.

Similar to this, and much more related to our work, in [15–17] Thirithala and Pollefeys developed the 1D Radial Camera model. They propose a multi-focal tensor able to auto-calibrate any radially symmetric set of cameras (including non-central), and also produce a non-parametric calibration. In this work we use this same model, however, in contrast to their approach, we develop a more general and robust geometric ordering constraint to calibrate with.

Several other methods make use of either a specific scene structure (enough straight lines) or use calibration objects to compute their calibration, *e.g.* [7, 13]. In [14] an approach for self-calibration of radially symmetric cameras is presented. They develop a plumb-line (using the fact that straight lines in space must project into straight lines in the image) and plane-based methods. In [6], they rely on the observation of at least three lines to compute the parameters of a para-catadioptric system. In contrast to these, we remain flexible by enforcing no requirements on the scene or a calibration object.

For our method we first estimate the (partial) extrinsics (*i.e.* the pose) w.r.t. a SfM model. There have been numerous advances geared toward pose estimation in the absence of calibration. For example, Kukulova *et al.* propose a 5-point pose estimator with unknown radial distortion and focal length [10]. They employ an idea similar to the 1D Radial Camera Model in order to efficiently obtain a pose and 3-parameter calibration. Also, minimal 4-point solvers for this same case are presented in [3, 9] in which they too make use of a parametric model to obtain a solution. In contrast to these methods, we do not restrict the camera to be able to be represented by a specific mode. Furthermore, we extend our method to seamlessly aggregate data from several views of the same camera to increase the accuracy of the calibration and to better handle non-central systems.

3. The 1D Radial Camera Model

In order to calibrate any type of radially symmetric camera, this paper builds on the 1D radial camera model. For a more in-depth analysis the reader is referred to [17], however we briefly review it in this section for completeness.

Let C_d be the center of distortion for a camera exhibiting radial distortion. Let \mathbf{x}_u denote the undistorted projection of a 3D point \mathbf{X} onto the camera’s image. As illustrated

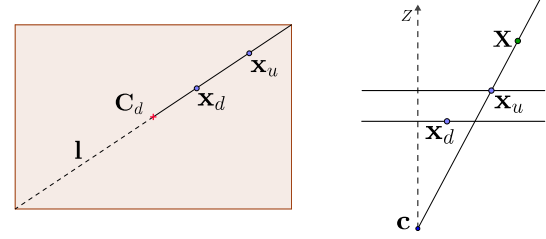


Figure 1: Radial 1D Camera. Image plane (left) and top view (right) of the projection of point \mathbf{X} .

in Fig. 1, applying radial distortion maps \mathbf{x}_u to a point on the *radial line* $l = \mathbf{x}_u \times C_d$ through the center of distortion and the undistorted image coordinates. Similarly, \mathbf{x}_u lies on the line $l = \mathbf{x}_d \times C_d$ defined by the distorted measurement \mathbf{x}_d . Instead of explicitly modeling the radial distortion, the 1D radial camera model defines a projection up to radial distortion. This is expressed as a mapping $\mathbb{P}^3 \rightarrow \mathbb{P}^1$ that associates each 3D point to a line $\lambda l = P_r \mathbf{X}$. The projection matrix $P_r \in \mathbb{R}^{2 \times 4}$ relates to the first two rows of the camera pose $(R|t)$ by

$$P_r = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 & t_x \\ \mathbf{R}_2 & t_y \end{bmatrix}, \quad (1)$$

where \mathbf{R}_i is the i -th row of the rotation matrix R . Notice that the 1D radial camera model, other than unit aspect ratio, makes no assumption on the internal calibration of the camera. In fact, it describes both central and non-central cameras as long as there is a single center of distortion, including pinhole, fisheye, and catadioptric cameras.

As in [17], we assume that C_d is known, enabling us to center the image around C_d . For most cameras, the center of the image is a reasonable approximation for C_d . Alternatively, it can be estimated using the visible rim of the catadioptric mirror or the edge of the fisheye lens (*c.f.* Fig. 8).

4. Calibrating Central Radially Symmetric Cameras

Given a 3D model of the scene, our goal is to estimate both the extrinsic and intrinsic calibration from 2D-3D correspondences $(\mathbf{x}^i, \mathbf{X}^i)$ between positions in an image taken with a radially symmetric camera and the model. Since the projection matrix P_r does not depend on the intrinsic calibration, we use a two-stage approach. In the first stage, we use RANSAC [5] to estimate the extrinsic calibration up to an unknown translation along the optical axis. The inliers to the pose are then used to non-parametrically estimate the intrinsic calibration. Sec. 4.1 details the computation of the partial extrinsic calibration. In Sec. 4.2, we then derive a novel ordering constraint that allows us to compute the intrinsic calibration by solving a convex optimization problem. Sec. 4.3 shows that the same constraint can be used to calibrate a camera from multiple images. We show in Sec. 5

how to extend our approach to handle non-central radially symmetric cameras.

4.1. Partial Extrinsic Calibration

Let $\mathbf{x}_d = (x_d, y_d)^\top$ be the position of a distorted measurement in a coordinate system centered at the center of distortion. The radial line of the i -th correspondence $(\mathbf{x}_d^i, \mathbf{X}^i)$ can then be expressed as

$$\mathbf{l}_i = \begin{bmatrix} -y_d^i/x_d^i \\ 1 \end{bmatrix} = \begin{bmatrix} l^i \\ 1 \end{bmatrix} = \mathbf{P}_r \mathbf{X}^i. \quad (2)$$

By multiplying \mathbf{l}_i by its perpendicular vector $(1, -l^i)^\top$, we obtain

$$\mathbf{P}_{r1} \cdot \mathbf{X}^i - l^i (\mathbf{P}_{r2} \cdot \mathbf{X}^i) = 0, \quad (3)$$

where \mathbf{P}_{rn} represents the n -th row of the matrix \mathbf{P}_r . Thus, each 2D-3D correspondence gives us one constraint. Since \mathbf{P}_r is only defined up to scale, it can be estimated linearly from seven matches by rearranging (3). Once we have an estimate for \mathbf{P}_r , we can recover the full rotation matrix \mathbf{R} by exploiting the fact that rotation matrices are orthonormal matrices with determinant one (c.f. (1)).

Given a set of 2D-3D correspondences, we estimate \mathbf{P}_r by using the 7-point solver inside a RANSAC loop. In order to distinguish between inliers and outliers, we measure the subtended angle between the predicted and the observed radial lines $\hat{\mathbf{l}}_i = \mathbf{P}_r \mathbf{X}^i$ and \mathbf{l}_i . A match is considered to be an inlier if the angle is below a given threshold σ (set to 1° in our experiments).

Notice that \mathbf{P}_r has only five degrees of freedom in total: Three degrees of freedom for the rotation and two degrees of freedom for the partial translation t_x, t_y . Thus, the linear 7-point solver is non-minimal. If a minimal solver is required due to a high outlier ratio, the 5-point approach from [10] can be used, which requires solving a fourth degree polynomial in a single variable.

4.2. Non-Parametric Intrinsic Calibration

The intrinsic calibration of a camera defines a mapping $\mathbf{r}(\mathbf{x})$ from image coordinates to viewing rays. In the case of radial symmetry, the angle θ between the ray $\mathbf{r}(\mathbf{x})$ and the optical axis for all positions \mathbf{x} with the same distance to the center of distortion, i.e., $\|\mathbf{x}\|_2 = r$, is constant. Consequently, the point \mathbf{X} projecting to \mathbf{x} has to lie on a cone along the optical axis with opening angle θ (c.f. Fig. 1). For two points $\mathbf{x}_{r1}, \mathbf{x}_{r2}$ with radii $r_1 < r_2$, we have $\theta_{r1} < \theta_{r2}$. In the following, we derive a geometric constraint from this observation from which we explicitly compute the mapping from radii to opening angles.

Given \mathbf{P}_r (c.f. Sec. 4.1), the transformation of the 3D points from the global into the local coordinate system of the camera is defined up to a translation along the optical axis. Using $\mathbf{R}, \mathbf{t}' = (t_x, t_y, 0)^\top$, we obtain an intermediate

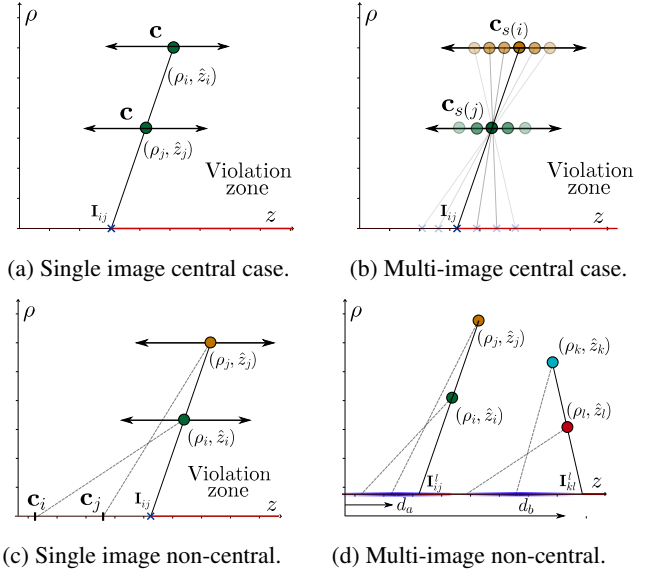


Figure 2: Ordering constraints for different systems. The abscissa for each figure, labeled z , are aligned with the optical axis of the camera. For Figures a, b and c, $r_d^i > r_d^j$.

coordinate system in which the unknown translation corresponds to the position of the camera center \mathbf{c} on the optical axis. We notice that fixing \mathbf{c} defines the opening angle θ_i for a given 3D point $\mathbf{R}\mathbf{X} + \mathbf{t}'$ in the intermediate coordinate system. Thus, fixing \mathbf{c} fully defines the intrinsic calibration of the camera.

A geometric ordering constraint on the camera center. We express each point $(x_i, y_i, z_i)^\top = (x_i, y_i, z_i - \mathbf{c})^\top$ as $(\varphi_i, \rho_i, \hat{z}_i)^\top$ in a cylindrical coordinate system (c.f. Fig. 2). Since we consider radially symmetric cameras, we can drop the angle φ of the point around the optical axis from the notation and only consider the distance of the 3D point to the optical axis ρ and its depth z . Consider two 3D points $p_i = (\rho_i, \hat{z}_i), p_j = (\rho_j, \hat{z}_j)$ in the intermediate frame (c.f. Fig. 2a), corresponding to radii r_d^i and r_d^j of the distorted image measurements $\mathbf{x}_d^i, \mathbf{x}_d^j$. Without loss of generality, let $\rho_i \neq \rho_j$ and let

$$\mathbf{l}_{ij} = (\hat{z}_j \rho_i - \hat{z}_i \rho_j) / (\rho_i - \rho_j) \quad (4)$$

be the intersection of the 2D line containing the point pair with the optical axis z . In the case where $r_d^i = r_d^j$, \mathbf{l}_{ij} corresponds to the camera center \mathbf{c} . Unfortunately, it is rather unlikely to find two features with exactly the same radius. In [15], the authors propose to fit a line through 3D points corresponding to similar radii to obtain a camera center per radius. In contrast, we use an ordering constraint to directly obtain a \mathbf{c} as explained below.

Without loss of generality, assume that $r_d^i > r_d^j$ and thus $\theta_i > \theta_j$. In the case that $\rho_i > \rho_j$, it follows that $\mathbf{c} < \mathbf{l}_{ij}$ (c.f. Fig. 2a). Similarly, $\rho_i < \rho_j$ yields the constraint $\mathbf{c} > \mathbf{l}_{ij}$.

Thus, for each point pair we get a *one-sided constraint* that restricts the value of \mathbf{c} to lie either to the left or to the right of \mathbf{I}_{ij} . For each constraint we then build a cost function which penalizes a given \mathbf{c} that violates a one-sided constraint by using a piecewise cost function. For $r_d^i > r_d^j$ and $\rho_i > \rho_j$

$$E_{ij}^l(\mathbf{c}; p_i, p_j) = \begin{cases} 0 & \mathbf{c} < \mathbf{I}_{ij} \\ f(\mathbf{I}_{ij} - \mathbf{c}) & \text{otherwise} \end{cases}, \quad (5)$$

which penalizes \mathbf{c} if it is to the *right* of \mathbf{I}_{ij} (c.f. Fig 2a). Here, f is a function depending on the distance between the intersection point and the center \mathbf{c} . For the opposite configuration, either $r_d^i < r_d^j$ or $\rho_i < \rho_j$, we may build a similar cost function which penalizes \mathbf{c} to be to the *left* of \mathbf{I}_{ij} .

Calibration through convex optimization. Using (5) we can then take the sum over all the cost functions

$$\mathbf{E}(\mathbf{c}) = \sum_{\{i,j\} \in L} E_{ij}^l + \sum_{\{i,j\} \in R} E_{ij}^r, \quad (6)$$

where $L = \{\{i,j\} \mid \rho_i > \rho_j \text{ and } r_d^i > r_d^j\}$ and $R = \{\{i,j\} \mid \rho_i > \rho_j \text{ and } r_d^i < r_d^j\}$. If f is chosen to be a convex function, $\mathbf{E}(\mathbf{c})$ will be convex. We can then obtain the camera center, and thus the intrinsic calibration, by optimizing (6) using, e.g., Gradient-Descent.

We choose f to be an $L1$ norm to be robust to outliers, while allowing \mathbf{E} to remain convex. Furthermore, we propose a very simple algorithm for computing (6) when f is a linear function. Since the slope of $\mathbf{E}(\mathbf{c})$ changes only whenever there is an intersection, we may efficiently compute it in two passes. We start by sorting the intersections, such that $\mathbf{I}_k < \mathbf{I}_{k+1}$. On the first pass, from left to right, we deal only with the intersections that constrain \mathbf{c} to be to their left (shown in green in Fig. 3a) and iteratively compute the cost for each intersection. Starting with $\mathbf{E}(\mathbf{I}_0)^l = 0$ we can express the cost of the k_{th} intersection as the cost of the previous intersection plus the *cost increase* of the k violating constraints from \mathbf{I}_{k-1} to \mathbf{I}_k . Since f is a linear function, the latter costs only depend on the distance between the current intersection and the last, i.e.

$$\mathbf{E}(\mathbf{I}_k)^l = \mathbf{E}(\mathbf{I}_{k-1})^l + k f(\mathbf{I}_k - \mathbf{I}_{k-1}). \quad (7)$$

On the second pass we sweep in the opposite direction taking into account the intersections that constrain \mathbf{c} to be to their right. The cost of a given \mathbf{c}_k is obtained by checking its nearest left- and right-constraining intersections and summing their costs $\mathbf{E}(\mathbf{c}_k) = \mathbf{E}^r(\mathbf{I}_k) + \mathbf{E}^l(\mathbf{I}_k)$.

Selecting point pairs. Given N points in an image, it is impractical to exhaustively take all point pairs since the number of pairs is $N(N-1)/2$. Instead, for a given image we want to only operate on a fixed number of pairs. To do so, we to sort the point pairs by their quality, i.e. pairs which yield stable intersections close to \mathbf{c} . For each pair p_{ij} we

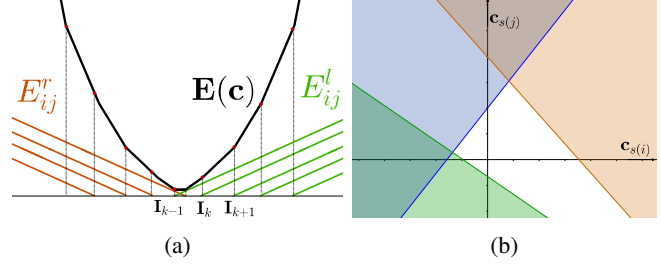


Figure 3: In a we show the single image linear cost function proposed. In the multi-image case, each point pair defines a 2D constraint, depicted in b.

get $\Delta r_{ij} = \|r_d^i - r_d^j\|$ and $\Delta \rho_{ij} = \|\rho_i - \rho_j\|$. First, we discard pairs with $\Delta \rho_{ij}$ less than a given threshold, which takes care of unstable intersections. Then we sort the pairs using Δr_{ij} in ascending order and take only the first N_s (set to 120 in our experiments) pairs of the sorted list.

4.3. Joint Calibration from Multiple Images

The approach presented in Sec. 4.2 essentially determines an interval in which the camera center can lie in. Using more points adds more constraints on this interval, which should lead to more accurate estimates. Synthetic experiments have shown that approximately 250 imaged points are enough to obtain a calibration that achieves less than 1 pixels of RMSE on the reprojected points (c.f. Fig. 5), while using less than 100 points leaves \mathbf{c} very underconstrained and the resulting calibration will be unreliable¹. Naturally, additional points can be obtained by using multiple images for the calibration. Thus, in this section we show that our geometric ordering constraint can easily be extended to allow calibrating a camera from $M > 1$ images.

By expressing 3D points in the ρz -plane we can transform all cameras to a common frame of reference by finding a one-dimensional relative translation between them (c.f. Fig. 2b). This allows us to employ our one-sided constraint to find this relative translation and a joint calibration.

Joint constraint for central cameras. Given two cameras $s(i)$ and $s(j)$ we can express the intersection of any point pair p_{ij} between them as

$$\mathbf{I}_{s(i),s(j)} = (z_j - \mathbf{c}_{s(j)}) \rho_i - (z_i - \mathbf{c}_{s(i)}) \rho_j, \quad (8)$$

where $s(i)$ indicates to which camera the point i corresponds. Notice that this is almost the same as (4), however (8) provides a constraint that now depends on two variables (c.f. Fig. 3b), making the position of one camera dependent on the other. The cost function $\mathbf{E} : \mathbb{R}^M \mapsto \mathbb{R}$ can be also designed as a piecewise function. For the configuration

¹Notice, that toolboxes as the one described in [13] suggest using 6 to 10 images. Assuming a calibration pattern with 48 corners, such methods use up to 480 points.

$r_d^i > r_d^j$ and $\rho_i > \rho_j$ we define

$$E_{ij}^l(\mathbf{c}_{s(i)}, \mathbf{c}_{s(j)}) = \begin{cases} 0 & \mathbf{I}_{s(i),s(j)} < 0 \\ g(\mathbf{c}_{s(i)}, \mathbf{c}_{s(j)}) & \text{otherwise,} \end{cases} \quad (9)$$

where g is a cost function on the distance from the given center pair to the intersection (c.f. Fig. 2b). As with (5), we decide to use the L1 norm as a cost function to remain robust to outliers.

Calibration through convex optimization. Similarly to (6), we take the sum of all relevant point pairs and to get \mathbf{E} , which can be minimized using a convex optimization method. Notice that the selection criteria for point pairs described in Sec. 4.1 applies here as well, since we may aggregate image radii from all cameras into one single sorted list to choose N_s relevant pairs. Finally, we get a calibration by translating the Z coordinate of the points by the camera center that obtained the observation $\theta_i(r_d^i) = \arctan(\rho_i/(z_i - \mathbf{c}_{s(i)}))$.

5. Calibrating Non-Central Cameras

Since we are only dealing with radially symmetric cameras, the centers of the camera can be expressed as a function of the distorted image radius $\mathbf{c} = \mathbf{c}(r_d)$. So, any point $p_i = (\rho_i, \hat{z}_i)$ has $\hat{z}_i = z_i - \mathbf{c}_i$, where $\mathbf{c}_i = \mathbf{c}(r_d^i)$.

Non-central constraint. Any point pair p_{ij} will constrain both centers \mathbf{c}_i and \mathbf{c}_j (see Fig. 2c). Given N image points, we have $N - 1$ constraints for each center we need to estimate. We treat each of the N camera centers as a different view of the scene (i.e. $s(i) = i$) and apply the method described in Sec. 4.3. However, in practice the $N - 1$ constraints might not limit the location of a given center enough, yielding inaccurate results for centers with weak or too few constraints (e.g. for centers that correspond to radii near the edge or center of the image, since these are mostly same-side constraints).

To solve this we propose to impose an *ordering* constraint to the centers. We first sort all the points p_i by their radii such that $r_d^{i-1} < r_d^i < r_d^{i+1}$, which restricts their corresponding centers

$$\mathbf{c}(r_d^{i-1}) < \mathbf{c}(r_d^i) < \mathbf{c}(r_d^{i+1}) . \quad (10)$$

This is sensible given that all radially symmetric non-central systems known to the authors follow this ordering (e.g. spherical catadioptric, para-catadioptric). This constraint can seamlessly be translated into the one-sided constraints (c.f. Fig. 4). We define

$$\mathbf{E}_k^r(\mathbf{c}) = \sum_{k=0}^N (E_{ik}^r(\mathbf{c}) + E_{ik}^l(\mathbf{c})) , \quad (11)$$

i.e. the cost of the k -th point against the rest. Then for a

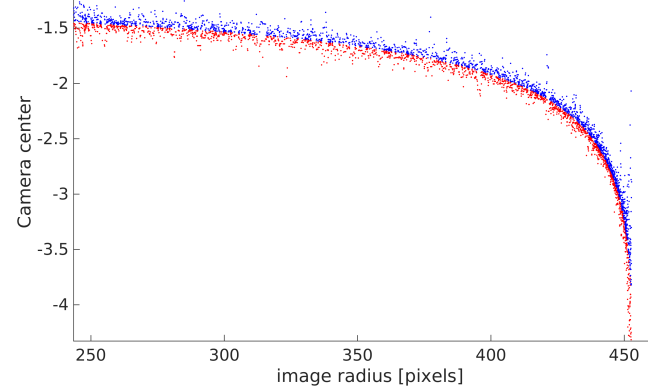


Figure 4: Illustration of the ordering constraint from simulated data. In red, each center is constrained to lie above a certain value, and vice versa for blue. Notice that we may use any of the blue constraints lying to the left of any given pixel radius and vice versa.

center \mathbf{c}_i the cost becomes

$$\mathbf{E}(\mathbf{c}_i) = \sum_{k=0}^i \mathbf{E}_k^r(\mathbf{c}) + \sum_{k=i}^N \mathbf{E}_k^l(\mathbf{c}) , \quad (12)$$

in other words, we use the one-sided *left* constraints of the centers that should be larger than \mathbf{c}_i and the one-sided *right* constraints of those centers that should be smaller than \mathbf{c}_i (see Fig. 2c). Minimizing (12) we get a set of centers which can be used to get the final calibration mapping $\theta_i(r_d^i) = \arctan(\rho_i/(z_i - \mathbf{c}_i))$.

Joint solution for non-central cameras. To get a joint non-central calibration we use a two step procedure. First, we treat each camera as a central system and solve for their joint calibration, which provides us with an estimate of their displacements d_k (see Fig. 2d). We use this to translate all the data points across different views to be on the same frame of reference, i.e. to have a mutually consistent depth. Second, we solve for a single non-central system by treating all the translated points as if they came from a single view. This allows us to keep the number of points needed for a successful calibration relatively low (around 350 points per image, c.f. Fig. 5).

5.1. Refinement and Final Calibration

One of the primary benefits of our method is that we provide a calibration that does not rely on a given parametrization, thus we can accommodate a very wide range of cameras; from planar to catadioptric, central and non-central. However, we wish to refine our obtained solution by removing views and points based on their reprojection errors, and to do so we must find a way to use the obtained mapping. For this we opt to use a *sliding median* [8] of the calibration

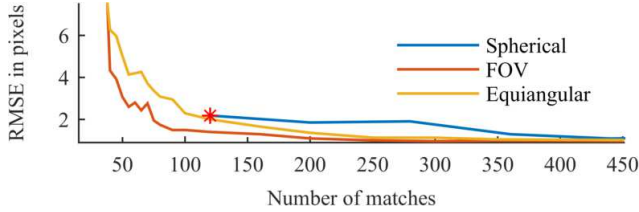


Figure 5: RMSE error when varying the number of matches used to obtain the calibration (pixel σ set to 1.2). The red asterisk indicates that prior to that number of matches, the calibration failed.

data obtained². For all views we compute corresponding reprojection errors and remove those points whose errors rise beyond a certain threshold (set to 5 pixels in our experiments). After this, we get a final set of inliers and recompute the calibration by repeating the corresponding procedure.

6. Experimental Evaluation

To evaluate the proposed method we perform experiments with real and synthetic data. Since one of the strengths of the method is that it can handle a very wide array of cameras, we make a point of trying as many cameras as possible (*c.f.* Fig. 8).

6.1. Synthetic Data

We first carried out experiments on synthetic data to evaluate the performance of our methods. We populate the scene with 320 data points distributed randomly. To simulate the central case, we project the data points into the camera using a pinhole model as well as two well-known fisheye models, the Field of View (FOV) model [4] and the equiangular model. For the non-central case, we chose a spherical catadioptric camera. To compute the reflections we use [2]. Fig. 7 shows the calibration output of the central as well as the non-central simulations, Fig. 6 compares the accuracy of the generated calibrations against those obtained using the toolbox in [13], and Fig. 5 shows the error w.r.t the number of matches used.

As it can be seen from Fig. 7, the results for the central systems perfectly match the ground truth. In Fig. 7c we show the benefit of relaxing the method to handle non-central systems. At the beginning of the curve both orange (central assumption) and blue (non-central) scatter plots match. However, as the non-centrality of the spherical model becomes more significant at higher radii of the simulated image, the deviation is more apparent. In Fig. 7d we show how the accuracy of $c(r_d)$ is affected when we do not enforce the ordering constraint (10). We show the re-

²However, having a calibration that is agnostic to the particular optics of the setup, one is free to use a more sophisticated method to approximate the distortion function.

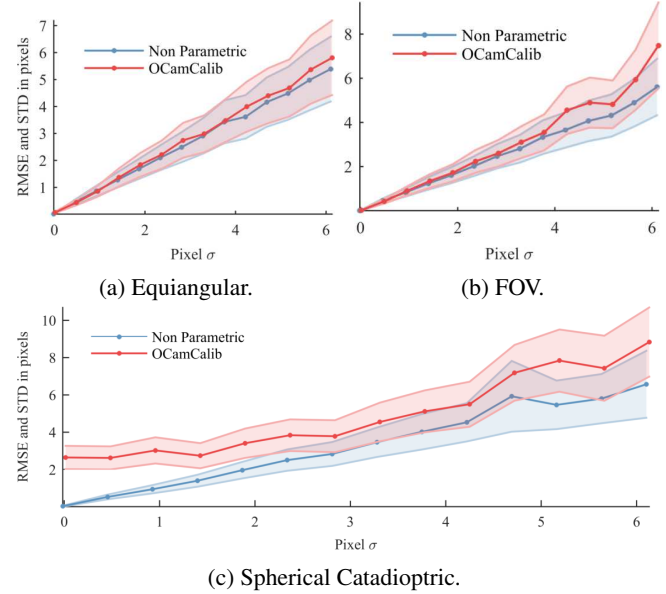


Figure 6: Comparison of the reprojection error and its standard deviation against [13]. To obtain the calibrations, our method used 320 points while the method we compare against used 21 images of a 48-point synthetic calibration pattern. Notice that for Fig. 6c the error is always lower using our calibration since we explicitly support non-central systems.

sulting calibration mappings compared against the ground truth of the simulated data. This is of particular importance for the simulated spherical catadioptric system since with real data we don't have reliable ground truth for $c(r_d)$.

6.2. Real Data

In order to test the flexibility of the method, we tried several different cameras and lenses (*c.f.* Fig. 8): A Nikon D300 coupled with a fisheye lens, a 360One VR catadioptric lens and a spherical catadioptric lens (using a 3-inch steel ball). To assess the performance for the mentioned mobile phone attachments, we run tests using an iPhone 4 with a GoPano catadioptric attachment. Results from other cameras and lenses are provided as supplemental material [1].

Due to the high distortion observed with most of the lenses, we don't obtain a very high number of matches and thus we must use the multi-camera methods. To get more complete calibrations in the catadioptric cases we need to increase the number of matches near the edges of the reflection. To do so, we first use an equiangular calibration: $\theta = k r_d$ where we find k by having $\theta = \pi/2$ map to the largest radii in the image. We use this to warp the image into a cylindrical map which we use to get matches against the SfM model³. Note that this places no restrictions on the

³The toolbox used for comparison had to be manually assisted to get the corners of the calibration pattern for this particularly difficult data.

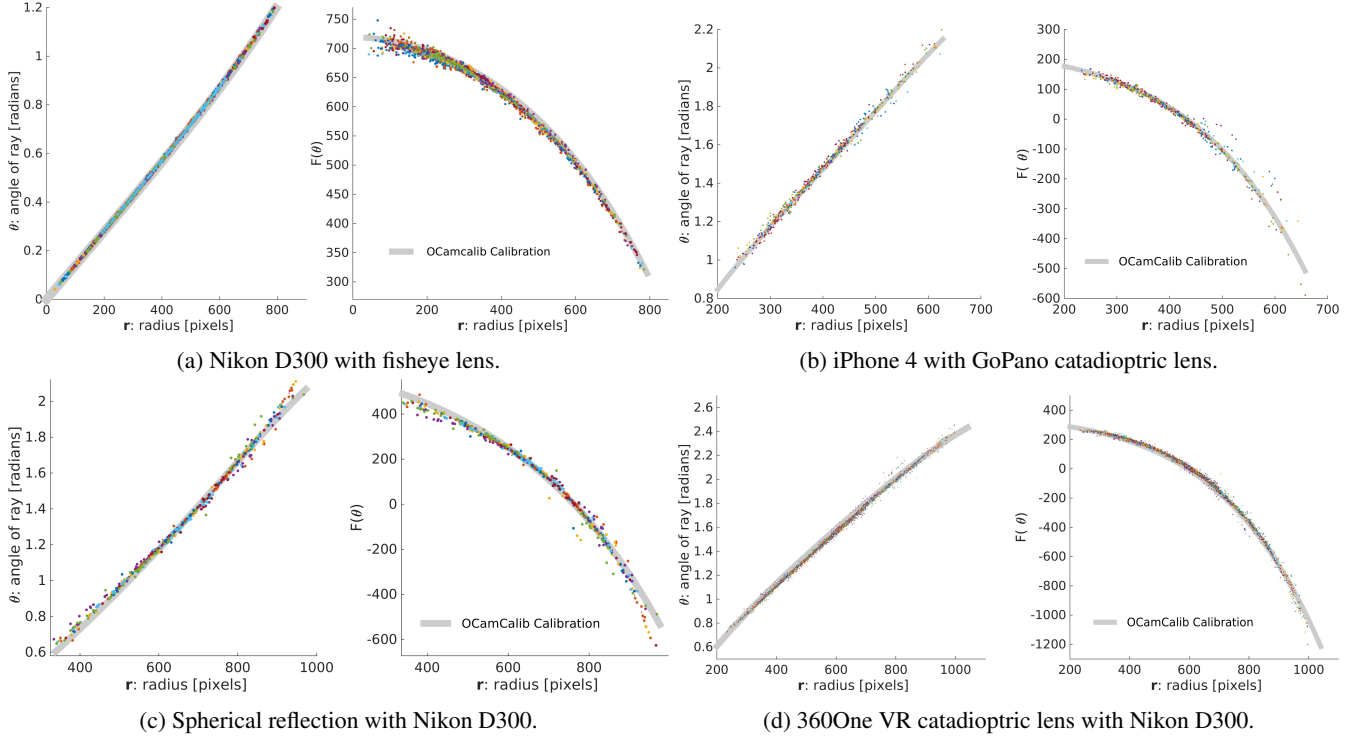


Figure 9: Real calibrations for central and non-central cases. All results shown are compared against the calibration obtained using [13], shown in gray. As it can be seen from all four cameras, our solution accurately match the one from this state-of-the-art calibration toolbox. Calibration points are colored according to the image index used to emphasize the number of images used per camera. To highlight the comparison with the reference calibration, we also plot the distortion function F (the plot on the right for each case) as detailed in [13].

images we are able to handle, since the only assumption, as before, is that the images are radially symmetric.

The SfM model used consists of a large-scale reconstruction, obtained in an outdoor location, *c.f.* Fig. 10. We took several images with each camera type at the same location and obtained putative 3D-2D matches (around 120 for each image). In order to maximize the number of matches obtained, we employed the method proposed in [12], modified to return as many matches as possible. Because of the drastically large distortion, we observed inlier ratios as low as 20%, and thus several images were needed for each camera type (between 20 and 25) from which we were left with approximately 500 points.

In Fig. 9 we show the calibration obtained for a selection of the tested systems. For each case, we obtained calibrations that closely match the calibration computed with the toolbox in [13]. To emphasize the correctness of the obtained calibration, we compare the previously discussed mapping ($\theta(r_d)$) and the function F . This function is described in [13] as the focal length as a function of the image radius, where a point with image coordinates (u, v) can be expressed in the camera frame as $(u, v, F(r_d))^T$. For the fisheye images (*c.f.* Fig. 9a), we got a very high number of matches, since the query images resembled the most to the

images used to construct the SfM model. For this particular model, we have enough data to see that the calibration near the center of the image suffers more than the rest due to weak constraints. However, for the case of the non-central system (*c.f.* Fig. 9c) there is a larger mismatch throughout between our obtained calibration and that of [13] since our method fully supports non-central systems. For the GoPano attachment, as shown in Fig. 9b, we have very few matches near the border and this is reflected in the scattered data points at the end of the curve. Nevertheless, we are overall able to calibrate even such a low-quality lens system.

The reference calibration method additionally computes the refined centers of distortion. However, the fact that our calibration closely matches the reference calibration shows that using the center of the image as the center of distortion is a valid assumption in practice.

7. Conclusion

In this work we presented a novel, flexible, structure-based calibration method for radially symmetric cameras. Indeed, such subset of cameras encompasses most of the systems used nowadays, such as planar, fisheye, catadioptric, WFOV, and so on. We are thus able to handle the cali-

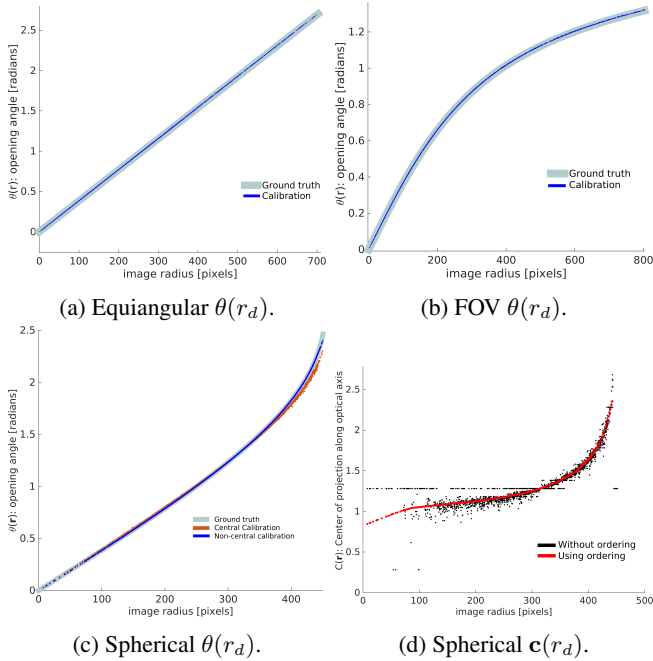


Figure 7: Synthetic calibration $\theta(r_d)$ for the central (a and b) and the non-central cameras c, where in orange we show how a central assumption would not be as accurate. Fig. d shows the estimate of $c(r_d)$ for the non-central case, notice here the effect of employing the ordering constraint (10).

bration of several systems under a single framework which would usually require several different calibration methods.

Furthermore, WFOV imagery is becoming more ubiquitous by products such as the GoPro and WFOV lens attachments for mobile phones. With our method, we can make use of this increasingly popular image modality to augment and strengthen SfM models produced from online photocollections. Online WFOV images can be thus calibrated and inserted as part of an existing SfM model. This would be greatly beneficial for the quality of the model since these type of images can strongly link several parts of the model which were never visible before from the same view.

The described calibration method makes use of the 1D Radial Camera [15] to decouple the estimation of the extrinsic (up to translation along the optical axis) and intrinsic calibration of any radially symmetric camera into two separate steps. In particular, the partial extrinsics are obtained via a linear 7-point solver in conjunction with RANSAC, while the computation of the intrinsics is carried out minimizing an outlier-robust convex cost function for both the single and the multi-image case.

We compute the calibration as a mapping from distorted image radii into the angle of its corresponding 3D ray w.r.t. the optical axis of the camera. By opting for a non-parametric calibration we are able to maintain a very broad compatibility with any camera that fits the 1D Radial model.

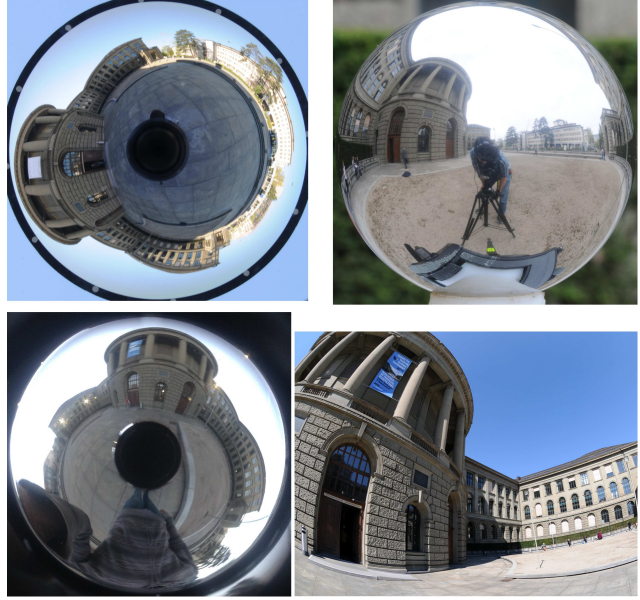


Figure 8: Sample of the data used for the experiments for the same outdoor location. From top to bottom, left to right; the 360One catadioptric lens, the spherical catadioptric setup, the GoPano iPhone attachment and the D300 with a fisheye lens. Notice the wide range of distortions as well as the low quality in the case of the GoPano.

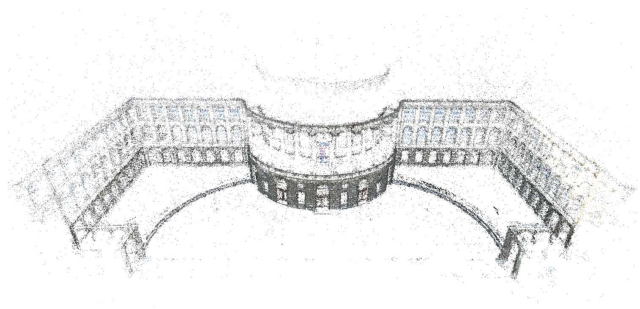


Figure 10: The SfM point cloud used for the experimental evaluation.

The approach is validated experimentally and using real data and its accuracy and robustness is assessed by comparing the obtained calibration mappings against the calibration from a state-of-the-art toolbox [13]. We make our source code available at [1].

Acknowledgements

The research leading to these results has received funding from Google's Project Tango. We would like to thank Dr. Martin Oswald for his valuable input regarding the convex optimization.

References

- [1] Project page. <http://www.cvg.ethz.ch/research/radially-symmetric-cameras/>.
- [2] A. Agrawal, Y. Taguchi, and S. Ramalingam. Analytical forward projection for axial non-central dioptric and catadioptric cameras. In *ECCV*, 2010.
- [3] M. Bujnak, Z. Kukelova, and T. Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *ACCV*, 2010.
- [4] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine vision and applications*, 13(1):14–24, 2001.
- [5] M. A. Fischler and R. C. Bolles. 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.
- [6] C. Geyer and K. Daniilidis. Paracatadioptric camera calibration. *PAMI*, 24(5):687–695, 2002.
- [7] A. Goshtasby. Correction of image deformation from lens distortion using bezier patches. *Computer Vision, Graphics, and Image Processing*, 47(3):385–394, 1989.
- [8] R. Hartley and S. B. Kang. Parameter-free radial distortion correction with center of distortion estimation. *PAMI*, 29(8):1309–1321, 2007.
- [9] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. In *CVPR*, 2009.
- [10] Z. Kukelova, M. Bujnak, and T. Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *ICCV*, 2013.
- [11] H. Li and R. Hartley. Plane-based calibration and auto-calibration of a fish-eye camera. In *ACCV*, 2006.
- [12] T. Sattler, B. Leibe, and L. Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, 2012.
- [13] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IROS*, 2006.
- [14] J.-P. Tardif, P. Sturm, and S. Roy. Self-calibration of a general radially symmetric distortion model. In *ECCV*, 2006.
- [15] S. Thirithala and M. Pollefeys. Multi-view geometry of 1d radial cameras and its application to omnidirectional camera calibration. In *CVPR*, 2005.
- [16] S. Thirithala and M. Pollefeys. The radial trifocal tensor: A tool for calibrating the radial distortion of wide-angle cameras. In *CVPR*, 2005.
- [17] S. Thirithala and M. Pollefeys. Radial multi-focal tensors. *International Journal of Computer Vision*, 96(2):195–211, 2012.