A Calibration Method for the Generalized Imaging Model with Uncertain Calibration Target Coordinates

David Uhlig[0000−0003−1996−4419] and Michael Heizmann[0000−0001−9339−2055]

Institute of Industrial Information Technology, Karlsruhe Institute of Technology, Germany
{david.uhlig,michael.heizmann}@kit.edu

Abstract. The developments in optical metrology and computer vision require more and more advanced camera models. Their geometric calibration is of essential importance. Usually, low-dimensional models are used, which however often have insufficient accuracy for the respective applications. A more sophisticated approach uses the generalized camera model. Here, each pixel is described individually by its geometric ray properties. Our efforts in this article strive to improve this model. Hence, we propose a new approach for calibration. Moreover, we show how the immense number of parameters can be efficiently calculated and how the measurement uncertainties of reference features can be effectively utilized. We demonstrate the benefits of our method through an extensive evaluation of different cameras, namely a standard webcam and a microlens-based light field camera.

1 Introduction

Accurate optical measurement methods are becoming increasingly important for high-precision manufacturing. The rising demand can be satisfied by modern imaging systems with advanced optics. The exact geometric calibration of these systems is of essential importance for computer vision and optical metrology. Most systems use perspective projection with a single projection center and are referred to as central cameras. They can often be described by low-dimensional, parametric models with few intrinsic parameters, *e.g.* the well-known pinhole model which even can compensate imperfections of the system, such as lens aberrations, with the help of polynomial correction parameters [1]. In some applications in the field of optical metrology, more complex imaging systems are needed. These can often no longer be described by a central camera model and are in many cases non-parametric and non-central, *e.g.* multi camera systems, catadioptric cameras or light field cameras [2–5]. Here, more sophisticated models are needed, which always have to be precisely adapted to the specific camera.

The disadvantage of low-dimensional models is that they have poor explanatory power and in modern cameras not every pixel of the many millions can be perfectly described by these models. The more complexity an imaging system has, the more difficult it becomes to model it. The more elaborate the optical elements are, the more challenging it becomes to find a mathematically adequate mapping between the light of the captured scene and the physical sensor plane of the camera. Consequently, in the recent years, the lack of flexibility and precision has led to the development of new camera models, where cameras can be described as generalized imaging systems, which are independent of the specific camera type and allow high-precision calibration.

The generalized camera model was originally introduced in the work of Grossberg and Nayar [6, 7]. An arbitrary imaging system is modeled as a non-parametric discrete black box containing photosensitive elements. Each pixel collects light from a bundle of rays that enter the imaging system, referred to as *raxel* which consists of geometrical ray coordinates and radiometric parameters. The set of all *raxels* builds the complete generalized imaging model. The authors perform the calibration by measuring the intersection of camera rays with known reference targets: a monitor that is moved by a linear translation stage with known steps. Sturm and Ramalingam [8, 9] and Ramalingam *et al.* [10] excluded the radiometric properties and proposed a calibration of the generalized model where poses may be unknown. A closed form solution can be obtained, if the same pixel sees three points of the reference objects. The downside of their method is that the ray distribution of the camera has to be known in advance. For example, different models apply when the imaging system is non-central or a perspective camera and complicated parametrization steps are necessary. Bothe *et al.* [11] and Miraldo *et al.* [12] achieve pixel wise calibration by bypassing the estimation of the target pose by simply tracking it using an external stereo-camera-system or an IR tracker, respectively. Some work has been done to simplify the calibration by reducing the number of parameters by, *e.g.*, fitting a spline surface onto the set of rays [13, 14]. Thus, the camera is evaluated on a subset of control points. However, this only works when the imaging system is smooth, *i.e.* multi camera systems, light field cameras or more complex optical systems are excluded. The work most similar to ours is that of Bergamasco *et al.* [15,16]. They assume unknown poses and calibrate the camera by iteratively calculating the projection of the rays onto a coded calibration monitor, and by minimizing the resulting coding-error on a pixel level. In a second step, they estimate the reference pose using an adapted iterative closest point method [17]. However, they don't use a unified global objective function during the minimization of the calibration error and without proper initialization their method tends to diverge.

Due to these disadvantages, we present in this article a new method to calibrate the generalized imaging model, which improves the work of Bergamasco *et al.* Our goal is to find a flexible calibration procedure that can accurately describe the geometrical properties of an arbitrary imaging system. In the end, however, one does not obtain an "image", but rather a set of rays with corresponding intensities. Still, this does not interfere with most applications in optical metrology, *e.g.*, profilometry, deflectometry or laser triangulation, where only the geometric ray properties are relevant [18–20]. For our method, we assume unknown poses of the calibration target and iteratively solve the subproblems of camera calibration and pose estimation, without the use of an additional translation or rotation stage. By processing every pixel individually and updating each pose one at a time, we can efficiently reduce the computational costs, whereby every camera ray and each observed point contribute to the result. Our main contributions are the following. We present a closed form least squares solution for the ray calibration subproblem. We correctly solve the pose estimation subproblem using a gradient descend optimization on the rotation manifold. And most importantly, we propose to use the measurement uncertainty of the target feature to increase the accuracy of the entire camera calibration. All is achieved by minimizing a single objective function, where convergence can be always achieved. Finally, acceleration techniques are applied to obtain an almost quadratic convergence rate.

In the next sections we outline the basic calibration procedure and its difficulties. We present how to measure reference target features and their corresponding uncertainties. After describing how to include this uncertainty in the optimization procedure, we will explain the individual calibration steps in more detail. Finally, experiments validate the accuracy of the proposed calibration method by analyzing different camera systems.

2 Background

2.1 Plücker-Line

The portion of the light that is sampled by a single pixel has a cone-like shaped expansion due to the effects of the depth-of-field. For simplicity's sake, one models a *raxel* as a ray running through the center of this cone along the direction of light propagation. There are various possibilities for a mathematical description of rays, but in this work the concept of Plücker-coordinates is used [21, 22]. In 6D-Plücker-space a Plücker-line $\mathbf{L} \in \mathbb{P}^6$ is defined by its direction $\mathbf{d} \in \mathbb{R}^3$ and its moment $\mathbf{m} \in \mathbb{R}^3$. A line in 3D-space has four degrees of freedom, therefore two constraints apply to the Plücker-line:

$$\mathbb{P}^6 = \left\{ \begin{pmatrix} \mathbf{d} \\ \mathbf{m} \end{pmatrix} \middle| \, \mathbf{d}\,, \mathbf{m} \in \mathbb{R}^3,\, \mathbf{d}^{\mathrm{T}}\mathbf{m} = 0\,, \|\mathbf{d}\| = 1 \right\} . \tag{1}$$

The moment can be calculated with $\mathbf{m} = \mathbf{p} \times \mathbf{d}$, where $\mathbf{p} \in \mathbb{R}^3$ is an arbitrary point on the line $\mathbf{L}$. The moment vector stands perpendicular on the line and its norm $\|\mathbf{m}\|$ corresponds to the Euclidean distance of the line to the origin. The Euclidean distance of a line $\mathbf{L}$ to an arbitrary point $\mathbf{x}$ can be found by calculating the distance to the closest point on the line, which results in [23]:

$$d(\mathbf{L}, \mathbf{x}) = \|\mathbf{x} \times \mathbf{d} - \mathbf{m}\| \ . \tag{2}$$

2.2 Alternating Minimization

Now, calibrating the camera just means to estimate for every single pixel its ray $\mathbf{L}$, with direction $\mathbf{d}$ and moment $\mathbf{m}$. In conclusion, we are looking for ray parameters that minimize a suitable distance measure between the camera rays and observed reference points, whereby the positions of the references are assumed to be unknown. The calibration can now be formulated in the sense of a least squares problem. Fig. 1 illustrates the approach.

$$f(\mathcal{R}, \mathcal{T}, \mathcal{L}) = \sum_{k,i} d\left(\mathbf{p}_{ik}, \mathbf{L}_i\right)^2 \ . \tag{3}$$

Here, index i represents the individual rays and index k the reference coordinate system. $d(\cdot)$ is a suitable ray-to-point distance measure and $\mathbf{p}_{ik} = \mathbf{R}_k \mathbf{x}_{ik} + \mathbf{t}_k$ are the observed features in 3D-space, where $\mathbf{x}_{ik}$ is a local point on a reference target. $\mathbf{R}_k, \mathbf{t}_k$ are the corresponding transformations to the camera coordinate system. And for the remainder of this article, we define $\mathcal{R} := \{\mathbf{R}_1, \mathbf{R}_2, \dots\}, \mathcal{T} := \{\mathbf{t}_1, \mathbf{t}_2, \dots\}, \mathcal{L} := \{\mathbf{L}_1, \mathbf{L}_2, \dots\}$.

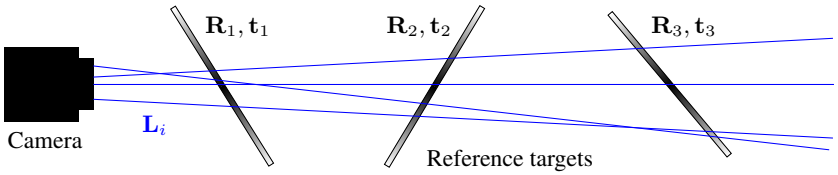


Fig. 1: Generalized calibration: The imaging system is treated as a black box that is independently of the internal optics described by a set of vision rays. Each individual ray observes the intersected reference target point. The ideal calibration results in a minimal distance between rays and points.

Regardless of the actual used distance measure, it is very difficult to minimize such a problem in a reasonable time and with appropriate use of computational resources. The ray model with six parameters and two constraints has four degrees of freedom per pixel. Even for today's standard cameras, this leads to a huge number of parameters that have to be optimized (*e.g.* a 40-megapixel camera has 240 million parameters). In addition, the reference target pose is in general not known. Thus, at the same time six degrees of freedom per pose have to be estimated. The coupling of poses and rays and the immense number of parameters results in an extremely high dimensional problem that cannot be solved using a single optimization method. The calculation of a gradient or a Hessian and the corresponding function evaluations would be computationally too expensive.

Therefore it is useful to divide the problem into subproblems and then solve them iteratively in the sense of an *Alternating Minimization* (AM) [24]. Accordingly, problem (3) is split into a camera calibration and a reference target pose estimation. The approach of an AM is to fix a parameter set and to solve the resulting problem. This way one has two particular problems to solve in each iteration:

$$\mathbf{L}_i^{(n)} = \underset{\mathbf{L}_i \in \mathbb{P}^6}{\arg\min} \quad f\left(\mathcal{R}^{(n-1)}, \mathcal{T}^{(n-1)}, \mathbf{L}_i\right) , \tag{4}$$

$$\mathbf{R}_k^{(n)}, \mathbf{t}_k^{(n)} = \underset{(\mathbf{R}_k, \mathbf{t}_k) \in \mathrm{SE}(3)}{\arg\min} \quad f\left(\mathbf{R}_k, \mathbf{t}_k, \mathcal{L}^{(n)}\right) , \tag{5}$$

where an appropriate initialization $\mathcal{R}^{(0)}, \mathcal{T}^{(0)}$ has to be chosen. The first problem is solved for each pixel i individually by fixing all of the reference target poses and the second one is solved for each single pose k by assuming fixed ray parameters.

2.3 Dense Feature Acquisition

To present the camera calibration as a per pixel problem and to treat each pixel independently from its neighbors, sufficient observations of reference features have to be available for every pixel. However, the widely used checkerboard patterns can provide only sparse features which aren't nearly enough for a generalized camera calibration. Instead, it is a good idea to use active targets, *e.g.* flat monitor displays, and active encoding strategies, to assign each camera ray a 2D point in the local reference target plane. Thus, each ray can observe one feature per pose.

The detection of features in the reference target plane and with it the registration of camera rays $\mathbf{L}_i$ to monitor display points $\mathbf{x}_{ik}$ is found in our approach via a temporal coding of the monitor pixels. Phase shift methods are particularly suitable because they are robust against noise, low-pass filtering by defocusing of the camera and variation of ambient light [25]. They encode a normalized reference coordinate $x \in [0, 1]$, where the encoding is done in horizontal and vertical direction separately, by generating a signal sequence of K sinusoidal patterns shifted by $\Psi_k = \frac{2\pi k}{K}$ with phase $\varphi(x) = 2\pi f x$:

$$g_k(x) = A + B\cos\left(\varphi(x) + \Psi_k\right). \tag{6}$$

The camera records the signal sequence $\tilde{g}_k$ with $k = 0 \ldots K-1$, calculates the phase and decodes the reference coordinate from it $x = \frac{\varphi}{2\pi f}$, with:

$$\varphi = \operatorname{atan2}(-a, b) \bmod 2\pi, \quad a = \sum_{k=0}^{K-1} \tilde{g}_k \sin\left(\Psi_k\right), \quad b = \sum_{k=0}^{K-1} \tilde{g}_k \cos\left(\Psi_k\right). \tag{7}$$

For symmetric K step phase shift methods, the uncertainty of the phase measurement σ_φ can be specified as a function of the sensor noise σ_I [26]:

$$\sigma_\varphi = \frac{1}{f}\sqrt{\frac{2}{K}}\frac{\sigma_I}{\widehat{B}}, \tag{8}$$

where $\widehat{B} = \frac{2}{K}\sqrt{a^2 + b^2}$ is an estimate of the modulation B of the signal. By using multi frequency phase shift coding methods [27], dense features and their local coordinates can be found for every camera ray in subpixel precision, with respect to monitor pixels. In addition, the uncertainty of this measurement can be quantified and used in the subsequent calibration procedure. The interested reader is advised to refer to the literature for more details [26, 28].

3 Generalized Camera Calibration

With the previous results, we can define an objective function that needs to be minimized in order to calibrate the camera and find all ray parameters $\mathbf{d}_i, \mathbf{m}_i$. Simultaneously, we estimate the pose of the calibration targets $\mathbf{R}_k, \mathbf{t}_k$ with respect to the camera. This is done in a weighted least squares sense by minimizing the distance between uncertain target points $\mathbf{x}_{ik}$ with uncertainty σ_{ik} and their corresponding camera rays. To this end, we utilize the phase shift coding strategy to estimate the uncertainties of the reference target points which results in a weighting factor $w_{ik} = \sigma_{ik}^{-2}$. In conclusion, we obtain the objective function:

$$f(\mathcal{R}, \mathcal{T}, \mathcal{L}) = \sum_{i,k} w_{ik} \left\| (\mathbf{R}_k \mathbf{x}_{ik} + \mathbf{t}_k) \times \mathbf{d}_i - \mathbf{m}_i \right\|^2. \tag{9}$$

As mentioned before, even with today's computing power, it is very difficult (or even impossible) to estimate the enormous number of parameters of this problem simultaneously and in a reasonable time. We therefore divide it into two subproblems, one for camera calibration and another for pose estimation, which are then handled iteratively. This allows to solve the subproblems more easily and to get an optimal result, which further leads to the overall problem converging towards a solution.

3.1 Camera Ray Calibration

One step in the camera calibration procedure is to estimate the ray parameters by assuming known poses of the calibration targets. This greatly reduces the complexity. Instead of calculating every parameter at once, we can calibrate the ray $\mathbf{L}_i = (\mathbf{d}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}} \in \mathcal{L}$ of each pixel individually (or in parallel). Hence, for every single ray we obtain a new optimization problem which we can write in a more compact form:

$$\begin{aligned} f(\mathbf{d}_i, \mathbf{m}_i) &= \sum_k w_{ik} \left\| \mathbf{p}_{ik} \times \mathbf{d}_i - \mathbf{m}_i \right\|^2 \\ &= \mathbf{d}_i^{\mathrm{T}} \mathbf{A}_{\mathrm{dd},i} \mathbf{d}_i + \mathbf{m}_i^{\mathrm{T}} \mathbf{A}_{\mathrm{md},i} \mathbf{d}_i + a_{\mathrm{mm},i} \|\mathbf{m}_i\|^2 \,. \end{aligned} \tag{10}$$

Here, $\mathbf{p}_{ik} = \mathbf{R}_k \mathbf{x}_{ik} + \mathbf{t}_k$ represents the target point in camera coordinates and $\mathbf{A}_{\mathrm{dd},i} = \sum_k w_{ik} \left[\mathbf{p}_{ik}\right]_\times^{\mathrm{T}} \left[\mathbf{p}_{ik}\right]_\times$, $\mathbf{A}_{\mathrm{md},i} = \sum_k 2 w_{ik} \left[\mathbf{p}_{ik}\right]_\times^{\mathrm{T}}$, $a_{\mathrm{mm},i} = \sum_k w_{ik}$ are found by summing over the pose index k, reordering and extracting the ray parameters $\mathbf{d}_i$ and $\mathbf{m}_i$. In addition, for better readability, we neglect the index i in the remainder of this section.

It can be easily shown that $\mathbf{A}_{\mathrm{dd}}$ is almost always positive definite and invertible. Thus, problem (10) is convex. Considering the characteristics of the Plücker-rays (1), finding the optimal rays results in minimizing a quadratic program with quadratic equality constraints: $\|\mathbf{d}\| = 1$, $\mathbf{d}^{\mathrm{T}}\mathbf{m} = 0$. Although the minimization of such a problem in general requires a difficult nonlinear minimization, we are able to find a global minimum in this specific case, using a few simple steps.

At first, it should be obvious that the solution of the constraint problem is scale ambiguous and that the norm of the ray direction $\|\mathbf{d}\|$ does not influence the actual ray properties [22]. Thus, after having found a solution, we can apply a normalization to the ray $\mathbf{L}_{\mathrm{n}} = \mathbf{L} / \|\mathbf{d}\| = (\mathbf{d}/\|\mathbf{d}\|, \mathbf{m}/\|\mathbf{d}\|)$ to obtain a geometrical meaningful point-to-ray distance (2). To deal with the inequality constraints, we formulate the Lagrangian:

$$g = \mathbf{d}^{\mathrm{T}} \mathbf{A}_{\mathrm{dd}} \mathbf{d} + \mathbf{m}^{\mathrm{T}} \mathbf{A}_{\mathrm{md}} \mathbf{d} + a_{\mathrm{mm}} \|\mathbf{m}\|^2 + \lambda \mathbf{d}^{\mathrm{T}} \mathbf{m} + \mu \left(\mathbf{d}^{\mathrm{T}} \mathbf{d} - 1 \right) , \tag{11}$$

with the Lagrange multipliers λ, μ. Further, the first order conditions for a minimum are:

$$\frac{\partial g}{\partial \mathbf{d}} = 2\mathbf{A}_{\mathrm{dd}} \mathbf{d} + \mathbf{A}_{\mathrm{md}}^{\mathrm{T}} \mathbf{m} + \lambda \mathbf{m} + 2\mu \mathbf{d} \stackrel{!}{=} \mathbf{0} \,, \tag{12}$$

$$\frac{\partial g}{\partial \mathbf{m}} = 2a_{\mathrm{mm}} \mathbf{m} + \mathbf{A}_{\mathrm{md}} \mathbf{d} + \lambda \mathbf{d} \stackrel{!}{=} \mathbf{0} \,, \tag{13}$$

$$\frac{\partial g}{\partial \lambda} = \mathbf{d}^{\mathrm{T}} \mathbf{m} \stackrel{!}{=} 0 \,, \tag{14}$$

$$\frac{\partial g}{\partial \mu} = \|\mathbf{d}\|^2 - 1 \stackrel{!}{=} 0 \,. \tag{15}$$

Using (13) and (14), this results in a solution for the ray moment $\mathbf{m}$ and λ:

$$\mathbf{m} = -\frac{1}{2a_{\mathrm{mm}}} \left(\mathbf{A}_{\mathrm{md}} + \lambda \mathbf{I} \right) \mathbf{d} \,, \tag{16}$$

$$\mathbf{d}^{\mathrm{T}} \mathbf{m} = -\frac{1}{2a_{\mathrm{mm}}} \mathbf{d}^{\mathrm{T}} \left(\mathbf{A}_{\mathrm{md}} + \lambda \mathbf{I} \right) \mathbf{d} \stackrel{!}{=} 0 \,, \tag{17}$$

$$\begin{aligned} \Rightarrow \lambda = \frac{\mathbf{d}^{\mathrm{T}} \mathbf{A}_{\mathrm{md}} \mathbf{d}}{\mathbf{d}^{\mathrm{T}} \mathbf{d}} &\stackrel{(15)}{=} \mathbf{d}^{\mathrm{T}} \mathbf{A}_{\mathrm{md}} \mathbf{d} = \mathbf{d}^{\mathrm{T}} \left(\sum_k 2 w_{ik} \left[\mathbf{p}_{ik}\right]_\times^{\mathrm{T}} \right) \mathbf{d} \\ &= -\mathbf{d}^{\mathrm{T}} \left(\left(\textstyle\sum_k 2 w_{ik} \mathbf{p}_{ik} \right) \times \mathbf{d} \right) = 0 \,, \end{aligned} \tag{18}$$

where the last equation holds because $\mathbf{d}$ is orthogonal to $\mathbf{p} \times \mathbf{d}\,,\ \forall\ \mathbf{p} \in \mathbb{R}^3$. Inserting these results into (12) leads to a simple eigenvalue problem for the solution of the ray direction $\mathbf{d}$ and Lagrange multiplier μ:

$$\left(\mathbf{A}_{\mathrm{dd}} - \frac{1}{4a_{\mathrm{mm}}}\mathbf{A}_{\mathrm{md}}^{\mathrm{T}}\mathbf{A}_{\mathrm{md}}\right)\mathbf{d} = \mu\mathbf{d}\,. \tag{19}$$

These equations still contain the trivial solution $\mathbf{d} = \mathbf{m} = \mathbf{0}$ which however has no geometric meaning for the calibration and is excluded by (15). Apart from that, the solution space of (19) consists of three eigenvalues μ_j with corresponding eigenvectors $\mathbf{d}_j$. After estimating a possible $\mathbf{d}_j$ and corresponding Lagrange multiplier μ_j, we need to scale the eigenvalue probem to normalize the ray such that $\|\mathbf{d}_j\| = 1$ in order to keep the geometrical meaning of (2) and to obtain an unambiguous scaling. We can get the corresponding ray momentum $\mathbf{m}_j$ through (16). And finally, from these at most three possible stationary points, we select the one with the smallest objective function value (11) to be the optimal solution. In conclusion, we find a closed form solution for the least squares problem of the weighted ray-to-point distance minimization.

3.2 Generalized Pose Estimation

As before, the estimation of the calibration target pose can drastically be simplified by assuming known ray parameters. Therefore we optimize each pose individually. The objective function for each pose k becomes:

$$f(\mathbf{R}_k, \mathbf{t}_k) = \sum_i w_{ik} \left\|(\mathbf{R}_k\mathbf{x}_{ik} + \mathbf{t}_k) \times \mathbf{d}_i - \mathbf{m}_i\right\|^2 . \tag{20}$$

However, solving for a pose $\mathbf{R}_k, \mathbf{t}_k$ is non-trivial because the solution space is restricted to the special Euclidean group $\mathrm{SE}(3)$, which combines rotations and translations in three dimensions, $\mathbf{R}_k \in \mathrm{SO}(3)$ and $\mathbf{t}_k \in \mathbb{R}^3$, respectively. Directly applying a nonlinear optimization procedure is not advisable, because every function evaluation results in the summation over all rays and is thus computationally very expensive. Therefore, as before, we need to find a more compact form of this quadratic function. Again for the sake of brevity, we omit the index k for the remainder of this section. Further, we use the vectorization operator $\mathbf{r} = \mathrm{vec}\,(\mathbf{R}) \in \mathbb{R}^9$ that stacks the columns of the 3×3 matrix $\mathbf{R}$. While computing the summation over all ray indices i only once, we obtain independence of the actual number of rays, which simplifies and speeds up later optimization steps:

$$\begin{aligned} \min f(\mathbf{R}, \mathbf{t}) &= \mathbf{r}^{\mathrm{T}}\mathbf{A}_{\mathrm{rr}}\mathbf{r} + \mathbf{t}^{\mathrm{T}}\mathbf{A}_{\mathrm{tt}}\mathbf{t} + \mathbf{t}^{\mathrm{T}}\mathbf{A}_{\mathrm{tr}}\mathbf{r} + \mathbf{b}_{\mathrm{r}}^{\mathrm{T}}\mathbf{r} + \mathbf{b}_{\mathrm{t}}^{\mathrm{T}}\mathbf{t} + h \\ \text{s.t. } \mathbf{r} &= \mathrm{vec}\,(\mathbf{R})\,,\ (\mathbf{R}, \mathbf{t}) \in \mathrm{SE}(3)\,. \end{aligned} \tag{21}$$

While observing the constraint quadratic objective (21), we notice that the main constraint lies in the rotational part and moreover the objective is convex in the translation part. Thus, the problem can further be reduced by decoupling of translation and rotation, which means that $\mathbf{t}$ can be expressed in terms of $\mathbf{R}$. We can find the optimal translation vector with the first order condition for a minimum, using $\frac{\partial f(\mathbf{R},\mathbf{t})}{\partial \mathbf{t}} \stackrel{!}{=} 0$. This leads to:

$$\mathbf{t} = -\frac{1}{2}\mathbf{A}_{\mathrm{tt}}^{-1}\left(\mathbf{A}_{\mathrm{tr}}\mathbf{r} + \mathbf{b}_{\mathrm{t}}\right) . \tag{22}$$

Inserting (22) into (21) results in the decoupling of the rotation and translation subproblem, which then again yields a quadratic optimization problem ($\mathbf{A}, \mathbf{b}, c$ are calculated from $\mathbf{A}_{\mathrm{rr}}, \mathbf{A}_{\mathrm{tt}}, \mathbf{A}_{\mathrm{tr}}, \mathbf{b}_{\mathrm{r}}, \mathbf{b}_{\mathrm{t}}, h$, see supplemental material for more details):

$$f(\mathbf{R}) = \mathbf{r}^{\mathrm{T}}\mathbf{A}\mathbf{r} + \mathbf{b}^{\mathrm{T}}\mathbf{r} + c\,, \quad \text{s.t. } \mathbf{r} = \mathrm{vec}\,(\mathbf{R})\,,\ \mathbf{R} \in \mathrm{SO}(3)\,. \tag{23}$$

After finding a solution for the rotation matrix we obtain the optimal translation vector with (22), assuming invertibility of $\mathbf{A}_{\mathrm{tt}}$. It can be shown that $\mathbf{A}_{\mathrm{tt}}$ is positive definite in most cases with the exception of a few exotic camera ray distributions (*e.g.* parallel rays, telecentric optics), and hence we truly get the minimum of the objective with respect to the translation.

Although minimization of (23) seems simple at first, we have the constraint to find an optimum in $\mathrm{SO}(3)$. This is equivalent to a non-convex problem with quadratic and cubic constraints on the rotation parameters. Bergamasco *et al.* [15] use an iterative closest point algorithm that iteratively calculates the transformation from the observed points to the closest point on the corresponding rays, which however only converges near the optimum. Kanatani [29] suggests a fast method by first calculating an Euclidean solution, with $\mathbf{R} \in \mathbb{R}^{3\times3}$, and projecting it onto the $\mathrm{SO}(3)$-manifold using singular value decomposition, which results in a not entirely correct minimization. Schweighofer and Pinz [30] solve the problem using sum-of-squares optimization and constrain the algorithm to $\mathrm{SO}(3)$. Ventura *et al.* [31] and Kneip *et al.* [32] find a solution computationally very efficient, using Gröbner basis [33]. This however requires further "root polishing" to resolve ambiguities and to achieve good results. Hence, since our main focus is not real-time optimization, but rather highly precise pose estimation, we are obliged to find an accurate minimum to ensure convergence of the AM calibration. In order to find this, we therefore directly use a gradient-based optimization approach on the Riemannian manifold $\mathrm{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} \,|\, \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}$, which implicitly considers all constraints. The mapping from any element of the tangent space $\boldsymbol{\eta} \in \mathfrak{so}(3)$ to $\mathbf{R} \in \mathrm{SO}(3)$ is called the exponential map $\mathbf{R} = \mathrm{e}^{[\boldsymbol{\eta}]_\times}$, and the reverse map is called the logarithmic map $[\boldsymbol{\eta}]_\times = \log(\mathbf{R})$. Both can be calculated in closed form, using the skew operator [34]:

$$[\boldsymbol{\eta}]_\times = \begin{bmatrix} 0 & -\eta_3 & \eta_2 \\ \eta_3 & 0 & -\eta_1 \\ -\eta_2 & \eta_1 & 0 \end{bmatrix}. \tag{24}$$

If a function is to be optimized on the manifold, the corresponding direction of descent must be sought in the local tangent space $f_{\boldsymbol{\eta}}(\mathbf{R}) = f(\mathrm{e}^{[\boldsymbol{\eta}]_\times}\mathbf{R})$. In order to use conventional optimization methods, a valid representation for both the gradient and the Hessian must be identified. These can be easily found by using directional derivatives of the locally parameterized manifold in the direction of the tangent space [35]:

$$Df_{\boldsymbol{\eta}}(\mathbf{R})[\boldsymbol{\eta}] = \lim_{\varepsilon\to0} \frac{\partial}{\partial\varepsilon} f_{\boldsymbol{\eta}\varepsilon}(\mathbf{R}) = \boldsymbol{\eta}^{\mathrm{T}}\mathrm{grad}(f)\,, \tag{25}$$

$$D\,\mathrm{grad}(f)[\boldsymbol{\eta}] = \lim_{\varepsilon\to0} \boldsymbol{\eta}^{\mathrm{T}} \frac{\partial}{\partial\varepsilon}\mathrm{grad}(f) = \boldsymbol{\eta}^{\mathrm{T}}\mathrm{Hess}(f)\boldsymbol{\eta}\,. \tag{26}$$

Looking back at our original problem (23), this approach leads to the explicit formulas for the Riemannian gradient and Riemannian Hessian (the interested reader is advised

to refer to the supplementary material or the literature for further details [35, 36]):

$$\mathrm{grad}(f) = 2\mathbf{H}^{\mathrm{T}} \left(\mathbf{R} \otimes \mathbf{I}\right) \left(\mathbf{A}\mathbf{r} + \mathbf{b}\right) , \tag{27}$$

$$\mathrm{Hess}(f) = 2\mathbf{H}^{\mathrm{T}} \left(\left(\mathbf{R} \otimes \mathbf{I}\right) \mathbf{A} \left(\mathbf{R} \otimes \mathbf{I}\right)^{\mathrm{T}} - \mathbf{I} \otimes \mathrm{mat}\left(\mathbf{A}\mathbf{r} + \mathbf{b}\right) \mathbf{R}^{\mathrm{T}} \right) \mathbf{H} . \tag{28}$$

Here, the reshape operator $\mathrm{mat}(\cdot)$ is the inverse of $\mathrm{vec}(\cdot)$. $\otimes$ is the Kronecker product, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are unit base vectors and $\mathbf{H} = \left[\mathrm{vec}([\mathbf{e}_1]_\times), \mathrm{vec}([\mathbf{e}_2]_\times), \mathrm{vec}([\mathbf{e}_3]_\times)\right] \in \mathbb{R}^{9\times 3}$.

After the formulas for the gradient and the Hesse matrix have been established, we can build a quadratic model of the local tangent space and are then able to minimize the objective (23) with the help of an appropriate Newton descend algorithm. Apart from important differences, the procedure is quite similar to the classic Euclidean approach. For the current iterate, we calculate $\mathrm{grad} f(\mathbf{R}^{(n)})$ and $\mathrm{Hess} f(\mathbf{R}^{(n)})$. After the search direction $\boldsymbol{\eta}^{(n)}$ has been found by solving the Newton equation, one has to calculate a projection of the tangent space back to the manifold to obtain a valid descend [36]:

$$\mathrm{Hess} f(\mathbf{R}^{(n)})\, \boldsymbol{\eta}^{(n)} = -\mathrm{grad} f(\mathbf{R}^{(n)}) , \tag{29}$$

$$\mathbf{R}^{(n+1)} = \mathrm{e}^{\left(\alpha [\boldsymbol{\eta}^{(n)}]_\times\right)} \mathbf{R}^{(n)} . \tag{30}$$

Finally, a subsequent 1D backtracking line search in $\mathrm{SO}(3)$ finds a sufficient step size α and accelerates the convergence. In order to initialize the algorithm an appropriate start is required, where in the context of an AM-camera-calibration, the pose estimate from the previous iteration may be used.

3.3 Convergence, Acceleration and Summary

The camera ray calibration provides the globally optimal solution in every step. Furthermore, the pose estimation converges towards a minimum and provides no inferior result than the previous iteration. Following the researches in the field of AM [24, 37], it is easy to show the convergence of the optimization procedure to a stationary point with an $\mathcal{O}\left(\frac{1}{n}\right)$ convergence rate (see supplementary material). In order to obtain a faster convergence, acceleration techniques may be applied. We modified Nesterov's acceleration scheme to obtain an almost $\mathcal{O}\left(\frac{1}{n^2}\right)$ convergence rate [38]. During the acceleration step, a weighted rate of the change of the pose parameters is added to the next estimate. When accelerating the rotation, of course, this has to be done on the $\mathrm{SO}(3)$-manifold: The current rotation is reversed by the previous rotation, projected onto the $\mathfrak{so}(3)$ tangent space using the $\log$-map, weighted by an acceleration parameter and finally transformed back into a rotation matrix using the $\exp$-map and multiplied onto the current estimate. Algorithm 1 summarizes the complete AM calibration.

Although we have a strictly convergent algorithm, of course, there does not exist a unique solution. Depending on the starting value, the optimization runs into an arbitrary coordinate system. Therefore, it is advisable to initialize the algorithm with a rough estimate of the reference target poses, which could for example be obtained using standard model-based approaches presented in the literature [1, 39] or the generalized approach by Ramalingam *et al.* [10]. However, here it is of utmost importance that the camera model is properly chosen. Of course, one can also randomly select starting

Algorithm 1 Alternating minimization

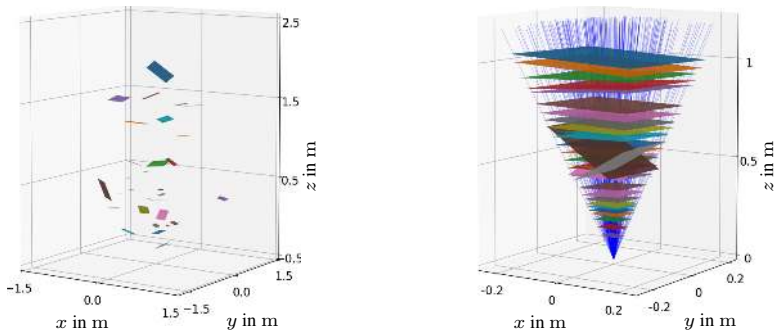
Input: For every pixel i and target pose k: measure monitor coordinates $\mathbf{x}_{ik}$ and weight w_{ik}
Output: Calibrated ray $\mathbf{L}_i$ for each pixel and pose $\mathbf{R}_k, \mathbf{t}_k$ of all reference targets
Initialize: Set poses of reference targets $\mathcal{R}^{(0)}, \mathcal{T}^{(0)}$ (*e.g.* use Ramalingam *et al.* [10])
Set acceleration parameter $\beta_0 = 0$

for $n = 1, 2, 3, \dots$ **do**
  for $i = 1, 2, 3, \dots$ **do**
    Hold pose parameters and optimize rays
    $\mathbf{L}_i^{(n+1)} = \underset{\mathbf{L}_i \in \mathbb{P}^6}{\arg\min}\, f\left(\mathcal{R}^{(n)}, \mathcal{T}^{(n)}, \mathbf{L}_i\right)$
  end for
  for $k = 1, 2, 3, \dots$ **do**
    Hold ray parameters and optimize poses
    $\mathbf{R}_k^*, \mathbf{t}_k^* = \underset{(\mathbf{R}_k, \mathbf{t}_k) \in \mathrm{SE}(3)}{\arg\min}\, f\left(\mathbf{R}_k, \mathbf{t}_k, \mathcal{L}^{(n+1)}\right)$
    Accelerate translation and rotation update
    $\mathbf{t}_k^{(n+1)} = \mathbf{t}_k^* + \beta_n \left(\mathbf{t}_k^* - \mathbf{t}_k^{(n)}\right)$
    $\mathbf{R}_k^{(n+1)} = \mathrm{e}^{\beta_n \log\left(\mathbf{R}_k^{\mathrm{T}(n)} \mathbf{R}_k^*\right)} \mathbf{R}_k^*$
    with a sequence β_n as defined by Nesterov [38].
  end for
end for

poses with the downside of an increased optimization time and the risk to converge to a non-optimal local minimum. Nonetheless, the arbitrary coordinate system doesn't change the geometrical properties of the rays and, accordingly, the calibrated camera can be used without loss of accuracy. Even more, the final calibration can be easily transformed to a standardized coordinate system, *e.g.* by defining the origin to be the point that is closest to all rays and by selecting the z-axis to be the mean ray direction.

4 Experiments

For the evaluation, a 27" monitor with a resolution of 2560×1440 and a pixel pitch of $233\,\mu\mathrm{m}$ was used to display the necessary calibration patterns. Two different imaging systems were used to evaluate the proposed method: A standard webcam (Logitech C920 HD Pro Webcam) and a more exotic example, a microlens-based light field camera (Lytro Illum). The latter one has a microlens array placed in front of the sensor, which allows to sample the plenoptic function of a scene [3, 40–42]. This ultimately results in a non-central camera with multiple projection centers, which can be used as a 3D-camera and in addition requires a much more complex camera model to be efficiently calibrated. The monitor was captured from 30 different poses, whereby several phase shift patterns have to be recorded at each pose to encode the target features [28]. The distances between monitor and camera were in the range of $5\,\mathrm{cm}$ to $2\,\mathrm{m}$. In order to compare the proposed technique to the classic methods, checkerboard patterns were displayed at the same positions. The webcam was calibrated using OpenCV [39] and Zhang's algorithm [1]. The light field camera was calibrated using the state-of-the-art method by Bok *et al.* [43].



(a) Iteration 0: $d_\mathrm{e} \approx 0.6\,\mathrm{m}$.

(b) Iteration 100: $d_\mathrm{e} \approx 80\,\mu\mathrm{m}$.

Fig. 2: Convergence of the calibration: Depict are the observed area of the monitor, the calibrated camera rays and the mean Euclidean distances at start and end. Even with an initially very bad pose estimation, the procedure converges towards reasonable results. Note the difference in scale.

Both methods use checker patterns. The proposed generalized calibration procedure was initialized with a rough pose estimation. The webcam was initialized with OpenCV and for the light field camera we used the calibration by Bok *et al.* with a succeeding standard pose estimation. As an alternative, one could also initialize using the generalized relative pose estimation algorithm proposed by Ramalingam *et al.* [10]. In many cases, we observed that it was acceptable to just "guess" the positions of the monitor, *e.g.* see Fig. 2, where the monitor poses at start and after convergence are shown. We eventually terminated the alternating minimization after 100 iterations, due to sufficient convergence. Since each ray is independent from one another, it is possible to process them in parallel, using a GPU. The optimization of 40 million pixels (Lytro Illum) thus only takes a few seconds per iteration (Intel Core i7-6700, Nvidia GTX 1080 Ti, 16GB RAM). The calibration procedure therefore converges after 10 minutes, whereas the method by Bok *et al.* uses a highly advanced model which takes more than three hours to calibrate. Our method is even faster when calibrating the two megapixel webcam.

Fig. 3 shows the convergence of the proposed method as a function of the root mean square error (RMSE) over the number of iterations. The plot shows the calibration of the webcam, which was initialized with the model-based OpenCV pose estimation. To investigate the robustness against bad initialization, the convergence behavior was investigated for 100 trials while random translations in the range $\pm 10\,\mathrm{cm}$ and rotations $\pm 10^\circ$ were added to each initialization. Therefore, the start of the plot corresponds approximately to the error of the OpenCV calibration. For comparison, the convergence behavior of the generalized calibration method of Bergamasco *et al.* [15] was also investigated, using the same initializations. Fig. 3 shows that the proposed method converges significantly faster than their method and that it is less sensitive to a bad initialization, which is shown by the smaller standard deviation in the RMSE.

Theoretical we need two different point observations to fit a ray, then $\mathbf{A}_\mathrm{dd}$ is positive definite. And to fit a pose, we need three non-parallel rays (then $\mathbf{A}_\mathrm{tt}$ is positive definite) that observe different points (then $\mathbf{A}_\mathrm{rr}$ is positive definite). With only two reference

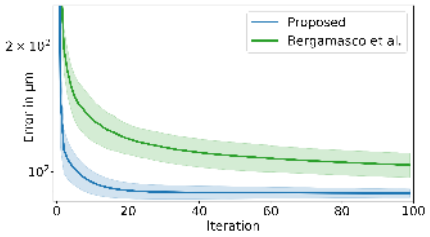


Fig. 3: Convergence and initialization of AM-calibration: The plot shows the mean value and the $\pm\sigma$-range of the convergence.

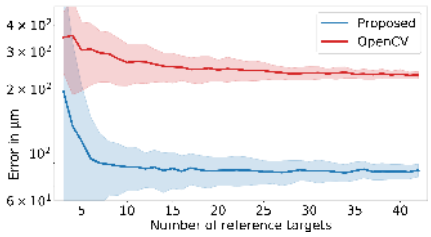


Fig. 4: Dependency on the number of patterns: The plot shows the mean value and the $\pm\sigma$-range of the error.

targets, we always converge to a perfect fit, which of course is useless. An unambiguous solution however can theoretically be obtained with at least three reference poses [10]. But of course, because we use a least squares based minimization approach and we want to reduce the impact of noise, we need more reference targets. This becomes apparent in Fig. 4. Here, we investigate the calibration error when different numbers of reference targets are used. For this purpose, the camera was calibrated 100 times, where each time a fixed number of target patterns was randomly selected from a total set of 60 poses. The mean error of all calibrations and their $\pm\sigma$ standard deviation are plotted over the number of used patterns. It can be seen that the overall calibration error needs at least a minimum of 15-20 poses to result in a good calibration, whereas more poses increase the overall robustness of the method. Too few patterns, on the other hand, result in a very unreliable calibration. We see similar results for the OpenCV calibration, although the dependency on the number of patterns is not as strong as compared to the proposed method. In summary, the proposed calibration needs more reference poses to correctly estimate the immense number of parameters. However, even with fewer poses, the error of the proposed calibration is several times smaller than the model-based calibration.

To ensure a fair comparison between the calibration methods, we examined the different models with regard to their point-to-ray distance of each ray to every observed feature in every monitor plane. The comparison is here done using normalized values, which results in a weighted distance $d_\mathrm{w} = (\sum w_{ik})^{-1} \sum w_{ik} \|\mathbf{p}_{ik} \times \mathbf{d}_i - \mathbf{m}_i\|$. For a demonstration of the benefit of using additional uncertainty information, the Euclidean distances d_e were evaluated, too. A comparison of the commonly used re-projection error is not possible, because in a generalized camera model there isn't anything like an "image plane", just a set of rays. Table 1 summarizes the respective calibration results.

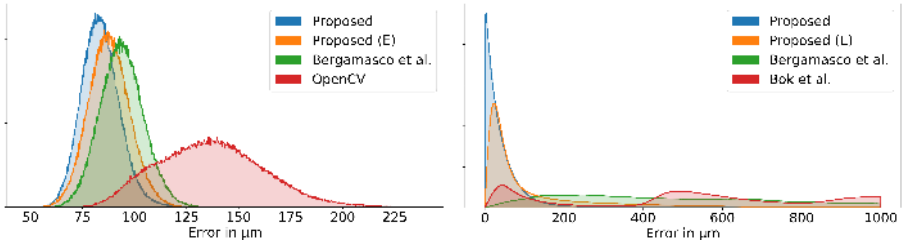
We can see that the proposed method produces the best results. Even for the webcam, with its relatively simple optics, the generalized approach delivers both a smaller mean error and RMSE, resulting in a more precise geometric calibration with fewer outliers at the same time. In the classic model, these outliers cannot be used because they are too far away from the model description. The generalized model, however, can effectively use each individual pixel as a source of information. This becomes particularly visible for the webcam if only the error regarding the checkerboard features is evaluated. Here, the error is smaller than when all phase shift features are used for every pixel. This demonstrates that the classic calibrations optimize the camera model for only a part of the pixels, by

Table 1: Calibration errors: Methods with the suffix "checker" were evaluated only at the sparsely detected checker features. For all others, the error was calculated over all observed phase-shift features. "Proposed (E)" does not use the uncertainty and only minimizes the Euclidean distance.

	Logitech Webcam				Lytro Illum			
	d_w in µm		d_e in µm		d_w in µm		d_e in µm	
	Mean	RMSE	Mean	RMSE	Mean	RMSE	Mean	RMSE
OpenCV [39] (checker)	–	–	267.4	340.0	–	–	–	–
OpenCV	243.1	323.8	343.3	375.3	–	–	–	–
Bok *et al.* [43] (checker)	–	–	–	–	–	–	448.4	851.8
Bok *et al.*	–	–	–	–	375.6	758.8	7165.1	8686.0
Bergamasco *et al.* [15]	88.8	121.0	93.9	130.3	922.1	1696.5	1041.8	1720.5
Proposed (E)	83.3	117.6	86.8	**124.3**	77.1	185.3	163.1	**438.5**
Proposed	**81.6**	**117.0**	**84.6**	125.5	**49.4**	**105.8**	**155.5**	457.4

neglecting outliers, while the generalized model optimally calibrates every pixel. Also, our proposed method performs better than the generalized approach by Bergamasco *et al.*, even if we don't take the uncertainties into account and only minimize the Euclidean distance. Furthermore, it can be seen that additional information about the coordinate uncertainty improves the calibration even more. Inaccurate points are weighted less strongly and therefore have a weaker effect on the result. Fig. 5a illustrates the results by showing the distribution of all point-to-ray distances.

Similar conclusions can be drawn with the Lytro Illum camera. Due to the more complex optics and the more extensive optimization associated with it, the differences here are much greater and the superiority of the proposed generalized calibration becomes even clearer. Although the model by Bok *et al.* is highly advanced, it is strongly adapted to the few checker features and only produces good results here. But if the model is evaluated using the phase shift features for every pixel, then this leads to high RMSE values caused by many outliers. In this case, we can see particularly well that a low dimensional model-based approach cannot ideally describe every pixel of a camera with complex optics, such as the light field camera. Moreover, the benefit of using uncertainties becomes very well apparent: the quality of pixels in microlens-based light field cameras (and the ability to accurately model the corresponding rays) deteriorates towards the edges of the microlenses [43, 44], leading to increased uncertainties. These can however be effectively suppressed by the proposed procedure, leading to much smaller mean errors and RMSE values. The method by Bok *et al.* can calibrate the center of each microlens very well. Here, their calibration error reduces to about $60\,\mu\mathrm{m}$ for the best pixels. But, the more the pixels move away from the center, the worse becomes the error. This reduces the overall calibration quality, as seen in the results. Also, the method by Bok *et al.* returns only 35 million of the total 41 million pixels. The worst pixels, which are between neighboring microlenses, cannot be modeled and are therefore cut off. Thus, they cannot be analyzed in the evaluation made here. However, our proposed model can effectively calibrate the rays of every pixel of the sensor, whereby we not only get good calibration results in the centers of the microlenses, but also at the edges, where it is very difficult to describe the light field camera with a uniform model. While



(a) Histogram of errors (Logitech Webcam). (b) Histogram of errors (Lytro Illum).

Fig. 5: The generalized model creates a much tighter distribution with less outliers as compared to the classical calibrations. Outliers can be suppressed even more with uncertainty information.

the method by Bergamasco *et al.* delivers good results for the webcam, we weren't able to get a good calibration for the light field camera. Although the calibration of the webcam shows that their approach works, it seems that it does not generalize as well as the proposed method and that it has difficulties with the poor quality of the pixels at the edges of the microlenses. The procedure diverged in our experiments. Only after improving the initialization for a few iterations using our method and by excluding the pixels with highest uncertainty, we were able to obtain a convergent result for their method, which still has a smaller error than the calibration by Bok *et al.*

Fig. 5b summarizes these results and shows the distribution of all point-to-ray distances for the Lytro Illum. The method by Bok *et al.* results in a flat distribution with a peak at $60\,\mu\mathrm{m}$. Also, several peaks systematically appear at higher distances, which are due to the difficulties of modeling a light field camera. The method by Bergamasco *et al.* results in an overall flat distribution with errors at high values (more than $1\,\mathrm{mm}$). Our proposed methods, on the other hand, are much tighter with peaks at far lower values. Moreover, larger errors from minimizing only the Euclidean distance can be shifted to smaller ones by using the generalized calibration with uncertainty-based weighting.

5 Summary

We presented a new calibration technique for the generalized imaging model. We proposed to split the calibration into two parts, ray calibration and pose estimation, and to apply an alternating minimization to efficiently optimize the immense number of parameters. Dense calibration features were obtained using phase shifting techniques, where we accounted for the measurement uncertainty that were estimated during the pre-processing. We presented a simple analytical solution to minimize the ray subpoblem. And further, we optimized the pose by decoupling rotation and translation and by using a gradient descent on the rotation manifold. Finally, experimental evaluation verified the advantages of the proposed method over conventional and other generalized approaches and demonstrated the benefit of using additional information about the uncertainty of the calibration target coordinates. We make the source code of our method publicly available and encourage others to use it to their own needs [45].

References

1. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1330–1334
2. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision second edition. Cambridge University Press (2000)
3. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Computer Science Technical Report CSTR **2** (2005) 1–11
4. Pless, R.: Using many cameras as one. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Los Alamitos, Calif, IEEE Computer Society (2003) II–587–93
5. Swaminathan, R., Kang, S.B., Szeliski, R., Criminisi, A., Nayar, S.K.: On the motion and appearance of specularities in image sequences. In Heyden, A., Johansen, P., Nielsen, M., Sparr, G., eds.: Computer Vision - ECCV 2002. Volume 2350 of Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg (2006) 508–523
6. Grossberg, M.D., Nayar, S.K.: A general imaging model and a method for finding its parameters. In: Proceedings / Eighth IEEE International Conference on Computer Vision, Los Alamitos, Calif., IEEE Computer Society (2001) 108–115
7. Grossberg, M.D., Nayar, S.K.: The raxel imaging model and ray-based calibration. International Journal of Computer Vision **61** (2005) 119–137
8. Sturm, P., Ramalingam, S.: A generic concept for camera calibration. In Pajdla, T., Matas, J., eds.: Computer Vision - ECCV 2004. Lecture Notes in Computer Science. Springer, Berlin and Heidelberg (2004) 1–13
9. S. Ramalingam, P. Sturm: A unifying model for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 1309–1319
10. Ramalingam, S., Sturm, P., Lodha, S.K.: Towards complete generic camera calibration. In Schmid, C., Tomasi, C., Soatto, S., eds.: CVPR 2005, Los Alamitos, Calif, IEEE Computer Society (2005) 1093–1098
11. Bothe, T., Li, W., Schulte, M., Kopylow, C.v., Bergmann, R.B., Jüptner, W.P.O.: Vision ray calibration for the quantitative geometric description of general imaging and projection optics in metrology. Applied Optics **49** (2010) 5851–5860
12. Miraldo, P., Araujo, H., Queiro, J.: Point-based calibration using a parametric representation of the general imaging model. In: IEEE International Conference on Computer Vision (ICCV), 2011, Piscataway, NJ, IEEE (2011) 2304–2311
13. Miraldo, P., Araujo, H.: Calibration of smooth camera models. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 2091–2103
14. Schops, T., Larsson, V., Pollefeys, M., Sattler, T.: Why having 10,000 parameters in your camera model is better than twelve. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
15. Bergamasco, F., Albarelli, A., Rodola, E., Torsello, A.: Can a fully unconstrained imaging model be applied effectively to central cameras? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, Piscataway, NJ, IEEE (2013) 1391–1398
16. Bergamasco, F., Albarelli, A., Cosmo, L., Torsello, A., Rodola, E., Cremers, D.: Adopting an unconstrained ray model in light-field cameras for 3d shape reconstruction. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Piscataway, NJ, IEEE (2015) 3003–3012
17. Chen, C.S., Chang, W.Y.: On pose recovery for generalized visual sensors. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 848–861
18. Balzer, J., Werling, S.: Principles of shape from specular reflection. Measurement **43** (2010) 1305–1317

19. Pankaj, D.S., Nidamanuri, R.R., Prasad, P.B.: 3-D imaging techniques and review of products. In: Proceedings of International Conference on Innovations in Computer Science and Engineering. (2013)
20. van der Jeught, S., Dirckx, J.J.: Real-time structured light profilometry: A review. Optics and Lasers in Engineering **87** (2016) 18–31
21. Shevlin, F.: Analysis of orientation problems using plücker lines. In: Proceedings, Silver Spring, MD and Los Angeles, CA, IEEE Computer Society Press (1984) 685–689
22. v. d. Hodge, W., Pedoe, D.: Methods of Algebraic Geometry. Cambridge University Press, Cambridge (1994)
23. Brox, T., Rosenhahn, B., Gall, J., Cremers, D.: Combined region and motion-based 3D tracking of rigid and articulated objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 402–415
24. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear gauss–seidel method under convex constraints. Operations Research Letters **26** (2000) 127–136
25. Zuo, C., Feng, S., Huang, L., Tao, T., Yin, W., Chen, Q.: Phase shifting algorithms for fringe projection profilometry: A review. Optics and Lasers in Engineering **109** (2018) 23–59
26. Fischer, M., Petz, M., Tutsch, R.: Model-based noise prediction for fringe projection systems - A tool for the statistical analysis of evaluation algorithms. TM-Technisches Messen **84** (2017) 111–122
27. Zuo, C., Huang, L., Zhang, M., Chen, Q., Asundi, A.: Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review. Optics and Lasers in Engineering **85** (2016) 84–103
28. Salvi, J., Pagès, J., Batlle, J.: Pattern codification strategies in structured light systems. Pattern Recognition **37** (2004) 827–849
29. Kanatani, K.: Analysis of 3-D rotation fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 543–549
30. Schweighofer, G., Pinz, A.: Globally optimal O(n) solution to the PnP problem for general camera models. In: BMVC. (2008)
31. Ventura, J., Arth, C., Reitmayr, G., Schmalstieg, D.: A minimal solution to the generalized pose-and-scale problem. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2014) 422–429
32. Kneip, L., Li, H., Seo, Y.: UPnP: An optimal O(n) solution to the absolute pose problem with universal applicability. In Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., eds.: Computer Vision – ECCV 2014, Cham, Springer International Publishing (2014) 127–142
33. Kukelova, Z., Bujnak, M., Pajdla, T.: Automatic generator of minimal problem solvers. In Forsyth, D., Torr, P., Zisserman, A., eds.: Computer Vision – ECCV 2008, Berlin, Heidelberg, Springer Berlin Heidelberg (2008) 302–315
34. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision: From Images to Geometric Models. SpringerVerlag (2003)
35. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2009)
36. Boumal, N.: Optimization and estimation on manifolds. PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium (2014)
37. Niesen, U., Shah, D., Wornell, G.: Adaptive alternating minimization algorithms. In: IEEE International Symposium on Information Theory, 2007, Piscataway, NJ, IEEE Service Center (2007) 1641–1645
38. Nesterov, Y.E.: A method for solving the convex programming problem with convergence rate O(1/kˆ2). In: Dokl. akad. nauk Sssr. Volume 269. (1983) 543–547
39. Bradski, G.: The opencv library. Dr. Dobb's Journal of Software Tools (2000)
40. Ihrke, I., Restrepo, J., Mignard-Debise, L.: Principles of light field imaging: Briefly revisiting 25 years of research. IEEE Signal Processing Magazine **33** (2016) 59–69

41. Zhang, Q., Zhang, C., Ling, J., Wang, Q., Yu, J.: A generic multi-projection-center model and calibration method for light field cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
42. Zhang, Q., Ling, J., Liu, Y., Yu, J.: Ray-space projection model for light field camera. CVPR 2019 (2019)
43. Bok, Y., Jeon, H.G., Kweon, I.S.: Geometric calibration of micro-lens-based light field cameras using line features. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 287–300
44. D. G. Dansereau, O. Pizarro, S. B. Williams: Decoding, calibration and rectification for lenselet-based plenoptic cameras. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. (2013) 1027–1034
45. Institute of Industrial Information Technology, Karlsruhe Institute of Technology: Public GitLab repositories. `https://gitlab.com/iiit-public`, GNU GPLv3 License (2020)