Camera Pose Estimation using Implicit Distortion Models

Linfei Pan
ETH Zurich
linpan@student.ethz.ch

Marc Pollefeys
ETH Zurich, Microsoft
marc.pollefeys@inf.ethz.ch

Viktor Larsson
Lund University
viktor.larsson@math.lth.se

Abstract

Low-dimensional parametric models are the de-facto standard in computer vision for intrinsic camera calibration. These models explicitly describe the mapping between incoming viewing rays and image pixels. In this paper, we explore an alternative approach which implicitly models the lens distortion. The main idea is to replace the parametric model with a regularization term that ensures the latent distortion map varies smoothly throughout the image. The proposed model is effectively parameter-free and allows us to optimize the 6 degree-of-freedom camera pose without explicitly knowing the intrinsic calibration. We show that the method is applicable to a wide selection of cameras with varying distortion and in multiple applications, such as visual localization and structure-from-motion.

1. Introduction

The intrinsic calibration of a camera describes the mapping between 2D pixels in the image and the corresponding rays in 3D. Knowing the intrinsic calibration, i.e. being able to project into the image (or vice versa), is a prerequisite for most geometric vision tasks. This mapping is usually parameterized using a low-dimensional parametric model.

In this paper we propose to instead implicitly model the intrinsic calibration. More specifically we look at estimating the 6 degree-of-freedom camera pose from given 2D-3D correspondences when the intrinsic calibration is unknown. Our approach assumes that the camera is central and radially-symmetric (i.e. the distortion only varies with the radial offset and not the angle) which is the case for most consumer cameras. The main idea is to replace the explicit parametric model with a regularization term that force the underlying distortion map to be smooth. The proposed implicit distortion model allows us to essentially parameterize the intrinsic calibration in terms of the camera's extrinsic parameters. It is effectively parameter-free and generalizes to a wide selection of camera and lens combinations from well-behaved pinhole images to highly non-linear optical systems such as fisheye or catadioptric cameras. The method can be further extended to leverage multiple images from the same camera and even be incorporated into a full bundle-adjustment, jointly refining 3D points and cameras.

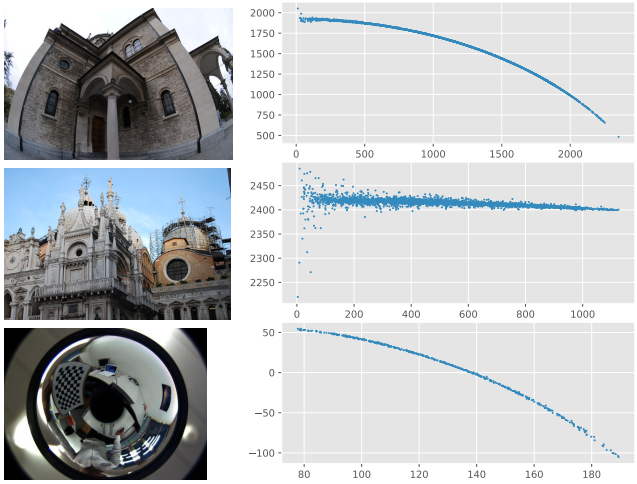

Figure 1. Example of the point-wise focal lengths f_i computed from (8) versus the image radii r_i. *Top:* Fisheye lens. *Middle:* 45mm lens. *Bottom:* Catadioptric camera. Negative values for the focal lengths correspond to points behind the camera center.

2. Background and Related Work

Modeling Cameras and Lens Distortion. The standard pinhole camera model works well for rectilinear lenses. To handle deviations from this, it is common practice to include a non-linear distortion function applied to the pinhole projections. This can be formalized as

$$\boldsymbol{x} = f\,\mathcal{D}(\pi\,(R\boldsymbol{X} + \boldsymbol{t})) + \boldsymbol{c} \tag{1}$$

where f is the focal length, π the pinhole projection (dehomogenization), $\boldsymbol{c}$ the principal point and $\mathcal{D}$ the non-linear function modeling the lens distortion. If the camera is radially symmetric, the distortion mapping only depends on the radial offset and thus has the following form $\mathcal{D}(\boldsymbol{z}) = d(\|\boldsymbol{z}\|)\boldsymbol{z}$, where $d : \mathbb{R}_+ \to \mathbb{R}$. The function d is often parameterized as a polynomial in the radius r,

$$d(r) = 1 + k_1 r^2 + k_2 r^4 + \dots \tag{2}$$

where $k_1, k_2, \ldots$ are the parameters which need to be calibrated per-camera. The polynomial in (2) is the Brown-Conrady model [4, 7] and is a popular choice in practice.[1]

While (1) works well for many cameras, applying the distortion model on top of the pinhole projections introduces problems for very wide field-of-view cameras (e.g. fisheye or catadioptric systems). In [38], the authors propose to instead reformulate the projection equations as

$$\lambda \begin{bmatrix} \boldsymbol{x} \\ F(\|\boldsymbol{x}\|) \end{bmatrix} = R\boldsymbol{X} + \boldsymbol{t} \tag{3}$$

where $\boldsymbol{x}$ is the centered image point and $F : \mathbb{R}_+ \to \mathbb{R}$ is the *distortion function*. In [38], the function F is also parameterized as a polynomials as in (2), except that the constant coefficient is not fixed to one, as it also encodes the focal length in this case. This approach can handle any radially symmetric camera (including $> 180^\circ$ FoV). Similar distortion (or rather undistortion, since it applies to the image observations) models have also been used by Fitzgibbon [8] and many others, see e.g. [5, 13, 15–20, 29, 30].

Pose Estimation of Radially Symmetric Cameras. For a radially-symmetric camera, it is possible to partially recover the camera pose by using that the distortion map is purely radial. If $\boldsymbol{x} = (x, y)$, we from (3) get a linear constraint as

$$(-y, x, 0) \cdot (R\boldsymbol{X} + \boldsymbol{t}) = 0. \tag{4}$$

This is the *radial alignment constraint* [44] which requires the projection to be somewhere on the radial-line passing through the image point. Each correspondence thus gives a linear constraint on $(R, \boldsymbol{t})$ which is independent of the distortion map F. Note that the constraint does not involve t_3 and it is therefore only possible to recover the camera pose up to an unknown forward translation. In [38, 44], this constraint was used to estimate the partial camera pose with respect to a planar calibration pattern. This was later generalized to non-planar configurations by Kukelova et al. [15] which solved the general minimal estimation problem (from 5 point correspondences). The multi-view geometry of cameras if you only consider the radial constraints was originally considered by Thirthala and Pollefeys [41], and later extended to full structure-from-motion [12, 14, 22].

Given the partial extrinsics (R, t_1, t_2), we can compute a partial repojection error; only measuring the deviation from the radial line. The *radial reprojection error* is defined as

$$\varepsilon_r(R, \boldsymbol{t}, \boldsymbol{x}, \boldsymbol{X}) = \|(I - \frac{\boldsymbol{z}\boldsymbol{z}^\top}{\boldsymbol{z}^\top \boldsymbol{z}})\boldsymbol{x}\|^2 \tag{5}$$
$$\text{where } \boldsymbol{z} = R_{12}\boldsymbol{X} + \boldsymbol{t}_{12}$$

where $(R_{12}, \boldsymbol{t}_{12})$ denotes the first two rows of $(R, \boldsymbol{t})$. This error can be used both to evaluate the quality of pose hypotheses in RANSAC, as well as for non-linear pose refinement. If we require $\lambda > 0$ in equation (3), we get the following constraint $\boldsymbol{x}^\top(R_{12}\boldsymbol{X} + \boldsymbol{t}_{12}) > 0$ which is analogous to the standard cheirality check and can be used for filtering.

Stratified Calibration Approaches. The radial alignment constraint has been used extensively in stratified calibration methods which first estimate the partial extrinsics (R, t_1, t_2), followed by joint estimation of t_3 and the intrinsic calibration. This approach was originally used by Tsai [44] (and later in [38]) for plane-based calibration. Similarly, Kukelova et al. [15] used it for joint pose estimation and self-calibration using the division model [8]. Later, Larsson et al. [21] extended the approach to a more general set of distortion models. Camposeco et al. [6] proposed a stratified approach for non-parametric calibration.

Non-Parametric Camera Models. There also exists non-parametric (or generic) models for intrinsic calibration which estimate independent rays for each pixels, allowing them to model arbitrary camera systems. This type of model was first proposed by Grossberg and Nayar [9]. Since then, there have been multiple works improving in various aspects; initialization [31], distortion center estimation [10], interpolating local B-splines [2, 32] or RBF [27] and ease-of-use [39]. Due to the high number of parameters in these models, more dense calibration patterns [39] or active-target (e.g. monitors) [3] are typically used.

The generic models still explicitly describe the intrinsic calibration. In contrast to these methods, we parameterize the intrinsic calibration in terms of the camera pose and then regularize it. Thus we only optimize the intrinsic calibration indirectly via the extrinsic parameters.

Related Work on Implicit Distortion Modeling. The work most closely related to ours is from Camposeco et al. [6] where the authors use a stratified approach for non-parametric self-calibration. They first estimate the rotation and two translation parameters using the radial alignment constraint (as described above). To estimate the forward translation t_3, similar to our approach, they avoid explicitly parameterizing the distortion map. In their method, they instead use that the mapping from image radii to opening angles (the angle to the principal axis) is non-decreasing. If (R, t_1, t_2) is fixed, each pair of correspondences then restrict t_3 to a half-interval (either $[a, \infty]$ or $[-\infty, b]$). The forward translation is then recovered by finding the position which satisfies the most intervals. This is formulated as the following convex optimization problem

$$\min_{t_3} \sum_i \max(a_i - t_3, 0) + \sum_j \max(t_3 - b_j, 0) \tag{6}$$

Once the camera pose is estimated they recover an explicit non-parametric intrinsic calibration.

The paper from Camposeco et al. [6] builds on a previ-

[1] The model also includes tangential terms which are often neglected.

ous work from Hartley and Kang [10] which takes a similar approach but for the planar case. For a planar scene they use the radial constraints to estimate homographies which map the planar pattern onto the radial line for each correspondence. This only yields the first two rows of the homography H_{12}, and in [10] they propose to optimize over the third row $\boldsymbol{h}_3$ for each homography by regularizing the radial offset of the mapped points, i.e. $r_i^u = \|H_{12}\boldsymbol{x_i}\|/\boldsymbol{h}_3^\top \boldsymbol{x_i}$.

Our approach is similar to [6] in the sense that we also solve for the camera pose without parameterizing the distortion. In contrast to [6], we use a stronger regularization (a generalization of the one proposed in [10], see Sec. 3.1) compared to their ordering constraint. Compared to [10], we consider the general non-planar case and parameterize the full camera matrix. For the initial upgrade step we only need to optimize t_3 (similar to [6]), instead of the three elements of the third row $\boldsymbol{h}_3$ as in [10].

Furthermore, by posing the problem in terms of the pointwise focal lengths f_i as in equation (8), the expressions are simplified and allow us to more easily perform optimization over the full 6 degree-of-freedom camera pose. In experiments we show that this allows for accurate pose estimates while still generalizing to a wide variety of cameras. Additionally, in both [6] and [10], they do not consider the full bundle adjustment problem where the 3D structure is optimized jointly with the camera poses.

In the following sections we present our approach for camera pose estimation. In Section 3 we first detail how we implicitly model the intrinsic calibration. Section 4 shows how to perform robust estimation using the proposed model and in Section 5 we extend it to full bundle-adjustment.

3. Implicit Distortion Modeling

Given a 2D-3D point correspondence $(\boldsymbol{x}_i, \boldsymbol{X}_i)$ we can define the *point-wise focal length* f_i as

$$\lambda \begin{bmatrix} \boldsymbol{x}_i \\ f_i \end{bmatrix} = R\boldsymbol{X}_i + \boldsymbol{t} \tag{7}$$

In the formulation from [38], each f_i is simply the pointwise evaluation of the distortion map, i.e. $f_i = F(\|\boldsymbol{x}_i\|)$. For example, pinhole cameras have $f_i = f$ for all i. From (7), we can then solve for the point-wise focal length f_i as

$$f_i = \frac{\|\boldsymbol{x}_i\|^2 \left(R_3\boldsymbol{X}_i + t_3\right)}{\boldsymbol{x}_i^\top \left(R_{12}\boldsymbol{X}_i + \boldsymbol{t}_{12}\right)} \tag{8}$$

Figure 1 show some example of the point-wise focal lengths computed for some cameras. In [38], the authors explicitly model f_i using a polynomial model similar to (2).

In this work we instead implicitly model the distortion by regularizing the mapping between radius in the image $\|\boldsymbol{x}_i\|$ and the corresponding point-wise focal length f_i. Equation (8) parameterizes the f_i in terms of the camera pose and using this we setup an optimization problem as

$$\min_{R,\boldsymbol{t}} \sum_{i=1}^{N} \varrho\left(\varepsilon_r(R, \boldsymbol{t}, \boldsymbol{x}_i, \boldsymbol{X}_i)\right) + \mathcal{R}\left(\{f_i\}_{i=1}^N\right) \tag{9}$$

where $\mathcal{R}$ is a regularizer of the pointwise focal lengths, ε_r is the radial reprojection error defined in Eq. (5) and ϱ is a robust loss function. Note that the optimization is over the 6 DoF camera pose, and only the regularizer $\mathcal{R}$ constrains the forward translation t_3 in the optimization problem.

The motivation for this formulation comes from considering the orthogonal decomposition of the true reprojection error into the radial and tangential components, i.e.

$$\|\boldsymbol{x} - \boldsymbol{z}\|^2 = \underbrace{\|(I - \frac{\boldsymbol{z}\boldsymbol{z}^\top}{\boldsymbol{z}^\top\boldsymbol{z}})\boldsymbol{x}\|^2}_{\varepsilon_r} + \underbrace{\|\frac{\boldsymbol{z}\boldsymbol{z}^\top}{\boldsymbol{z}^\top\boldsymbol{z}}(\boldsymbol{x} - \boldsymbol{z})\|^2}_{\varepsilon_t} \tag{10}$$

where $\boldsymbol{z}$ is the true projection (using the unknown intrinsic calibration). Here only the tangential component ε_t depends on the radial offset $\|\boldsymbol{z}\|$ (as $\boldsymbol{z}\boldsymbol{z}^\top/\boldsymbol{z}^\top\boldsymbol{z}$ is scale-invariant). In the cost (9), ε_t is replaced by the regularizer.

3.1. Regularization of the Pointwise Focal Lengths

Each 2D-3D correspondence $(\boldsymbol{x}_i, \boldsymbol{X}_i)$ yields one observation of the unknown distortion mapping,

$$F : r_i \mapsto f_i \tag{11}$$

where $r_i = \|\boldsymbol{x}_i\|$. Similar to [6, 10] we sort the correspondences such that $r_i < r_{i+1}$. For the regularization we consider a generalization of local linearity assumption from [10], but instead of only considering the two neighbouring points, we do a least square fitting to the k-nearest neighbours and penalize the deviation from the line. For each i define the vector $\boldsymbol{r}_i = \begin{bmatrix}1 & r_i\end{bmatrix}^\top$ and form A^i and $\boldsymbol{f}$ from the $k = 2m$ symmetric nearest neighbours as

$$A^i = \begin{bmatrix}\boldsymbol{r}_{i-m} & \cdots & \boldsymbol{r}_i & \cdots & \boldsymbol{r}_{i+m}\end{bmatrix}^\top \tag{12}$$

$$\boldsymbol{f}^i = \begin{bmatrix}f_{i-m} & \cdots & f_i & \cdots & f_{i+m}\end{bmatrix}^\top \tag{13}$$

If $\{(r_i, f_i)\}$ lie on a line, there exist parameters $\boldsymbol{\beta}^i$ such that $A_i\boldsymbol{\beta}^i = \boldsymbol{f}^i$ exactly. The best fitting line $\boldsymbol{\beta}^i \in \mathbb{R}^2$ can be found by solving a linear least square problem,

$$\boldsymbol{\beta}^i = ({A^i}^\top A^i)^{-1}{A^i}^\top \boldsymbol{f^i} \tag{14}$$

Evaluating the line at r_i we get the estimated focal length as

$$\tilde{f}_i = \boldsymbol{r_i}^\top \boldsymbol{\beta}^i \tag{15}$$

We now propose to minimize the difference $f_i - \tilde{f}_i$ as the regularization. Note that the difference is a linear combination of the pointwise focal lengths in the neighbourhood,

$$\begin{aligned}\tilde{f}_i - f_i &= \boldsymbol{r_i}^\top\boldsymbol{\beta}^i - f_i \\ &= \boldsymbol{r_i}^\top({A^i}^\top A^i)^{-1}{A^i}^\top \boldsymbol{f^i} - f_i = \boldsymbol{a}_i^\top \boldsymbol{f}\end{aligned} \tag{16}$$

where $\boldsymbol{a}_i \in \mathbb{R}^N$ is a constant vector (depending only on the image radii) and $\boldsymbol{f} \in \mathbb{R}^N$ is the vector of all focal lengths.

The regularization function $\mathcal{R}$ is then computed by summing over the robust difference between the locally estimated focal length $\tilde{f}_i$ and the actual f_i

$$\mathcal{R}(\boldsymbol{f}) = \sum_i \varrho_\epsilon \left(|\tilde{f}_i - f_i| \right) = \sum_i \varrho_\epsilon \left(|\boldsymbol{a}_i^\top \boldsymbol{f}| \right) \tag{17}$$

where each $\boldsymbol{a}_i$ encode the coefficients for each neighborhood (see (16)), and ϱ_ϵ is a robust loss function. In the experiments we use the Huber loss [11] with parameter $\epsilon = 1$ px which seems to work well for the cameras we evaluate on. In Sec. 6.3 we evaluate the different regularization strategies as well as the impact of the robust loss.

4. Camera Pose Estimation and Calibration

In this section we present our pipeline for robust camera pose estimation using the implicit distortion model presented in Section 3. Similar to [6, 15, 21, 38, 44] we first estimate partial extrinsic parameters using the radial alignment constraint, followed by an upgrade step and local refinement. Our pipeline consists of the following steps:

1. Initialization using the radial constraints.
2. Estimation of the forward translation t_3.
3. Filtering spurious inliers.
4. Full refinement of the 6 DoF camera pose.

In the following paragraphs we detail each of these steps.

Partial Initialization of Camera Pose. Using the 5 point minimal solver from Kukelova [15] we estimate the partial camera pose (consisting of R, t_1 and t_2) in an LO-RANSAC framework [23]. The radial reprojection error (5) is used in MSAC [42] scoring and minimized during the local optimization. The principal point is chosen as the image center. Optionally we can optimize the principal point at this stage.

Estimating the Forward Translation. The previous step only recovers the orientation and two of the translation parameters. To recover t_3 we keep the other pose parameters fixed and minimize the regularization function $\mathcal{R}$ with respect to t_3, i.e.

$$\min_{t_3} \mathcal{R}\left(\boldsymbol{f}(t_3)\right). \tag{18}$$

Note that by fixing $R_{12}, \boldsymbol{t}_{12}$ in (8), each f_i can be written as

$$f_i = \alpha_i + t_3 \beta_i \tag{19}$$

where $\alpha_i = \frac{\|\boldsymbol{x}_i\|^2 R_3 \boldsymbol{X}_i}{\boldsymbol{x}_i^\top (R_{12}\boldsymbol{X}_i + \boldsymbol{t}_{12})}$ and $\beta_i = \frac{\|\boldsymbol{x}_i\|^2}{\boldsymbol{x}_i^\top (R_{12}\boldsymbol{X}_i + \boldsymbol{t}_{12})}$, are constants. If the robust loss function ϱ_ϵ is chosen to be convex, e.g. as the Huber loss, then the optimization problem in (18) is also convex. Thus we can recover globally optimal t_3 by applying any local optimization scheme. In the experiments we used Levenberg-Marquardt [24, 26]. If we instead would use the squared loss ($\varrho(r) = r^2$) this leads to a linear least squares problem which can be solved non-iteratively.

Filtering Spurious Inliers. The correspondences used for the previous estimation are only filtered based on the radial reprojection error. This allows for points which by chance lie close to the corresponding radial line to survive the filtering process. While these spurious inliers have a small effect on the radial estimate, they are actually outliers and have a larger impact for the estimation of the intrinsic calibration and forward translation. To identify these outliers, we use a *sliding median*-based filter similarly to [6, 10]. For each point, we calculate the difference between the pointwise focal length f_i given by (8) and the median $\bar{f}_i$ of its neighbors $\bar{f}_i$, and filter points where $|f_i - \bar{f}_i|$ is larger than k times the median error $\text{MED}(|f_i - \bar{f}_i|)$.

Refinement of 6 DoF Camera Pose. The initial camera pose recovered from the previous steps is now refined by minimizing the cost in (9) over the full 6 DoF camera pose. Note that differentiating the pointwise focal lengths f_i (8) with respect to the camera pose is not more expensive than computing Jacobians for the standard pinhole projection (as they have a similar structure). For the optimization we use Levenberg-Marquardt [24, 26].

4.1. Joint Estimation of Multiple Images

So far we have presented a method for robust pose estimation and implicit calibration of a single image. In settings where one has multiple images taken by the same camera (i.e. they have the same distortion mapping), it is possible to jointly estimate and calibrate them. To do this we again use the radial alignment constraint to estimate partial intrinsics independently for each camera. For the initializing of the forward translations, we can leverage correspondences from multiple images in the regularization function (18). The same approach was used in Camposeco et al. [6] for their multi-camera estimation.

Let $t_3^1, t_3^2, \ldots, t_3^M$ denote the forward translations for the M images. Collecting all correspondences and again sorting them by the image radii, the pointwise focal lengths are

$$f_i = \alpha_i + t_3^{k(i)} \beta_i \tag{20}$$

where $k(i) \in \{1, 2, \ldots, M\}$ is the image index from which the correspondence came from. This does not introduce any extra non-linearities into the cost and the optimization problem is still convex. Similarly, the final non-linear refinement can be extended to multiple images using the same idea.

4.2. Non-Parametric Intrinsic Calibration

To allow for explicit distortion and undistortion we can recover a non-parametric intrinsic calibration. Given a fixed

camera pose, each correspondence gives one observation of the distortion mapping, as in (11). Note however that here the focal lengths are chosen to perfectly fit the measurements (i.e. the radial offset exactly matches the 2D point), which results in the noisy f_i in Figure 1. In [6, 10] the authors use a *sliding median* filter to reduce the impact of this noise. We propose to instead denoise the f_i with the regularizer $\mathcal{R}$ by solving the following optimization problem

$$\min_{\boldsymbol{f}} \sum_i \varrho\left(|f_i - \hat{f}_i|\right) + \lambda \mathcal{R}(\boldsymbol{f}) \tag{21}$$

where $\hat{f}_i$ are the noisy pointwise focal lengths computed from (8). Since we want to recover corrected f_i which agree with the regularizer, we do not use a robust loss in $\mathcal{R}$ here. The parameter λ controls the trade-off between fitting the data and adhering to the regularization. For the correct focal lengths, the residual errors should be equally distributed in the tangential and radial components, assuming the image noise is isotropic. This is illustrated in Figure 2. In Algorithm 1 we propose a simple scheme for selecting the trade-off parameter λ based on this idea. Note that the radial reprojection for each point is invariant to the choice of point-wise focal length f_i.

This process yields a collection of pairs of image radii and corrected pointwise focal lengths, (r_i, f_i). To undistort a point in the image, we can simply find the pairs with most similar radii and interpolate to get the focal length. Similarly, to project a 3D point, we instead compute the opening angles $\theta_i = \text{atan2}(r_i, f_i)$ for each point and interpolate to get the image radius r of the projected point. Sorting the lists we can quickly find the pairs for the interpolation.

Algorithm 1: Automatic selection of λ

$\lambda \leftarrow \lambda_{init}$, $best\_res \leftarrow \infty$
$\epsilon_{rad} \leftarrow$ radial reprojection error
for $i \leftarrow 0$ **to** *max-iters* **do**
  Solve (21) to recover distortion mapping
  $\epsilon_{tan} \leftarrow$ tangential reprojection error
  $res \leftarrow |\epsilon_{rad} - \epsilon_{rms}|$
  if $\epsilon_{tan} \leq \epsilon_{rad}$ **then**
    $\lambda \leftarrow 10\lambda$ `// under-regularized`
  else
    $\lambda \leftarrow \lambda/2$ `// over-regularized`
  end
  if $res < best\_res$ **then**
    $best\_res \leftarrow res$
    Save best calibration found so far
  end
end

5. Bundle Adjustment with Implicit Distortion

The proposed cost function that we minimize (9) naturally extends to multiple images and optimization of the 3D points as well, allowing for full bundle-adjustment. However, as our regularization is built on the correlation between neighboring points, it breaks the independence of the 3D points which is required to perform the Schur complement trick [43]. This greatly limits the size of the problems which can be tackled in practice.

To solve this problem we propose an iterative scheme. In each iteration we solve a surrogate problem where the Schur complement trick applies. For each residual in $\mathcal{R}$, we replace all 3D points except for one with copies of the previous iterations value, i.e. $\boldsymbol{f}^i$ in (13) is replaced by

$$\tilde{\boldsymbol{f}}^i = \left[f_{i-m}(\boldsymbol{X}_{i-m}^{t-1}) \ \cdots \ f_i(\boldsymbol{X}_i) \ \cdots \ f_{i+m}(\boldsymbol{X}_{i+m}^{t-1})\right]^\top \tag{22}$$

where $\boldsymbol{X}_k^{t-1}$ denotes the 3D point from the previous iteration. Note that each f_i still depends on the pose parameters.

The inner optimization problem is again solved using Levenberg-Marquardt [24, 26] in the experiments. Since the inner iterations are only an approximation of the original cost, it is not necessary to run the inner optimization problem until convergence each time.

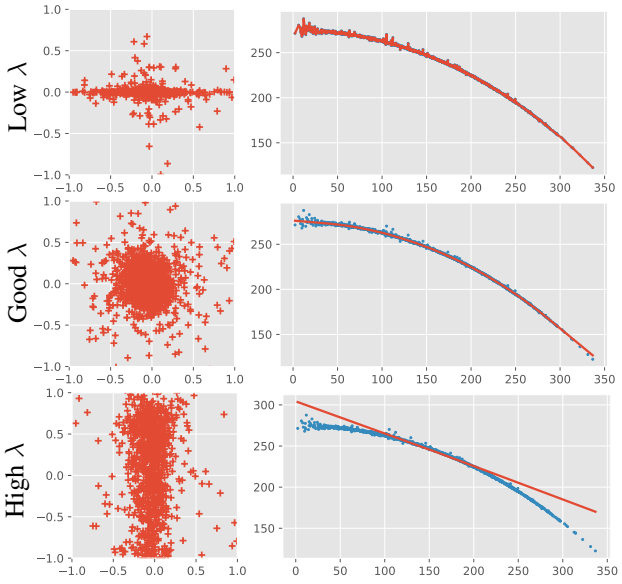

Figure 2. Impact of regularization parameter λ in (21). *Left:* The distribution of the reprojection errors in polar coordinates (radial/tangential). *Right:* The estimated f_i as a function of r_i after solving (21). *Middle:* For an appropriately chosen λ the radial and tangential errors are balanced. *Top/Bottom:* Too low or high values of λ lead to under/over-regularization of the distortion mapping.

6. Experimental Evaluation

In the experimental evaluation we show that the implicit model can handle a wide variety of cameras. We evaluate both in controlled settings (such as checkerboard calibration) and on in-the-wild self-calibration scenarios (visual localization / Structure-from-Motion). Unless otherwise stated, all experiments use the same regularization function; local linear fitting to symmetric neighbors (taking 2 on each side). In the experiments we mainly compare with Camposeco et al. [6] which also does model-free pose estima-

tion. In [6] they derive their constraints from pairs of correspondences. To limit the computation cost they propose to take $N = 120$ pairs for their regularization. For a fair comparison we take the same number of pairs as there are image correspondences, making it comparable to the number of terms in the regularizer $\mathcal{R}$.

In the supplementary material, we show additional results and detail parameter settings from the experiments.

6.1. Evaluation of Implicit Calibration

Checkerboard calibration. We start by comparing our approach on classical checkerboard calibration data. For the evaluation we consider an aggregate of calibration datasets (collected by the authors of [25]). The dataset contains images from 41 different cameras, with field-of-views spanning from 88° to 268° degrees. The images for each camera is split into (≈ 35) training images and (≈ 15) test images. Please see the supplementary for more details. We use the calibration toolbox from BabelCalib [25] to fit a parametric model used as a psuedo-ground truth for the experiment. Note that some of the datasets have multiple checkerboard patterns and thus the scene is not always planar.

First we evaluate the ability to recover accurate camera poses. For the training set of each camera we jointly estimate the poses with the implicit distortion model and compare with the camera poses from [25]. We also compare with the non-parametric approach from Camposeco et al. [6] which also performs model-free pose estimation. Table 1 shows the rotation and translation errors. Since absolute scale is not available for all datasets, we report the position error relative to the calibration board diagonal. The principal point is initialized to the image center and optimized in the radial estimation. The `OV plane` dataset contains sequences where the calibration board is close to fronto-parallel. This is a degenerate configuration for focal length calibration which leads to larger errors.

Next we evaluate the accuracy of the intrinsic calibration recovered by the method. For this we fit the explicit non-parametric model on the training set (as described in Section 4.2). Using the calibration we estimate poses in the test set; undistorting the keypoints and running P3P+RANSAC, followed by non-linear refinement of reprojection error. Table 2 show the errors and number of images which had less than 1px RMS reprojection error. For comparison we show both the error obtained with the explicit parametric model from [25] as well as the reprojection error obtained from the non-parametric calibration obtained with [6].

Structure-from-Motion. Next we evaluate our approach in the context of Structure-from-Motion. We consider four datasets captured with two different cameras (one with low-distortion and one with a fisheye lens). Example images from the cameras can be seen in the first two rows of Figure 1. For each dataset we first create a reference reconstruction using COLMAP together with the ground truth intrinsic calibration (obtained via offline calibration). The model is then manually rescaled to be approximately metric. Each image is then re-matched to the reconstruction to obtain 2D-3D matches to the model (potentially containing outlier matches). For each image we re-estimate the camera pose and compare to the SfM poses. Table 3 shows the statistics for the pose estimation error using both single image optimization and multiple image optimization. Figure 3 shows the cumulative error histograms. The proposed method which optimizes the full 6 DoF camera pose consistently out-performs the method from Camposeco et al. [6].

	Proposed			Camposeco et al. [6]		
	ϵ_{rot}	ϵ_{pos}	$< 1°, 1\%$	ϵ_{rot}	ϵ_{pos}	$< 1°, 1\%$
`OV corner`	**1.07**	**0.58**	**122 / 280**	1.20	0.59	81 / 280
`OV cube`	**0.07**	**0.03**	**105 / 105**	0.04	0.11	**105 / 105**
`OV plane`	1.23	6.78	**35 / 92**	**1.06**	**1.78**	32 / 92
`Kalibr`	**0.17**	**0.18**	**277 / 280**	0.31	0.86	231 / 280
`OCamCalib`	0.62	**0.26**	**61 / 79**	**0.58**	0.59	55 / 79
`UZH DAVIS`	**0.74**	**1.91**	**110 / 140**	2.14	8.28	62 / 140
`UZH Snapdragon`	**0.16**	**0.25**	**137 / 140**	0.43	0.89	122 / 140

Table 1. Pose estimation on the training set for the calibration datasets. The table shows the average rotation error (degrees) and position error (percentage of calibration pattern size) compared to the poses obtained from BabelCalib [25], and the number of images which obtain less than $1°$ rotation and 1% positional error.

	[25]		Proposed		Camposeco et al. [6]	
	ϵ^{BC}_{rms}	ϵ_{pp}	ϵ_{rms}	< 1px	ϵ_{rms}	< 1px
`OV corner`	1.52	16.28	**2.09**	**16/120**	2.96	0/120
`OV cube`	0.29	0.40	**0.31**	**49/49**	0.40	49/49
`OV plane`	0.60	0.89	**0.82**	**33/41**	2.84	9/41
`Kalibr`	0.21	0.88	**0.30**	**118/120**	0.61	113/120
`OCamCalib`	0.68	2.17	**0.97**	**31/40**	2.62	17/40
`UZH DAVIS`	0.41	0.37	**0.42**	**58/60**	0.72	49/60
`UZH Snapdragon`	0.26	0.56	**0.28**	**60/60**	0.46	59/60

Table 2. Evaluation of non-parametric intrinsic calibration on the test set. The table shows the RMS reprojection error (in pixels) on the test set using the non-parametric intrinsic calibration obtained from the training set. ϵ_{pp} is the average error in the estimated principal point (in pixels). We also show the reprojection error obtained with the parametric model estimated using BabelCalib [25]

Undistorting images. Figure 4 shows qualitative results for the intrinsic calibration. The undistorted image was not part of the set used for estimation, showing that the method does not overfit to the calibration set.

6.2. Implicit Self-Calibration in Visual Localization

To evaluate the robustness of the method we consider two challenging visual localization datasets; Aachen Day-Night [37] and InLoc [40]. We use the hloc [33–35] framework (SuperPoint+SuperGlue matching with NetVLAD [1] image retrieval) to create 2D-3D correspondences. For each set of correspondences we estimate the camera pose, both independently and jointly (as in Section 4.1). We again

		Kirchenge [22] (369)		Grossmunster [22] (373)		Kazan [28] (282)		Doge Palace [28] (241)	
		ϵ_{rot} (deg.)	ϵ_{pos} (cm.)	ϵ_{rot} (deg.)	ϵ_{pos} (cm.)	ϵ_{rot} (deg.)	ϵ_{pos} (cm.)	ϵ_{rot} (deg.)	ϵ_{pos} (cm.)
Single image	Proposed	**0.023**	**0.6**	**0.038**	**0.9**	**0.386**	**8.7**	**0.338**	**11.7**
	Camposeco et al. [6]	0.027	0.8	0.044	3.8	0.403	19.8	0.364	121.2
Multiple images	Proposed	**0.020**	**0.4**	**0.036**	**0.5**	**0.379**	**3.4**	**0.328**	3.4
	Camposeco et al. [6]	0.035	0.9	0.056	1.3	0.403	6.5	0.364	**1.1**

Table 3. Average rotation error (in degree) and camera position error (in centimeters) with COLMAP reconstruction result as pseudo groundtruth. The number of images for each dataset is shown in the bracket. Single image optimization and multiple images optimization are presented separately and both compared with method proposed in [6].

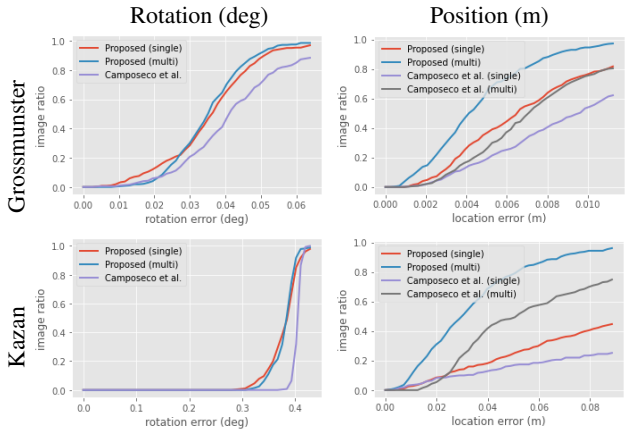

Figure 3. Pose estimation in Structure-from-Motion. Cumulative errors for the proposed method and [6], for both single and multi-image optimization. *Left:* Cumulative rotation error in degree. *Right:* cumulative camera location error in meters. Since [6] does not optimize the rotation after the initial estimate, only one line is presented here. *Top:* Grossmunster [22]. *Bottom:* Kazan [28]

compare with Camposeco et al. [6] and report the results in Table 4. For our approach we show the results with and without the filtering from Sec. 3. The filtering has significant impact on the Aachen Day-Night while it does not improve on InLoc. We can also see that going from single to multi-image optimization improves the results significantly.

For comparison we also report the errors obtained with using the ground truth intrinsic parameters, as well as jointly estimating pose and a one-parameter parametric model using the solver from [21] in LO-RANSAC [23, 36].

6.3. Comparison of Regularization Functions

In this section we perform an ablation study for the regularization function and the method for recovering the intrinsic calibration. For the experiment we consider the checkerboard dataset used in Section 6.1 and consider the average errors over the test set. Table 5 shows the full results. We compare the following: The regularization function *(Reg.)* Here we compare penalizing the variation as also proposed in [10], i.e. $|f_{i+1} - f_i|$ with locally fitting linear and quadratic functions (as in Section 3.1). The number of neighbouring points *(kNN)* used in the local fitting. The *robust loss* ϱ used. The method for recovering the intrinsic calibration *(Calib.)*. Here we compare the raw estimates given by equation (8), with Algorithm 1 and median filtering as used in [6, 10]. In the table it can be seen that the choice of regularization function is not too critical as the differences between them are quite small.

Aachen Day-Night [37]		day	night
Single image	Proposed (w/ filter)	**58.3 / 76.5 / 94.2**	**61.2 / 77.6 / 99.0**
	Proposed (w/o filter)	51.3 / 67.4 / 92.8	50.0 / 68.4 / 94.9
	Camposeco et al. [6]	46.0 / 61.9 / 83.1	45.9 / 69.4 / 85.7
Multiple images	Proposed (w/ filter)	**82.6 / 92.4 / 98.3**	**73.5 / 88.8 / 100.0**
	Proposed (w/o filter)	77.8 / 90.8 / **98.3**	**73.5 / 88.8 / 100.0**
	Camposeco et al. [6]	18.6 / 34.3 / 83.5	37.8 / 63.3 / 99.0
Parametric model	hloc [33] + GT calib.	**89.6 / 95.4 / 98.8**	**86.7 / 93.9 / 100.0**
	hloc [33] + [21]	60.6 / 82.8 / 98.2	64.3 / 82.7 / 100.0
InLoc [40]		duc1	duc2
Single image	Proposed (w/ filter)	28.3 / 46.0 / 63.6	**26.7 / 48.1 / 61.8**
	Proposed (w/o filter)	**29.8 / 46.5 / 64.6**	26.0 / 42.7 / 59.5
	Camposeco et al. [6]	23.2 / 40.4 / 55.1	18.3 / 31.3 / 42.7
Multiple images	Proposed (w/ filter)	34.8 / 52.5 / **69.7**	**38.9** / 57.3 / **74.0**
	Proposed (w/o filter)	**35.4 / 53.0 / 69.7**	35.9 / **58.0 / 74.0**
	Camposeco et al. [6]	34.8 / 51.0 / 69.2	35.1 / **58.0 / 74.0**
Parametric model	hloc [33] + GT calib.	**46.5 / 66.2 / 78.3**	**51.9 / 74.8 / 78.6**
	hloc [33] + [21]	25.8 / 47.5 / 62.6	27.5 / 55.0 / 66.4

Table 4. Visual localization on the datasets *Aachen Day-Night* [37] and *InLoc* [40]. The table shows the percentage of images within the thresholds (0.25m, 2°) / (0.5m, 5°) / (5m, 10°) and (0.25m, 10°) / (0.5m, 10°) / (1m, 10°) respectively. For reference we include the result for parametric models (both with GT calib. and est. using [21]). Best result for each category highlighted in bold.

Reg.	kNN	ϱ	Intr. Calib.	Mean ϵ_{rms}	Median ϵ_{rms}	$<$ 1px
Diff.	1	Huber	Alg.1	6.881	0.693	326/490
Linear	4	Huber	Alg.1	0.847	0.418	365/490
Quadratic	4	Huber	Alg.1	1.208	0.466	349/490
Linear	2	Huber	Alg.1	0.881	0.413	362/490
Linear	4	Huber	Alg.1	0.847	0.418	365/490
Linear	6	Huber	Alg.1	0.839	0.424	364/490
Linear	8	Huber	Alg.1	0.841	0.420	364/490
Linear	10	Huber	Alg.1	0.842	0.424	**366/490**
Linear	4	ℓ_2	Alg.1	0.853	0.426	363/490
Linear	4	Huber	Alg.1	0.847	0.418	365/490
Linear	4	Cauchy	Alg.1	**0.830**	**0.409**	365/490
Linear	4	Huber	None	1.039	0.440	351/490
Linear	4	Huber	Alg.1	0.847	0.418	365/490
Linear	4	Huber	Med-3	1.000	0.436	350/490
Linear	4	Huber	Med-5	1.177	0.430	351/490

Table 5. Ablation study for the regularization function and intrinsic calibration method.

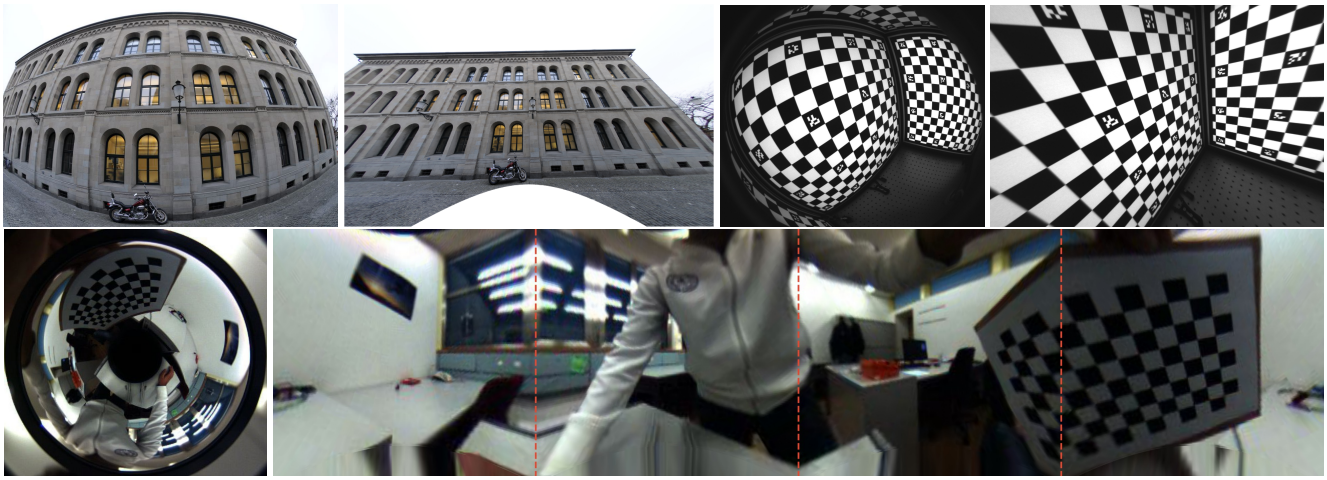

Figure 4. Qualitative results of implicit self-calibration. The images were not seen during calibration showing that our calibration does not overfit to the calibration data. For the catadioptric camera we undistorted the image to four 90° fov images orthogonal to the principal axis.

		Upgraded		Bundle Adjustment	
		mean	median	mean	median
Kirchenge	ϵ_{rot} (deg.)	**1.720**	**0.398**	1.726	0.403
	ϵ_{pos} (m.)	0.414	0.028	**0.401**	**0.026**
	ϵ^{GT}_{proj} (px)	2.199	1.630	**1.469**	**1.015**
	ϵ^{est}_{proj} (px)	1.633	0.955	**1.294**	**0.728**
Grossmunster	ϵ_{rot} (deg.)	2.322	**0.643**	**2.291**	0.716
	ϵ_{pos} (m.)	0.990	0.129	**0.939**	**0.093**
	ϵ^{GT}_{proj} (px)	2.866	2.110	**1.631**	**1.325**
	ϵ^{est}_{proj} (px)	2.866	2.110	**1.516**	**1.075**
Kazan	ϵ_{rot} (deg.)	0.463	0.463	**0.403**	**0.396**
	ϵ_{pos} (m.)	**0.053**	**0.048**	0.152	0.112
	ϵ^{GT}_{proj} (px)	1.359	1.084	**0.834**	**0.682**
	ϵ^{est}_{proj} (px)	0.957	0.720	**0.658**	**0.488**
Doge Palace	ϵ_{rot} (deg.)	0.401	0.393	**0.324**	**0.360**
	ϵ_{pos} (m.)	**0.040**	**0.033**	0.107	0.067
	ϵ^{GT}_{proj} (px)	**0.914**	**0.710**	1.091	0.981
	ϵ^{est}_{proj} (px)	0.893	0.690	**0.619**	**0.453**

Table 6. BA with implicit distortion model. Table shows the rotation (deg.) and position (m) errors. We also report reprojection errors, both with the GT calib. and estimated (Sec. 4.2)

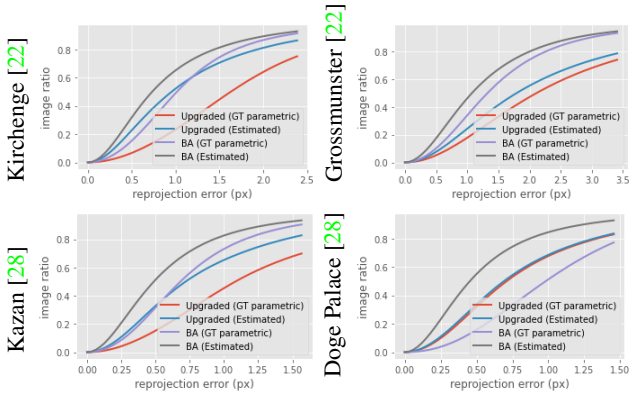

Figure 5. Bundle adjustment with implicit distortion. Cumulative reprojection errors using GT calib and estimated (Sec. 4.2).

6.4. Implicit Distortion in Bundle Adjustment

To evaluate the bundle adjustment, we use the 1D radial SfM framework from Larsson et al. [22] to reconstruct the datasets from Section 6.1. Since it solely relies on radial constraints, it only recovers (R, t_1, t_2) for each camera. We first upgrade the reconstruction using by estimating t_3 with the proposed method. Next, we perform full bundle adjustment as described in Section 5. Table 6 and Figure 5 shows how the reconstruction improves significantly as a result of the non-linear refinement. We believe the slightly worse results for Doge-Palace are due to the scene being close to degenerate (fronto-parallel plane) in some images, since we obtain low errors for reprojection but not camera pose.

7. Conclusions

In this paper we have presented a method for camera pose estimation which does not require the intrinsic calibration of the camera. This was achieved by regularizing the point-wise focal lengths, i.e. the relative scaling between the radius of the projections and image observations. The proposed method outperforms the previous work on model-free pose estimation [6], extending it to allow refinement of the full 6 degree-of-freedom camera pose and even bundle adjustment. In [22], the authors presented a *calibration-free* Structure-from-Motion framework, however it can only partially recover the camera poses. Using the proposed model we can upgrade these reconstructions to allow for the first truly calibration-free pipeline which recovers 6 DoF poses.

However, there is still a gap in performance for implicit distortion models compared to an explicit parametric one. In particular, the weaker constraints make it more challenging to filter outlier correspondences in difficult matching scenarios (see Sec. 6.2). Nevertheless, as the implicit model is mostly camera-agnostic and can be applied to any radially symmetric camera, we believe it can be a useful tool for bootstrapping the camera pose when both the intrinsic calibration and the appropriate model are unknown.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[2] Johannes Beck and Christoph Stiller. Generalized b-spline camera model. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 2

[3] Filippo Bergamasco, Luca Cosmo, Andrea Gasparetto, Andrea Albarelli, and Andrea Torsello. Parameter-free lens distortion calibration of central cameras. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[4] Duane C Brown. Decentering distortion of lenses. *Photogrammetric Engineering and Remote Sensing*, 1966. 2

[5] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *Asian Conference on Computer Vision (ACCV)*, 2010. 2

[6] Federico Camposeco, Torsten Sattler, and Marc Pollefeys. Non-parametric structure-based calibration of radially symmetric cameras. In *International Conference on Computer Vision (ICCV)*, 2015. 2, 3, 4, 5, 6, 7, 8

[7] Alexander Eugen Conrady. Decentred lens-systems. *Monthly notices of the royal astronomical society*, 79(5):384–390, 1919. 2

[8] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Computer Vision and Pattern Recognition (CVPR)*, 2001. 2

[9] Michael D Grossberg and Shree K Nayar. A general imaging model and a method for finding its parameters. In *International Conference on Computer Vision (ICCV)*, 2001. 2

[10] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2007. 2, 3, 4, 5, 7

[11] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. 4

[12] Jose Pedro Iglesias and Carl Olsson. Radial distortion invariant factorization for structure from motion. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[13] Klas Josephson and Martin Byröd. Pose estimation with radial distortion and unknown focal length. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[14] Jae-Hak Kim, Yuchao Dai, Hongdong Li, Xin Du, and Jonghyuk Kim. Multi-view 3d reconstruction from uncalibrated radially-symmetric cameras. In *International Conference on Computer Vision (ICCV)*, 2013. 2

[15] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *International Conference on Computer Vision (ICCV)*, 2013. 2, 4

[16] Zuzana Kukelova, Jan Heller, Martin Bujnak, Andrew W. Fitzgibbon, and Tomás Pajdla. Efficient solution to the epipolar geometry for radially distorted cameras. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[17] Zuzana Kukelova, Jan Heller, Martin Bujnak, and Tomás Pajdla. Radial distortion homography. *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[18] Zuzana Kukelova and Viktor Larsson. Radial distortion triangulation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[19] Zuzana Kukelova and Tomas Pajdla. A minimal solution to radial distortion autocalibration. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2011. 2

[20] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making minimal solvers for absolute pose estimation compact and robust. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[21] Viktor Larsson, Torsten Sattler, Zuzana Kukelova, and Marc Pollefeys. Revisiting radial distortion absolute pose. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 4, 7

[22] Viktor Larsson, Nicolas Zobernig, Kasim Taskin, and Marc Pollefeys. Calibration-free structure-from-motion with calibrated radial trifocal tensors. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 7, 8

[23] Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized ransac. In *British Machine Vision Conference (BMVC)*, 2012. 4, 7

[24] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 4, 5

[25] Yaroslava Lochman, Kostiantyn Liepieshov, Jianhui Chen, Michal Perdoch, Christopher Zach, and James Pritts. Babelcalib: A universal approach to calibrating central cameras. In *International Conference on Computer Vision (ICCV)*, 2021. 6

[26] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. 4, 5

[27] Pedro Miraldo and Helder Araujo. Calibration of smooth camera models. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2012. 2

[28] Carl Olsson and Olof Enqvist. Stable structure from motion for unordered image collections. In *Scandinavian Conference on Image Analysis (SCIA)*, 2011. 7, 8

[29] James Pritts, Zuzana Kukelova, Viktor Larsson, and Ondřej Chum. Radially-distorted conjugate translations. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[30] James Pritts, Zuzana Kukelova, Viktor Larsson, and Ondřej Chum. Rectification from radially-distorted scales. In *Asian Conference on Computer Vision (ACCV)*, 2018. 2

[31] Srikumar Ramalingam and Peter Sturm. A unifying model for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2016. 2

[32] Dennis Rosebrock and Friedrich M Wahl. Generic camera calibration and modeling using spline surfaces. In *IEEE Intelligent Vehicles Symposium*, 2012. 2

[33] Paul-Edouard Sarlin. Visual localization made easy with hloc. https://github.com/cvg/Hierarchical-Localization/. 6, 7

[34] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[36] Torsten Sattler et al. RansacLib - A Template-based *SAC Implementation, 2019. 7

[37] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions, 2018. 6, 7

[38] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *International Conference on Intelligent Robots and Systems (IROS)*, 2006. 2, 3, 4

[39] Thomas Schops, Viktor Larsson, Marc Pollefeys, and Torsten Sattler. Why having 10,000 parameters in your camera model is better than twelve. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[40] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis, 2018. 6, 7

[41] SriRam Thirthala and Marc Pollefeys. Radial multi-focal tensors. *International Journal of Computer Vision (IJCV)*, 2012. 2

[42] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding (CVIU)*, 2000. 4

[43] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, 1999. 5

[44] Roger Tsai. 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. 2, 4