Stable radial distortion calibration by polynomial matrix inequalities programming

Jan Heller[1], Didier Henrion[2,3,1], Tomáš Pajdla[1]

Abstract

Polynomial and rational functions are the number one choice when it comes to modeling of radial distortion of lenses. However, several extrapolation and numerical issues may arise while using these functions that have not been covered by the literature much so far. In this paper, we identify these problems and show how to deal with them by enforcing nonnegativity of certain polynomials. Further, we show how to model these nonnegativities using polynomial matrix inequalities (PMI) and how to estimate the radial distortion parameters subject to PMI constraints using semidefinite programming (SDP). Finally, we suggest several approaches on how to incorporate the proposed method into the overall camera calibration procedure.

1 Introduction

Radial distortion modeling is the most important non-linear part of the camera calibration process [9]. The first works on the topic came from the photogrammetric community [4, 5, 14]. Since then, a plethora of models has been suggested in the literature [16]. Among the proposed models, the ones based on polynomial and rational functions are the most popular. This popularity undoubtedly stems from the fact that these function are easily manipulated and yet provide sufficient fitting power for wide range or distortions. Unfortunately, the extrapolation qualities of polynomials can be quite unpredictable in situations where little or no data is available. However, even if data points are missing, the overall shape of the distortion is known *a priori* in many calibration scenarios, *e.g.*, the lens introduces barrel or pincushion distortions. Based on such *a priori* information, the shape of the polynomial and rational distortion functions can be controlled by enforcing nonnegativity of certain polynomials. For example, in the case of pincushion distortion we can accomplish the desired shape by enforcing nonnegativity of the first and the second derivatives of the distortion function on the whole field of view of the camera.

In this paper, we propose a radial distortion calibration procedure where a polynomial cost function, *e.g.*, reprojection error, is minimized subject to such shape constraints.

[1] Faculty of Electrical Engineering, Czech Technical University in Prague, CZ-16627 Praha 6, Technická 2, Czech Republic.

[2] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France.

[3] Université de Toulouse; F-31400 Toulouse; France.

This shape optimization procedure is designed to stabilize the shape of the distortion function. It is based on polynomial matrix inequalities (PMI) programming and can be easily incorporated into an existing camera calibration procedure.

In Section 2, we formally introduce the radial distortion function and present several extrapolation issues arising while using polynomial and rational distortion models. Next, in Section 3 we provide a minimal theoretical background needed for our shape stabilization approach. In Section 4, we demonstrate the proposed method on three types of radial distortion shapes and models and show how to incorporate the method into an overall camera calibration procedure. Finally, in Section 5 we experimentally validate our approach and show that the method guarantees the correct shape of a distortion function without compromising the quality of the overall camera calibration as measured by the reprojection error.

2 Camera Radial Distortion

Let us suppose that a set of scene points $\mathbf{X}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$ is observed by a camera. If $\mathtt{R} \in SO(3)$, $\mathbf{t} \in \mathbb{R}^3$ are the camera extrinsic parameters, a scene point $\mathbf{X}_i$ gets projected into an image point $(x_i, y_i, 1)^\top$:

$$\lambda_i (x_i, y_i, 1)^\top = \mathtt{R}\mathbf{X}_i + \mathbf{t},\ \lambda_i \in \mathbb{R}.$$

In reality, some amount of radial distortion is always present and the camera observes a point $(\hat{x}_i, \hat{y}_i, 1)^\top$ which does not coincide with the ideal (and unobservable) point $(x_i, y_i, 1)^\top$. In pixel coordinates, the camera observes a point $\mathtt{K}(\hat{x}_i, \hat{y}_i, 1)^\top$, where $\mathtt{K} \in \mathbb{R}^{3\times 3}$ is the matrix of intrinsic camera parameters, the so-called calibration matrix. *Radial distortion function* $L\colon \mathbb{R} \to \mathbb{R}$ is a function of radius $r = \sqrt{x_i^2 + y_i^2}$ that models the radial displacement of the ideal image point position from the center of the radial distortion as

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix} = L(r) \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \tag{1}$$

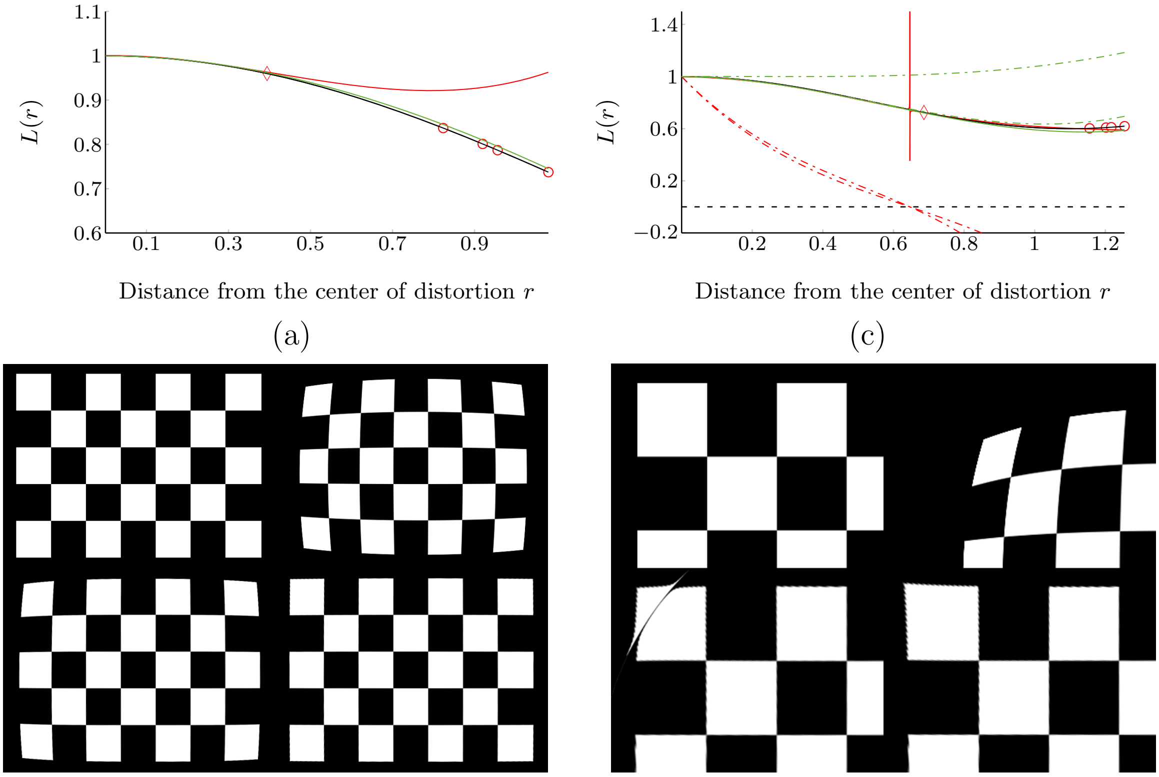
The function $L(r)$ is only defined for $r > 0$ and $L(0) = 1$, $L(r) > 0$. For the purposes of demonstration of the proposed shape optimization procedure, we will use $L(r)$ defined as follows

$$L(r) = \frac{f(r)}{g(r)} = \frac{1 + k_1 r + k_2 r^2 + k_3 r^3}{1 + k_4 r + k_5 k^2 + k_6 r^3}, \tag{2}$$

where $\mathbf{k} = (k_1, k_2, \ldots, k_6)$ is the vector of model parameters. This definition accommodates several models already proposed in the literature [12]. However, we will see that the shape optimization procedure holds for any rational function.

2.1 Extrapolation issues of radial distortion calibration

Let us motivate the need for the radial distortion shape optimization by demonstrating two examples of extrapolation issues arising while using polynomial and rational distortion models.



(a) (c)

(b) (d)

Figure 1: *Calibration issues.* Examples of issues arising while using polynomial and rational function for radial distortion calibration. See text for details.

First, let's suppose a calibration scenario where images of a calibration target were taken, but the image projections of the known 3D points lie close to the center of the images with no points covering the corners of the images. Figure 1(a) shows in black the graph of the amount of barrel distortion introduced by the camera lens as a function of the distance from the center of the radial distortion. When a polynomial distortion model $L(r) = f(r)$ is used, see Equation 2, in combination with an unconstrained calibration method [21, 2] (in red), the real distortion is fitted successfully near the center of the image on intervals where the data points are available (left of the diamond symbol). However, the recovered polynomial quickly drifts away elsewhere (red circles depict the distances of the projections of the image corners). In green, a polynomial recovered by the method proposed in this paper is shown. Here, the negativity of the first and the second derivatives of the polynomial on the whole field of view was enforced. This caused the model to fit the original distortion much closer on the whole field of view. Figure 1(b) shows a synthetic checkerboard image (the upper left corner) and the same image distorted by the original barrel distortion (the upper right corner). In the lower left corner, the image is undistorted back using the polynomial recovered by [21, 2]. In the lower right corner, the image successfully undistorted by the polynomial recovered using the proposed shape optimization method is shown.

Let us consider a similar calibration scenario to the one from the previous paragraph, this time with a lens causing a mustache type radial distortion, see Figure 1(c). If the radial distortion model is used, $L(r) = \frac{f(r)}{g(r)}$, the classical calibration approach [21, 2] is able to correctly recover the original shape. However, the polynomials $f(r)$ and $g(r)$ share a common root (red dash-dot lines), which causes a numerical instability presented as a sharp spike in $L(r)$ around the common root—an issue we will call the *zero-crossing problem*. When the nonnegativity of $g(r)$ is enforced using the proposed approach, not only is the correct shape recovered, but since there is now no root in the field of view interval (green dash-dot lines), the spike in $L(r)$ is also gone. Figure 1(d) shows a similar arrangement as Figure 1(b), now with only the upper left part of the checkerboard shown. The numerical instability of $L(r)$ is presented as a notable ringing in the upper left corner of the checkerboard. One can argue that the common root is a consequence of the fact that the degrees of $f(r), g(r)$ are higher that needed and that a model with fewer coefficients should be used. This may be true in some cases, however, we observed just as many situations where the lower degree polynomials resolved the zero-crossing problem only at the cost of a considerably higher reprojection error.

3 Polynomials and PMI Programming

In this section, we present a minimal theoretical background needed for the proposed shape optimization procedure.

3.1 Polynomials and polynomial matrices

An univariate polynomial $p(x) \in \mathbb{R}_n[x]$ of degree $n \in \mathbb{N}$ is a real function defined as

$$p(x) = p_n x^n + p_{n-1}x^{n-1} + \cdots + p_1 x + p_0 = \mathbf{p}^\top \boldsymbol{\psi}_n(x),$$

where $\mathbf{p} = (p_0, p_1, \ldots, p_n)^\top \in \mathbb{R}^{n+1}$ is the vector of coefficients with a nonvanishing coefficient p_n and $\boldsymbol{\psi}_n(x) = (1, x, x^2, \ldots, x^n)^\top$ is the canonical basis. Let $q(x) \in \mathbb{R}_{2n}[x]$. A symmetric matrix $\mathtt{Q} \in \mathbb{R}^{n' \times n'}$, $\mathtt{Q} = (q_{i,j})$, where $n' = n + 1$, is called *Gram matrix* associated with $q(x)$ and the basis $\boldsymbol{\psi}_n(x)$ [6] if

$$q(x) = \boldsymbol{\psi}_n^\top(x)\,\mathtt{Q}\,\boldsymbol{\psi}_n(x). \tag{3}$$

Generally, there is more than one Gram matrix associated with a polynomial $q(x)$ and we will denote the set of such matrices as $\mathcal{G}(q(x))$.The polynomial $q(x)$ can be expressed in the elements of $\mathtt{Q}$ by simply expanding the right hand side of Equation 3 and by comparing the coefficients.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ be a real vector and $\boldsymbol{\alpha} = (\alpha_1, \alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ an integer vector. A *monomial* of degree $n = \sum \alpha_i$ is defined as $\mathbf{x}^{\boldsymbol{\alpha}} = \prod_{i=1}^n x_i^{\alpha_i}$. A multivariate polynomial $p(\mathbf{x}) \in \mathbb{R}_n[\mathbf{x}]$ of degree $n \in \mathbb{N}$ is a mapping from $\mathbb{R}^d$ to $\mathbb{R}$ defined as a linear combination of monomials up to degree n,

$$p(\mathbf{x}) = \sum_{|\boldsymbol{\alpha}| \le n} p_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} = \sum_{|\boldsymbol{\alpha}| \le n} p_{\boldsymbol{\alpha}} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d} = (p_{\boldsymbol{\alpha}})_{|\boldsymbol{\alpha}| \le n}^\top (\mathbf{x}_{\boldsymbol{\alpha}})_{|\boldsymbol{\alpha}| \le d} = \mathbf{p}^\top \boldsymbol{\psi}_n(\mathbf{x}),$$

where $\mathbf{p} \in \mathbb{R}^m$ is the vector of coefficients and $\boldsymbol{\psi}_n(\mathbf{x})$ is the canonical basis of $m = \binom{d+n}{d}$ monomials up to degree n. By a polynomial matrix we will understand a symmetric matrix whose elements are polynomials. In the next, $\mathbb{S}^n(\mathbb{R}[\mathbf{x}])$ will denote the set of $n \times n$ symmetric polynomial matrices. The degree of $\mathtt{P} = (p_{i,j}(\mathbf{x})) \in \mathbb{S}^n(\mathbb{R}[\mathbf{x}])$ is the largest degree of all the polynomial elements of $\mathtt{P}$, $\deg \mathtt{P} = \max_{i,j} \deg p_{i,j}(\mathbf{x})$.

Besides parameterizing polynomials by the associated Gram matrices, we will also need to "linearize" them, *i.e.*, to substitute every monomial $\mathbf{x}^{\boldsymbol{\alpha}}$ by a new variable $y_{\boldsymbol{\alpha}} \in \mathbb{R}$. To do this, we define the *Riesz functional* $\ell_{\mathbf{y}} \colon \mathbb{R}_n[\mathbf{x}] \to \mathbb{R}[\mathbf{y}]$, a linear functional that for a d-variate polynomial of degree n, $p(\mathbf{x}) = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$, returns an m-variate polynomial of degree one, $\ell_{\mathbf{y}}(p(\mathbf{x})) = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}}$, $m = \binom{d+n}{d}$. With a slight abuse of notation, we will also use $\ell_{\mathbf{y}}$ as a matrix operator acting on $\mathbb{S}^n(\mathbb{R}[\mathbf{x}])$: if $\mathtt{P} \in \mathbb{S}^n(\mathbb{R}[\mathbf{x}])$, then $\mathtt{P}' = \ell_{\mathbf{y}}(\mathtt{P})$ if and only if $p'_{i,j}(\mathbf{y}) = \ell_{\mathbf{y}}(p_{i,j}(\mathbf{x}))$.

3.2 Polynomials positive on finite intervals

The shape optimization procedure presented in this paper is based on enforcing nonnegativity of certain polynomials. Since most of the real cameras have limited fields of view, we only need to control the behavior of $L(r)$ for values $r \in [0, \bar{r}]$, where $\bar{r}$ is the maximal distance between the center of the radial distortion and an (undistorted) image point. For this, we need to characterize the set of univariate polynomials nonnegative on finite intervals. In [13], based on Markov-Lukacs theorem, Nesterov showed how to characterize such a set using positive semidefinite Gram matrices:

Theorem 1 *Let $\alpha < \beta$, $p(x) \in \mathbb{R}[x]$ and $\deg p(x) = 2n$. Then $p(x) \geq 0$ for all $x \in [\alpha, \beta]$ if and only if*

$$p(x) = s(x) + (x - \alpha)(\beta - x)t(x),$$

where $s(x) = \boldsymbol{\psi}_n^\top(x)\,\mathtt{S}\,\boldsymbol{\psi}_n(x)$, $t(x) = \boldsymbol{\psi}_{n-1}^\top(x)\,\mathtt{T}\,\boldsymbol{\psi}_{n-1}(x)$, such that $\mathtt{S}, \mathtt{T} \succeq 0$ (i.e., $\mathtt{S} \in \mathcal{G}(s(x))$, $\mathtt{T} \in \mathcal{G}(t(x))$ are positive semidefinite Gram matrices of polynomials $s(x)$ and $t(x)$, respectively).

If $\deg p(x) = 2n + 1$, then $p(x) \geq 0$ for all $x \in [\alpha, \beta]$ if and only if

$$p(x) = (x - \alpha)s(x) + (\beta - x)t(x),$$

where $s(x) = \boldsymbol{\psi}_n^\top(x)\,\mathtt{S}\,\boldsymbol{\psi}_n(x)$, $t(x) = \boldsymbol{\psi}_n^\top(x)\,\mathtt{T}\,\boldsymbol{\psi}_n(x)$, such that $\mathtt{S}, \mathtt{T} \succeq 0$.

Even though Theorem 1 is an equivalence, we will only use it as an implication: as long as we will have matrices $\mathtt{S}, \mathtt{T}$ that are positive semidefinitive, Theorem 1 guarantees that a polynomial $p(x)$ constructed using these matrices will be nonnegative on a given interval.

3.3 Polynomial Matrix Inequalities

According to Theorem 1, a polynomial is nonnegative on an interval as long the matrices $\mathtt{S}, \mathtt{T}$ are positive semidefinite. By combining these constraints with a polynomial cost function, we get a problem of polynomial matrix inequalities (PMI) programming. A PMI program can be formally defined as follows:

Problem 1 (Polynomial matrix inequalities program)

$$\begin{array}{rl} \textit{minimize} & p(\mathbf{x}) \\ \textit{subject to} & \mathtt{G}_i(\mathbf{x}) \succeq 0,\ i = 1, \ldots, m, \\ \textit{where} & p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}], \mathtt{G}_i \in \mathbb{S}^{n_i}(\mathbb{R}[\mathbf{x}]). \end{array}$$

In general, Problem 1 is a hard non-convex problem. Note however, that if the cost function $p(\mathbf{x})$ and the matrices $\mathtt{G}_i(\mathbf{x})$, $i = 1, \ldots, m$ have degree one, then Problem 1 reduces to a linear matrix inequality (LMI) program and as such is a semidefinite program (SDP) solvable by any available SDP solver. In fact, most of the time the shape optimization problems in this paper lead to such a program.

Sometimes still, $\mathtt{G}_i(\mathbf{x})$ will not be linear. In such cases, we will use the relaxation approach suggested by Henrion and Lasserre [10]. In [10], the authors proposed a hierarchy of LMI programs $\mathcal{P}_1, \mathcal{P}_2, \ldots$ that produces a monotonically non-decreasing sequence of lower bounds $p(\mathbf{x}_1^*) \leq p(\mathbf{x}_2^*) \leq \ldots$ on Problem 1 that converges to the global minimum $p(\mathbf{x}^*)$. Practically, the series converges to $p(\mathbf{x}^*)$ in finitely many steps, *i.e.*, there exists $j \in \mathbb{N}$, such that $p(\mathbf{x}_j^*) = p(\mathbf{x}^*)$. The authors also showed how this situation can be detected and how the value of $\mathbf{x}^*$ can be extracted from the solution of the relaxation by the tools of linear algebra.

Let us show here how to construct $\mathcal{P}_\delta$, *i.e.*, the LMI relaxation of Problem 1 of order δ; see [10] for the technical justification of this procedure. Let $\mathtt{G} \in \mathbb{S}^n(\mathbb{R}[\mathbf{x}])$, $n = \sum_{i=1}^m n_i$ denote a block diagonal matrix with matrices $\mathtt{G}_i$ on it's diagonal. Since $(\forall i\colon \mathtt{G}_i(\mathbf{x}) \succeq 0) \Leftrightarrow \mathtt{G}(\mathbf{x}) \succeq 0$, we can replace the PMI constraints $\mathtt{G}_i(\mathbf{x}) \succeq 0$ with one PMI constraint $\mathtt{G}(\mathbf{x}) \succeq 0$. Next, we construct the so-called *moment matrix* $\mathtt{M}_\delta(\mathbf{y})$ and *localizing matrix* $\mathtt{M}_\delta(\mathtt{G}, \mathbf{y})$ *of* $\mathtt{G}$, defined as

$$\begin{array}{rcl} \mathtt{M}_\delta(\mathbf{y}) & = & \ell_\mathbf{y}(\boldsymbol{\psi}_\delta(\mathbf{x})\boldsymbol{\psi}_\delta^\top(\mathbf{x})), \\ \mathtt{M}_\delta(\mathtt{G}, \mathbf{y}) & = & \ell_\mathbf{y}((\boldsymbol{\psi}_\delta(\mathbf{x})\boldsymbol{\psi}_\delta^\top(\mathbf{x})) \otimes \mathtt{G}), \end{array}$$

where $\otimes$ denotes the Kronecker product [10]. Let $\gamma = 1$ if $\deg \mathtt{G} \leq 2$, $\gamma = \frac{\lceil \deg \mathtt{G} \rceil}{2}$ otherwise. Now, we can formally write the relaxation $\mathcal{P}_\delta$ as

Problem 2 (LMI relaxation $\mathcal{P}_\delta$ of order δ)

$$\begin{array}{rl} \textit{minimize} & \ell_\mathbf{y}(p(\mathbf{x})) \\ \textit{subject to} & \mathtt{M}_{\delta-\gamma}(\mathtt{G}, \mathbf{y}) \succeq 0, \\ & \mathtt{M}_\delta(\mathbf{y}) \succeq 0. \end{array}$$

As the Riesz functional $\ell_\mathbf{y}$ was used to "linearize" both the cost function and the constraints, we can easily see that Problem 2 is an LMI program.

4 Shape optimization for radial distortion calibration

In this section, we show how to combine the results presented in Section 3 into the radial distortion shape optimization procedure. Technically, the procedure consists of minimization of a polynomial cost function in the vector of radial distortion parameters $\mathbf{k}$ subject to PMI constraints enforcing nonnegativity of certain polynomials in the radius r. Such a minimization problem is a PMI program that can be dealt with using the approach from Section 3.3.

As mentioned in Section 3.2, we only need to control the shape of $L(r)$ on the interval $[0, \bar{r}]$. Note, that $\bar{r}$ is the maximal distance between the center of the radial distortion and *undistorted* image points, *i.e.*, the value of $\bar{r}$ is not known prior to the actual calibration. The value of $\bar{r}$ is therefore a user supplied parameter. Fortunately, the proposed method is not very sensitive to the value of this parameter and even a gross overestimate yields minima identical to the ground truth value.

4.1 Unconstrained radial distortion calibration

There are several ways how to determine the vector of parameters $\mathbf{k}$ of the distortion function $L(r)$ [9, 17]. All we need for our shape optimization approach is a polynomial cost function. Here, we will define and use one of such possible cost functions. Let us rewrite Equation 1 using $L(r)$ from Equation 2 as

$$g(r)\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix} - f(r)\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} g(r)\,\hat{x}_i - f(r)\,x_i \\ g(r)\,\hat{y}_i - f(r)\,y_i \end{pmatrix} = \mathbf{0}.$$

By factoring out the vector of parameters $\mathbf{k}$ and by denoting

$$\mathtt{A}_i = \begin{pmatrix} -r\,x_i & -r^2\,x_i & -r^3\,x_i & \hat{x}_i\,r & \hat{x}_i\,r^2 & \hat{x}_i\,r^3 \\ -r\,y_i & -r^2\,y_i & -r^3\,y_i & \hat{y}_i\,r & \hat{y}_i\,r^2 & \hat{y}_i\,r^3 \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} x_i - \hat{x}_i \\ y_i - \hat{y}_i \end{pmatrix},$$

we get a linear system $\mathtt{A}_i\mathbf{k} = \mathbf{b}_i$. Now, we can stack $\mathtt{A} = (\mathtt{A}_1^\top, \mathtt{A}_2^\top, \ldots, \mathtt{A}_n^\top)^\top$, $\mathbf{b} = (\mathbf{b}_1^\top, \mathbf{b}_2^\top, \ldots \mathbf{b}_n^\top)^\top$ and estimate the radial distortion parameters $\mathbf{k} = (k_1, k_2, \ldots, k_6)$ as a solution to an overdetermined system $\mathtt{A}\mathbf{k} = \mathbf{b}$ in the least square sense, *i.e.*, by minimizing $\|\mathtt{A}\mathbf{k} - \mathbf{b}\|^2$. Note that for polynomial model, *i.e.*, $g(x) = 1$, this corresponds to the minimization of the reprojection error.

Let us now express the minimization of $\|\mathtt{A}\mathbf{k} - \mathbf{b}\|^2$ as an LMI program. By expanding

$$\|\mathtt{A}_i\mathbf{k} - \mathbf{b}_i\|^2 = (\mathtt{A}_i\mathbf{k} - \mathbf{b}_i)^\top(\mathtt{A}_i\mathbf{k} - \mathbf{b}_i) = \mathbf{k}^\top\mathtt{A}_i^\top\mathtt{A}_i\mathbf{k} - 2\mathbf{b}_i^\top\mathtt{A}_i\mathbf{k} + \mathbf{b}_i^\top\mathbf{b}_i$$

and by denoting $\mathtt{M} = \sum_{i=1}^n \mathtt{A}_i^\top\mathtt{A}_i$, $\mathbf{m} = -2\sum_{i=1}^n \mathtt{A}_i^\top\mathbf{b}_i$, $c = \sum_{i=1}^n \mathbf{b}_i^\top\mathbf{b}_i$, we can write the polynomial form of the cost function as

$$\|\mathtt{A}\mathbf{k} - \mathbf{b}\|^2 = \mathbf{k}^\top\mathtt{M}\,\mathbf{k} + \mathbf{m}^\top\mathbf{k} + c. \tag{4}$$

As expected, Equation 4 is a quadratic polynomial in $\mathbf{k}$ and by construction $\mathtt{M} \succeq 0$, *i.e.*, $\mathtt{M}$ is a positive semidefinite matrix. Even though the cost function is quadratic, it can be converted into a linear function using the Schur complement trick [3]:

$$\mathtt{F} = \begin{pmatrix} \mathtt{I} & \mathtt{L}\mathbf{k} \\ \mathbf{k}^\top \mathtt{L}^\top & -\mathbf{m}^\top \mathbf{k} - c + \gamma \end{pmatrix} \succeq 0 \;\Leftrightarrow\; \mathbf{k}^\top \mathtt{L}^\top \mathtt{L}\mathbf{k} + \mathbf{m}^\top \mathbf{k} + c - \gamma \leq 0.$$

By decomposing $\mathtt{M}$ as $\mathtt{M} = \mathtt{L}^\top \mathtt{L}$, *e.g.*, using the Cholesky or the spectral decomposition [7] (recall that $\mathtt{M} \succeq 0$), we can rewrite the minimization of Equation 4 as the following LMI program:

Problem 3 (Unconstrained radial distortion calibration)

$$\begin{array}{ll} \textit{minimize} & \gamma \\ \textit{subject to} & \mathtt{F} = \begin{pmatrix} \mathtt{I} & \mathtt{L}\mathbf{k} \\ \mathbf{k}^\top \mathtt{L}^\top & -\mathbf{m}^\top \mathbf{k} - c + \gamma \end{pmatrix} \succeq 0. \end{array}$$

4.2 Barrel distortion and the polynomial model

As we can see from the example of barrel radial distortion in Figure 1(a), this type of distortion can be characterized by the negativity of the first and the second derivatives:

$$\forall r \in [0, \bar{r}]\colon L'(r) \leq 0 \,\&\, L''(r) \leq 0, \tag{5}$$

where $[0, \bar{r}]$ spans the field of view of the camera. If we consider the polynomial model $L(r) = f(r)$, the constraints above mean that we need to enforce nonnegativity of polynomials

$$-f'(r) = -k_1 - 2k_2\, r - 3k_3\, r^2, -f''(r) = -2k_2 - 6k_3\, r$$

on the interval $[0, \bar{r}]$. According to Theorem 1, $-f'(r) \geq 0$ for $\forall r \in [0, \bar{r}]$ iff

$$-f'(r) = -k_1 - 2k_2\, r - 3k_3\, r^2 = \boldsymbol{\psi}_1(r)^\top \mathtt{S}_1 \boldsymbol{\psi}_1(r) + r\,(\bar{r} - r)\,\mathtt{T}_1, \tag{6}$$

where

$$\mathtt{S}_1 = \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & s_{13} \end{pmatrix} \succeq 0,\; \mathtt{T}_1 = (t_{11}) \succeq 0.$$

By expanding the right hand side of Equation 6 and by comparing the polynomial coefficients, we get a parameterization of $\mathbf{k}$ in the elements of $\mathtt{S}_1$ and $\mathtt{T}_1$:

$$\left.\begin{array}{rcl} -k_1 & = & s_{11} \\ -2k_2 & = & 2s_{12} + \bar{r}\, t_{11} \\ -3k_3 & = & s_{13} - t_{11} \end{array}\right\} \Rightarrow \mathbf{k} = (-s_{11}, -s_{12} - \tfrac{1}{2}\bar{r} t_{11}, \tfrac{1}{3}(t_{11} - s_{13}), 0, 0, 0). \tag{7}$$

Let's apply Theorem 1 to $-f''(r)$ to get the following constraint:

$$-f''(r) = -2k_2 - 6k_3\, r = r\,\mathtt{S}_2 + (\bar{r} - r)\mathtt{T}_2, \mathtt{S}_2 = (s_{21}) \succeq 0, \mathtt{T}_2 = (t_{21}) \succeq 0. \tag{8}$$

By combining Equations 8 and 7, we can express the entries of $\mathtt{S}_2$ and $\mathtt{T}_2$ in the entries of $\mathtt{S}_1, \mathtt{T}_1$:

$$\left.\begin{array}{rcl} -2k_2 & = & \bar{r}\,t_{21} \\ -6k_3 & = & s_{21} - t_{21} \end{array}\right\} \Rightarrow \left\{\begin{array}{rcl} s_{21} & = & \frac{1}{\bar{r}}\,(2s_{12} + 2\bar{r}\,s_{13} - \bar{r}t_{11}) \\ t_{21} & = & \frac{2}{\bar{r}}\,(s_{12} + \frac{1}{2}\bar{r}t_{11}) \end{array}\right. \tag{9}$$

Now, we have four PMI constraints on the shape of $L(r)$. If we combine these constraints along with the parameterization of $\mathbf{k}$ from Equation 7 with Problem 3, we get a radial distortion calibration problem that enforces a barrel type distortion shape of the resulting distortion model:

Problem 4 (Barrel distortion calibration)

$$\begin{array}{ll} \textit{minimize} & \gamma \\ \textit{subject to} & \mathtt{F} \succeq 0,\ \mathtt{S}_1 \succeq 0,\ \mathtt{T}_1 = (t_{11}) \succeq 0, \\ & \mathtt{S}_2 = \left(\frac{1}{\bar{r}}\,(2s_{12} + 2\bar{r}\,s_{13} - \bar{r}t_{11})\right) \succeq 0, \\ & \mathtt{T}_2 = \left(\frac{2}{\bar{r}}\,(s_{12} + \frac{1}{2}\bar{r}t_{11})\right) \succeq 0. \end{array}$$

Problem 4 is a PMI program in 5 variables $\gamma, s_{11}, s_{12}, s_{13}, t_{11}$. Since both the cost function and the PMI constraints have degree one, Problem 4 is in fact an SDP problem. Once it is solved, the unknown distortion parameters $\mathbf{k}$ can be easily recovered using Equation 7.

4.3 Pincushion distortion and the division model

Let us make an analogous analysis for the pincushion distortion shape and the division model $L(r) = \frac{1}{g(r)}$. This type of distortion is characterized by the nonnegativity of the first and the second derivatives of $L(r)$ on the field of view of the camera $[0, \bar{r}]$. From the first derivative we get the following constraint on the polynomial denominator $g(r)$:

$$L'(r) = \frac{-g'(r)}{g^2(r)} \quad \Rightarrow \quad L'(r) \geq 0 \Leftrightarrow -g'(r) \geq 0.$$

The second derivative yields a bit more complicated constraint:

$$L''(r) = \frac{g(r)h(r)}{g^4(r)} = \frac{h(r)}{g^3(r)} \Rightarrow L''(r) \geq 0 \Leftrightarrow \left\{\begin{array}{r} (g(r) \geq 0 \,\&\, h(r) \geq 0) \vee \\ (g(r) \leq 0 \,\&\, h(r) \leq 0), \end{array}\right.$$

where $h(r) = 2(g'(r))^2 - g(r)g''(r)$. However, since we know that $L(r) > 0$ by definition, we only need to consider the constraints $g(r) \geq 0, h(r) \geq 0$. Let us start with the constraint $g(r) \geq 0$. According to Theorem 1, $g(r) \geq 0$ for $\forall r \in [0, \bar{r}]$ iff

$$g(r) = 1 + k_4 r + k_5 r^2 + k_6 r^3 = \boldsymbol{\psi}_1(r)^\top \mathtt{S}_1 \boldsymbol{\psi}_1(r) + (\bar{r} - r)\,\boldsymbol{\psi}_1(r)^\top \mathtt{T}_1 \boldsymbol{\psi}_1(r), \tag{10}$$

where

$$\mathtt{S}_1 = \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & s_{13} \end{pmatrix} \succeq 0,\ \mathtt{T}_1 = \begin{pmatrix} t_{11} & t_{12} \\ t_{12} & t_{13} \end{pmatrix} \succeq 0.$$

This leads to the following parameterization of $\mathbf{k}$ as well as to a constraint on the variable t_{11}:

$$\left.\begin{array}{rcl} 1 & = & \bar{r}\, t_{11} \\ k_4 & = & s_{11} - t_{11} + 2\bar{r}t_{12} \\ k_5 & = & 2s_{12} - 2t_{12} + \bar{r}t_{13} \\ k_6 & = & s_{13} - t_{13} \end{array}\right\} \Rightarrow \left\{\begin{array}{rcl} \mathbf{k} & = & (0,0,0,s_{11} - t_{11} + 2\bar{r}t_{12}, \\ & & \quad 2s_{12} - 2t_{12} + \bar{r}t_{13}, s_{13} - t_{13}) \\ t_{11} & = & \frac{1}{\bar{r}} \end{array}\right. \tag{11}$$

By applying Theorem 1 to the constraint $-g'(r) \geq 0$, we get

$$-g'(r) = -k_4 - 2k_5\, r - 3k_6\, r^2 = \boldsymbol{\psi}_1(r)^\top \mathtt{S}_2 \boldsymbol{\psi}_1(r) + r\,(\bar{r} - r)\,\mathtt{T}_2, \tag{12}$$

where

$$\mathtt{S}_2 = \begin{pmatrix} s_{21} & s_{22} \\ s_{22} & s_{23} \end{pmatrix} \succeq 0, \mathtt{T}_2 = (t_{21}) \succeq 0.$$

As in the case of the barrel distortion optimization, we can express the entries of $\mathtt{S}_2$ and $\mathtt{T}_2$ in the entries of $\mathtt{S}_1, \mathtt{T}_1$. This time, however, we have more variables than equations and we have to set one of the entries free—we chose s_{22}:

$$\left.\begin{array}{rcl} -3k_4 & = & s_{21} \\ -2k_5 & = & 2s_{22} + \bar{r}t_{21} \\ -3k_6 & = & s_{23} - t_{21} \end{array}\right\} \Rightarrow \left\{\begin{array}{rcl} s_{21} & = & t_{11} - s_{11} - 2\bar{r}t_{12} \\ s_{23} & = & -\frac{1}{\bar{r}}(s_{12} + 2s_{22} - 4t_{12} + \bar{r}(3s_{13} - t_{13})) \\ t_{21} & = & -\frac{1}{\bar{r}}(2s_{12} + s_{22} - 2t_{12} + \bar{r}t_{13}) \end{array}\right. \tag{13}$$

The final constraint is the most complicated because of the quadratic monomials in $\mathbf{k}$: $h(r) > 0$ for $\forall r \in [0, \bar{r}]$ iff

$$\begin{array}{rcl} h(r) & = & (6k_6r^2 + 4k_5r + 2k_4)(3k_6r^2 + 2k_5 + k_4) - \\ & & \quad -(2k_5 + 6k_6r)(k_6r^3 - k_5r^2 + k_4r + 1) \\ & = & \boldsymbol{\psi}_2(r)^\top \mathtt{S}_3 \boldsymbol{\psi}_2(r) + (\bar{r} - r)\,\boldsymbol{\psi}_1(r)^\top \mathtt{T}_3 \boldsymbol{\psi}_1(r), \end{array} \tag{14}$$

where

$$\mathtt{S}_3 = \begin{pmatrix} s_{31} & s_{32} & s_{33} \\ s_{32} & s_{34} & s_{35} \\ s_{33} & s_{35} & s_{36} \end{pmatrix} \succeq 0, \mathtt{T}_3 = \begin{pmatrix} t_{31} & t_{32} \\ t_{32} & t_{33} \end{pmatrix} \succeq 0.$$

Equation 14 gives us 5 constraints on 9 entries of $\mathtt{S}_3$ and $\mathtt{T}_3$. We chose to set free variables $s_{32}, s_{34}, s_{36}, t_{32}$; System 15 shows the form of the remaining 5 variables. Finally, we can combine these 6 PMI constraints, Problem 3 and the parameterization of $\mathbf{k}$ from Equation 11 into a radial distortion calibration problem that enforces a pincushion type distortion shape:

Problem 5 (Pincushion distorion calibration)

$$\begin{array}{ll} \textit{minimize} & \gamma \\ \textit{subject to} & \mathtt{F} \succeq 0,\ \mathtt{S}_1 \succeq 0,\ \mathtt{T}_1 \succeq 0, \mathtt{S}_2 \succeq 0,\ \mathtt{T}_2 \succeq 0, \mathtt{S}_3 \succeq 0,\ \mathtt{T}_3 \succeq 0. \end{array}$$

$$\left.\begin{array}{rcl} 12k_6^2 & = & s_{36}-t_{33} \\ 16k_5k_6 & = & 2s_{35}-2t_{32}+\bar{r}t_{33} \\ 6k_5^2+6k_4k_6 & = & 2s_{33}+s_{34}-t_{31}+2\bar{r}t_{32} \\ 6k_4k_5-6k_6 & = & 2s_{32}+\bar{r}t_{31} \\ 2k_4^2-2k_5 & = & s_{31} \end{array}\right\} \Rightarrow \left\{\begin{array}{rcl} s_{31} & = & 4t_{12}-s_{12}-2\bar{r}t_{31}+2(s_{11}-t_{11}+2\bar{r}t_{12})^2 \\ s_{33} & = & -\frac{1}{2\bar{r}}(6s_{13}+2s_{32}-6t_{13}+\bar{r}s_{34} \\ & & -6\bar{r}(2s_{12}-2t_{12}+\bar{r}t_{13})^2+2\bar{r}^2t_{32}- \\ & & -6(s_{11}-t_{11}+2\bar{r}t_{12}) \\ & & (2s_{12}-2t_{12}+\bar{r}t_{13}+\bar{r}s_{13}-\bar{r}t_{13})) \\ s_{35} & = & t_{32}-\frac{2}{\bar{r}}s_{36}+6\bar{r}(s_{13}-t_{13})^2+ \\ & & +8(s_{13}-t_{13})(2s_{12}-2t_{12}+\bar{r}t_{13}) \\ t_{31} & = & -\frac{2}{\bar{r}}(3s_{13}+s_{32}-3t_{13}- \\ & & 3(2s_{12}-2t_{12}+\bar{r}t_{13})(s_{11}-t_{11}+2\bar{r}t_{12})) \\ t_{33} & = & s_{36}-12(s_{13}-t_{13})^2 \end{array}\right. \tag{15}$$

Problem 5 is a PMI program in 11 variables γ, s_{11}, s_{12}, s_{13}, t_{12}, t_{13}, s_{22}, s_{32}, s_{34}, s_{36}, and t_{32}. Since $\mathtt{S}_3$ and $\mathtt{T}_3$ are polynomial matrices of degree 2, Problem 5 has to be dealt with using the relaxation scheme from Section 3.3.

4.4 Zero-crossing problem of the rational model

Also the zero-crossing problem of the rational model $L(r) = \frac{f(r)}{g(r)}$ can be dealt with using the proposed shape optimization technique. A sufficient condition for avoiding a common root of the polynomials $f(r)$ and $g(r)$ on the interval $[0, \bar{r}]$ is to force at least one on them to have no root. Here, we decided on enforcing the constraint

$$\forall r \in \langle 0, \bar{r}\rangle : g(r) - p \geq 0, \text{ where } p > 0. \tag{16}$$

Since Theorem 1 guarantees only nonnegativity of a polynomial, we need a strictly positive parameter p to enforce strict positivity of $g(r)$. Even though parameter p must be user supplied, the method is not overly sensitive to its value; in our experiments, we set $p = 0.1$. By applying Theorem 1 to the above constraint and the interval $[0, \bar{r}]$, we get

$$g(r) - p = 1 - p + k_4r + k_5r^2 + k_6r^3 = \boldsymbol{\psi}_1(r)^\top \mathtt{S}_1 \boldsymbol{\psi}_1(r) + (\bar{r} - r)\, \boldsymbol{\psi}_1(r)^\top \mathtt{T}_1 \boldsymbol{\psi}_1(r),$$

where

$$\mathtt{S}_1 = \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & s_{13} \end{pmatrix} \succeq 0,\ \mathtt{T}_1 = \begin{pmatrix} t_{11} & t_{12} \\ t_{12} & t_{13} \end{pmatrix} \succeq 0.$$

This yields a parameterization of $\mathbf{k}$ as well as a constraint on t_{11}:

$$\left.\begin{array}{rcl} 1-p & = & \bar{r}\,t_{11} \\ k_4 & = & s_{11} - t_{11} + 2\bar{r}t_{12} \\ k_5 & = & 2s_{12} - 2t_{12} + \bar{r}t_{13} \\ k_6 & = & s_{13} - t_{13} \end{array}\right\} \Rightarrow \left\{\begin{array}{rcl} \mathbf{k} & = & (k_1, k_2, k_3, s_{11} - t_{11} + 2\bar{r}t_{12}, \\ & & \quad 2s_{12} - 2t_{12} + \bar{r}t_{13}, s_{13} - t_{13}) \\ t_{11} & = & \frac{1-p}{\bar{r}} \end{array}\right. \tag{17}$$

Again, by combining the two PMI constraints with Problem 3 and the parameterization of $\mathbf{k}$ from Equation 17, we get a radial distortion calibration problem that eliminates the zero-crossing problem:

Problem 6 (Zero-crossing distortion calibration)

$$\begin{array}{ll} \textit{minimize} & \gamma \\ \textit{subject to} & \mathtt{F} \succeq 0,\ \mathtt{S}_1 \succeq 0,\ \mathtt{T}_1 \succeq 0. \end{array}$$

Problem 6 is an LMI program in 9 variables $\gamma, s_{11}, s_{12}, s_{13}, t_{12}, t_{13}, k_1, k_2, k_3$.

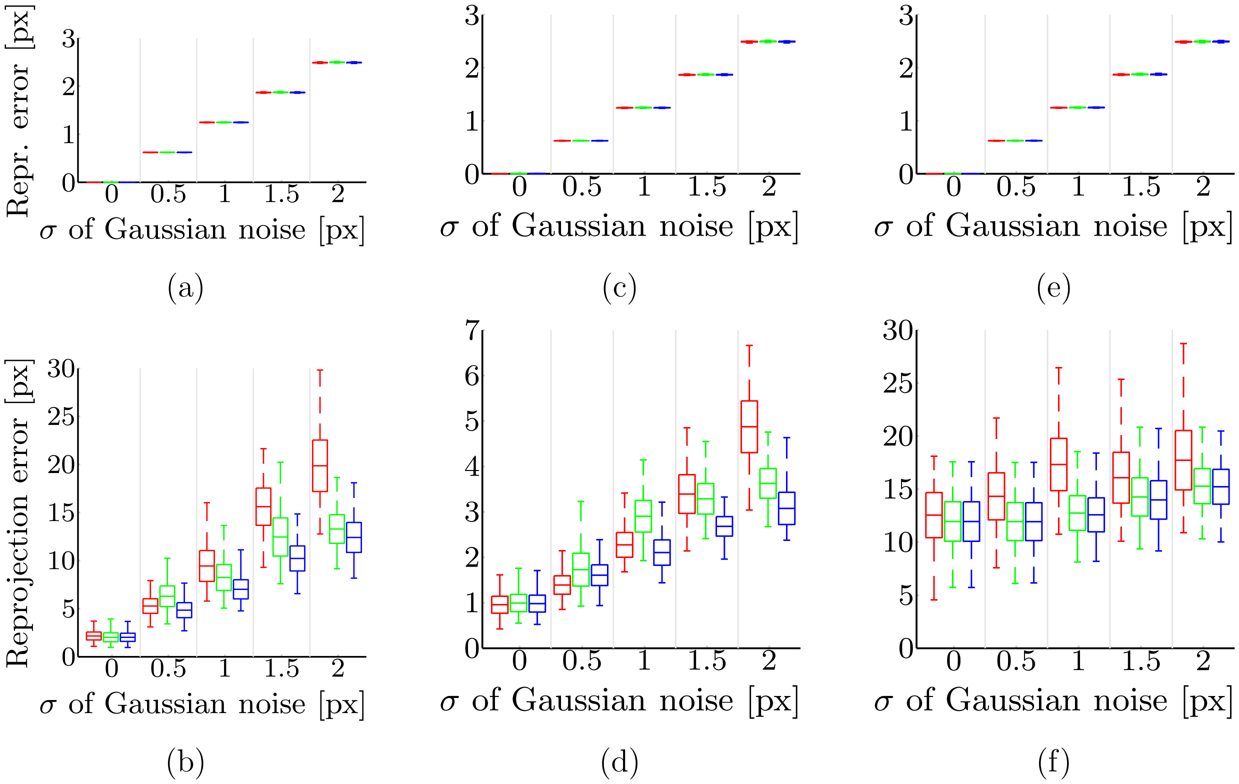
4.5 Shape optimization in Camera Calibration Procedure

All of the calibration problems presented in this paper expect the projection coordinates x_i, y_i, $\hat{x}_i$, and $\hat{y}_i$ to be known, see Equation 1. This assumes a known calibration target $\mathbf{X}_i \in \mathbb{R}^3$ as well as known camera parameters $\mathtt{R} \in SO(3)$, $\mathbf{t} \in \mathbb{R}^3$, and the calibration matrix $\mathtt{K} \in \mathbb{R}^{3\times3}$. A straightforward idea how to fold the shape optimized radial distortion calibration into the camera calibration procedure is to first perform "classical" camera calibration [20, 21, 8, 18], including radial distortion estimation. Once the projection coordinates are known, the shape optimized radial distortion calibration can be performed to replace the radial distortion parameters estimated by a classical method. One might argue that the quality of such a solution could be compromised, since different error functions may be considered by the camera and the shape optimization calibration methods. To mitigate this problem, we suggest an alternating approach to "shape-optimize" the results of the classical camera calibration: first, the shape optimization procedure is performed, followed by a bundle adjustment [19] step where the radial distortion parameters are fixed. This can be repeated in a loop for a fixed number of times, or until desired convergence is reached.

5 Experiments

To validate the proposed approach, this section presents several experimental results on synthetic as well as real world datasets. We implemented Problems 4, 5, and 6 in Matlab using Yalmip toolbox [11] with SeDuMi [15] as the underlying SDP solver. Yalmip toolbox is a modeling language that can be used to solve LMI as well as PMI programs, which it automatically translates into LMI relaxations using the scheme presented in Section 3.3. All of the resulting SDP programs were solved under a second on an Intel i7 3.50GHz based desktop computer running Linux and 64bit Matlab.

Synthetic experiment. In the synthetic experiment, we studied the performance of the proposed method with respect to the image noise. We generated a synthetic 16×16 planar calibration target. A scene consisted of 9 random 640×480 pixel cameras randomly positioned on a hemisphere around the target and rotated to face its center. The focal length was set to approx. 540 px and the distances of the camera centers from the target were set up so that the target (calibration data point set) covered only the middle part of the field of view, approx 50%. For each of the three model-shape problem combinations, we generated 100 scenes and corrupted the projections of the calibration target by an increasing amount of Gaussian image noise in 5 levels, standard deviation $\sigma \in [0, 2]$ px in $1/2$ px steps. We calibrated all scenes with OpenCV [2] made to disregard the



(a) (c) (e)

(b) (d) (f)

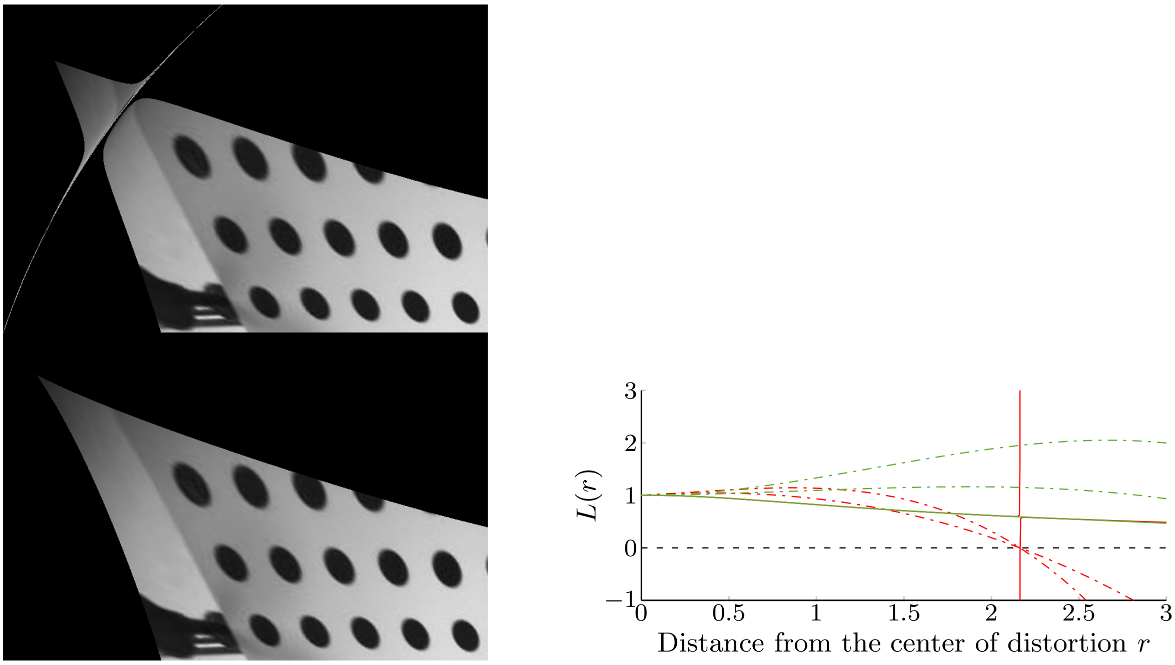
Figure 2: *Image noise experiment.* Methods BA, SO, and ASO in red, green, and blue, respectively, on calibration and validation data point sets. (a–b) barrel distortion, (c–d) pincushion distortion (e–f) zero-crossing problem.

radial distortion component. We compare three methods: the first method (BA) is the bundle adjustment method that included the respective radial distortion model performed together with the OpenCV calibration results, the second method (SO) is the respective shape-optimization method performed after the BA step, and the last method (ASO) is the alternating approach from Section 4.5, fixed to 10 iterations.

Barrel distortion. First, we experimented with the barrel distortion and the polynomial model $L(r) = f(r)$. Figure 2(a) shows the mean of the reprojection errors on the calibration data point set for methods BA, SO, and ASO using MATLAB function `boxplot`. The methods show identical performance, however when a validation data set of points covering the whole field of view is used, see Figure 2(a), we see both SO and ASO outperforming the classical BA approach.

Pincushion distortion. Next, Figures 2(c–d) show the analogous measure for the pincushion distortion and the division model $L(r) = \frac{1}{g(x)}$. Here, both BA and shape-optimization methods perform significantly better on the validation data point set. Still, we can see superior performance of SO and ASO as the noise increases.

Zero-crossing. Finally, we experimented with the rational model $L(r) = \frac{f(r)}{g(r)}$ and the mustache type distortion. Figure 2(d) shows identical performance on the calibration dataset. On the other hand, we can see poor performance on the validation data point set even if no noise is present, Figure 2(e). This is caused by the fact that too few calibration points were on the outer parts of the field of view where the convexity of the distortion



(a) (b)

Figure 3: *Real experiment.* Correction of the zero-crossing problem of the rational model.

function changes. Again, we see better performance of SO and ASO methods.

Real experiment. In the real experiment, we calibrated a 2 MPix camera from Point Grey's Ladybug 3 system [1] using 12 images of a known 28×20 planar target. Calibration using BA method and the rational model introduced quite noticeable zero-crossing problem. As expected, calibration using ASO method does not suffer from this type of problem. In this experiment, we set $\bar{r} = 4$ and $p = 0.1$. Figure 3(a) shows the upper left corner of a rectified calibration image using $\mathbf{k}$ provided by methods BA and ASO, respectively. Figure 3(b) shows the shape of the BA calibration function in red and the ASO calibration in green.

6 Conclusion

The aim of this work was not to argue for a specific radial distortion model, but to point out extrapolation problems inherent to all polynomial and rational models. We solved these problems by enforcing a predetermined shape of the distortion function. For most shapes and models, the proposed approach leads to small semidefinite programming problems that can be solved fast and globally optimally. We also showed how to deal with shapes and models that lead to PMI problems using a LMI relaxation scheme. We showed experimentally that in terms of the reprojection error on the known data points the proposed approach provides radial distortion models that are equivalent to those provided by the classical bundle adjustment approach, yet with the added value of having the correct shape that mollifies or completely removes all extrapolation issues.

References

[1] Ladybug 3 camera. `www.ptgrey.com/products/ladybug3`.

[2] Open source computer vision library. `www.opencv.org`.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, March 2004.

[4] Duane C. Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):444–462, 1966.

[5] Duane C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[6] Man-Duen Choi, Tsit Yuen Lam and Bruce Reznick. Sums of squares of real polynomials. In *Proceedings of Symposia in Pure mathematics*, volume 58, pages 103–126. American Mathematical Society, 1995.

[7] Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 2012.

[8] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1309–1321, 2007.

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge University, Cambridge, 2nd edition, 2003.

[10] Didier Henrion and Jean-Bernard Lasserre. Convergent relaxations of polynomial matrix inequalities and static output feedback. *IEEE Transactions on Automatic Control*, 51(2):192–202, 2006.

[11] Johan Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the IEEE Symposium on CACSD*, Taipei, Taiwan, 2004.

[12] Lili Ma, YangQuan Chen, and Kevin L. Moore. Rational radial distortion models of camera lenses with analytical solution for distortion correction. *International Journal of Information Acquisition*, 1(02):135–147, 2004.

[13] Yurii Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000.

[14] Chester C. Slama, Charles Theurer, Soren W. Henriksen, et al. *Manual of photogrammetry.* Number Ed. 4. American Society of photogrammetry, 1980.

[15] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[16] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and Joao Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1–2):1–183, 2011.

[17] Richard Szeliski. *Computer vision: algorithms and applications.* Springer, 2010.

[18] Jean-Philippe Tardif, Peter Sturm, Martin Trudeau, and Sebastien Roy. Calibration of cameras with radially symmetric distortion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1552–1566, 2009.

[19] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000.

[20] Roger Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1986.

[21] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.