Two Minimal Problems for Cameras with Radial Distortion

Zuzana Kukelova Tomas Pajdla
Center for Machine Perception, Dept. of Cybernetics, Faculty of Elec. Eng.
Czech Technical University in Prague, Karlovo nám. 13, 121 35 Prague, Czech Rep.
{kukelova,pajdla}@cmp.felk.cvut.cz

Abstract

Epipolar geometry and relative camera pose computation for uncalibrated cameras with radial distortion has recently been formulated as a minimal problem and successfully solved in floating point arithmetics. The singularity of the fundamental matrix has been used to reduce the minimal number of points to eight. It was assumed that the cameras were not calibrated but had same distortions. In this paper we formulate two new minimal problems for estimating epipolar geometry of cameras with radial distortion. First we present a minimal algorithm for partially calibrated cameras with same radial distortion. Using the trace constraint which holds for the epipolar geometry of calibrated cameras to reduce the number of necessary points from eight to six. We demonstrate that the problem is solvable in exact rational arithmetics. Secondly, we present a minimal algorithm for uncalibrated cameras with different radial distortions. We show that the problem can be solved using nine points in two views by manipulating polynomials by a sequence of Gauss-Jordan eliminations in exact rational arithmetics. We demonstrate the algorithms on synthetic and real data.

1. Introduction

Estimating [1] camera models from image matches in an important problem. Recently, epipolar geometry and relative camera pose computation for cameras with radial distortion has been formulated as a minimal problem and successfully solved in floating point arithmetics [12]. The epipolar geometry was estimated altogether with the radial distortion parameter following the paradigm of autocalibration of cameras with radial distortion [21, 27, 11, 5, 17, 18, 25, 20] as a minimal problem in the spirit of previous work on minimal problems, e.g. the perspective three point problem [4, 7], the five point relative pose problem [19, 22, 15], the six point focal length problem [23, 13], the six point generalized camera problem [24], the nine-point problem for estimating para-catadioptric fundamental matrices [8] and the nine point radial distortion problem [14].

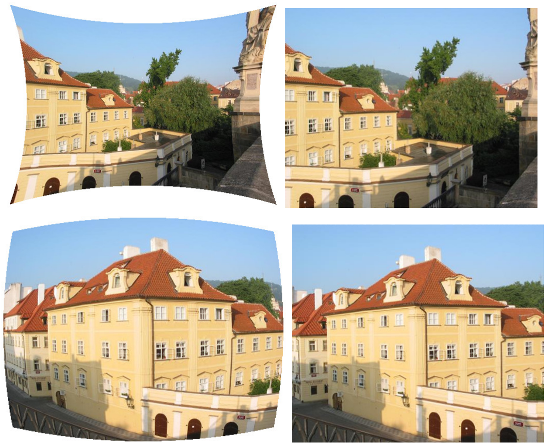

Figure 1. Real data. (Left) Input images with different radial distortions for left (Top) and right camera (Bottom). (Right) Corrected images.

The problem [12] was formulated as a system of algebraic equations and the singularity of the fundamental matrix was used to reduce the minimal necessary number of points from nine [5, 14] to eight. It was assumed that cameras were not fully calibrated and that the radial distortion was same for both of them.

In this paper we use the same minimal problem paradigm and extend the result [12] in two ways.

Many digital cameras have square pixels and the principal point as well as the radial distortion center are in the center of image. Thus they are partially calibrated [10]. In the first contribution of this paper we use the trace constraint which holds for the epipolar geometry of completely calibrated cameras to reduce the number of necessary points from eight to six. Here we also assume same distortion in both cameras. We demonstrate that the problem is solvable in exact arithmetics.

Without any information about images, it is not safe to

[1] This work has been supported by grants EU FP6-IST-027787 DIRAC and MSM6840770038 DMCM III.

assume anything about camera calibration and possibly not even that both cameras in the stereopair have the same radial distortion. In the second contribution of this paper we allow radial distortion be different in different cameras and provide a minimal solution which can compute epipolar geometry of cameras from nine correspondences. This is a considerable advancement of the previosuly known non-minimal solution which used 15 points [1].

A smaller number of the points considerably reduces the number of samples in RANSAC [4, 10]. On the other hand, the resulting systems of polynomial equations are more difficult than, e.g., the systems arising from similar problems for estimating epipolar geometry of perspective cameras [23, 22]. In principal, every solvable system of polynomial equations can be algorithmically solved by finding the reduced Gröbner basis of its corresponding ideal [2, 3]. In practice however, two main problems occur. First, the construction of a Göbner basis has in general exponential complexity and many practical problems lead to difficult an impractical computations. Secondly, even tractable computations may require extreme numerical precision to yield useful results.

Two problems presented in this paper are considerably more difficult than the corresponding problems for perspective cameras but they are still tractable. The first "calibrated problem" of this paper, which is more difficult, required to generate 119 new polynomials in 182 monomials up to 6^{th} order and to do 7 Gauss-Jordan eliminations to obtain 52 solutions. The second "uncalibrated problem" was solved by generating 97 new polynomials in 125 monomials up to 5^{th} order and required to do 7 Gauss-Jordan eliminations to obtain 24 solutions.

Problems solved in this paper seem to call for significantly higher precision than the problem solved in [12]. Therefore, we present here a solution computed in exact rational arithmetic in Maple. Construction of a robust floating point solver is a non-trivial issue which will be treated elsewhere. Computing in rational arithmetics is much slower than floating point computations (seconds instead of milliseconds) but it is directly applicable to all real measurements in images. Consider that image measurements are always obtained in limited resolution, say 1/10 of pixel. Thus, we can represent an image measurement u as integer $10\,u$ and start with equations with integer coefficients. Manipulating equations can be done exactly and will produce rational coefficients. Only at the end we will need to compute solutions approximately, e.g. in floating point arithmetics, to find approximations of possibly non-rational roots of polynomial equations.

In the next section we briefly summarize the method for solving systems of algebraic equations, and then present the formulations and solutions of the two problems for this paper. Finally we demonstrate that the algorithms work.

2. Solving algebraic equations

Our goal is to solve a system of algebraic equations $f_1(x) = ... = f_m(x) = 0$ which are given by a set of m polynomials $F = \{f_1, ..., f_m | \, f_i \in \mathbb{C}[x_1, ..., x_n]\}$ in n variables over the field of complex numbers. The method which we use to solve a system of algebraic equations is based on the method proposed in [12]. Here the solver consists of two main parts.

The goal of the first part is to simplify the input equations by reducing the number of variables. This is done because the complexity of the second part of the solver, computation of so called "action matrix" depends on the complexity of polynomials. It is better to have the degrees as well as the number of variables low. The reduction of the number of variables may considerably help to solve the system.

The second part of the solver is based on the Gröbner basis G of the ideal $I = \{\sum_{i=1}^{m} f_i \, p_i \, | \, p_i \in \mathbb{C}[x_1, ..., x_n]\}$ generated by polynomials F and on constructing the action matrix in the *quotient ring* $A = \mathbb{C}[x_1, ..., x_n]/I$, i.e. the set of equivalence classes represented by remainders modulo I. The action matrix $\mathtt{M}_f$ is the matrix of the linear operator $T_f \colon A \rightarrow A$ of the multiplication by a suitably chosen polynomial f w.r.t. the basis $B = \left\{\mathbf{x}^\alpha | \overline{\mathbf{x}^\alpha}^G = \mathbf{x}^\alpha\right\}$ of A, where $\mathbf{x}^\alpha$ is a monomial $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} ... x_n^{\alpha_n}$ and $\overline{\mathbf{x}^\alpha}^G$ is the reminder of $\mathbf{x}^\alpha$ on the division by G. The solutions to the set of equations can be read off directly from the eigenvalues and eigenvectors of the action matrix [3].

To construct the action matrix we use the same method as was proposed in [12] where this matrix was constructed without computing the complete Gröbner basis G. In this method the basis B of A is found once in advance and then the action matrix is constructed by generating polynomials $q_i = f\mathbf{x}^{\alpha(i)} + h_i$ from I with $h_i = \sum_{j=1}^{N} c_{ji} \mathbf{x}^{\alpha(j)} \in A$ and $\mathbf{x}^{\alpha(i)} \in B$.

To construct the action matrix we need to compute $T_f\left(\mathbf{x}^{\alpha(i)}\right) = \overline{f\mathbf{x}^{\alpha(i)}}^G$ for all $\mathbf{x}^{\alpha(i)} \in B$ [3]. But if for some $\mathbf{x}^{\alpha(i)} \in B$ and chosen f, $f\mathbf{x}^{\alpha(i)} \in A$, then $T_f\left(\mathbf{x}^{\alpha(i)}\right) = \overline{f\mathbf{x}^{\alpha(i)}}^G = f\mathbf{x}^{\alpha(i)}$ and we are done. For all other $\mathbf{x}^{\alpha(i)} \in B$ for which $f\mathbf{x}^{\alpha(i)} \notin A$ we consider mentioned polynomials q_i. For these $\mathbf{x}^{\alpha(i)}$, $T_f\left(\mathbf{x}^{\alpha(i)}\right) = \overline{f\mathbf{x}^{\alpha(i)}}^G = \overline{q_i - h_i}^G = -h_i \in A$.

Since polynomials q_i are from the ideal I, we can generate them as algebraic combinations of the initial generators F. The method starts with F and then systematically generate new polynomials from I by multiplying already generated polynomials by individual variables and reducing them by the Gauss-Jordan elimination. This algorithm is similar to F4 algorithm [6] for constructing Gröbner bases. See [12] for more about this method for constructing the action matrix.

3. Minimal problems for correcting radial distortion

We want to correct radial lens distortion using the minimal number of image point correspondences in two views. We consider and two new minimal problems:

1. The problem of estimating one-parameter radial distortion model and epipolar geometry from image point correspondences in two partially calibrated views.
2. The problem of estimating one-parameter radial distortion model and epipolar geometry from image point correspondences in two uncalibrated views with different radial distortions in each image.

It is well known that for the standard calibrated case without considering radial distortion, 5 point correspondences are sufficient and necessary to estimate the epipolar geometry. When estimating radial distortion from point correspondences in calibrated views, we have one more parameter, the radial distortion parameter λ. Therefore, for this problem, we will need 6 point correspondences to estimate λ and the epipolar geometry.

For standard uncalibrated case, 7 point correspondences are sufficient and necessary. When estimating two different radial distortions in two views, two radial distortion parameters λ_1 and λ_2 appear. Therefore, for this problem the minimal number of point correspondences which is needed to estimate λ_1, λ_2 and the epipolar geometry is 9.

In both these problems we assume one-parameter division model [5] given by the formula

$$\mathbf{p}_u \sim \mathbf{p}_d/(1+\lambda r_d^2) \tag{1}$$

where λ is the distortion parameter, $\mathbf{p}_u = (x_u, y_u, 1)$, resp. $\mathbf{p}_d = (x_d, y_d, 1)$, are the corresponding undistorted, resp. distorted, image points, and r_d is the radius of $\mathbf{p}_d$ w.r.t. the distortion center. We assume that the distortion center has been found, e.g., by [9]. We also assume square pixels, i.e. $r_d^2 = x_d^2 + y_d^2$. To use the standard notation, we write the division model as

$$\mathbf{x} + \lambda\mathbf{z} = \begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 0 \\ r_d^2 \end{pmatrix} \sim \begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix}. \tag{2}$$

In the next sections we give solutions to these minimal problems for correcting radial distortion.

3.1. Calibrated cameras

To get solutions to the minimal problem of estimating one parameter radial distortion model from point correspondences in two calibrated images, we have to use equations from the *epipolar constraint* for 6 point correspondences

$$\mathbf{p}_{u_i}^{\top}(\lambda)\,\mathtt{E}\,\mathbf{p}'_{u_i}(\lambda) = 0, \quad i = 1,\ldots,6 \tag{3}$$

$$\mathtt{E} = \begin{pmatrix} e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,1} & e_{3,2} & e_{3,3} \end{pmatrix}, \tag{4}$$

the *singularity of the essential matrix* $\mathtt{E}$

$$\det(\mathtt{E}) = 0, \tag{5}$$

and the *trace condition*, which says that two singular values of the essential matrix are equal

$$2\left(\mathtt{E}\mathtt{E}^T\right)\mathtt{E} - trace(\mathtt{E}\mathtt{E}^T)\mathtt{E} = 0. \tag{6}$$

We obtain 16 equations in 10 unknowns. Assuming $e_{3,3} \neq 0$, we can set $e_{3,3} = 1$.

3.1.1 Eliminating variables

The epipolar constraint gives 6 equations with 15 monomials $(\lambda e_{1,3}, \lambda e_{2,3}, \lambda e_{3,1}, \lambda e_{3,2}, \lambda^2, e_{1,1}, e_{1,2}, e_{1,3}, e_{2,1}, e_{2,2}, e_{2,3}, e_{3,1}, e_{3,2}, \lambda, 1)$ and 9 variables $(e_{1,1}, e_{1,2}, e_{1,3}, e_{2,1}, e_{2,2}, e_{2,3}, e_{3,1}, e_{3,2}, \lambda)$.

Among them, we have four variables which appear in one monomial only $(e_{1,1}, e_{1,2}, e_{2,1}, e_{2,2})$ and four variables which appear in two monomials $(e_{1,3}, e_{2,3}, e_{3,1}, e_{3,2})$. Since we have six equations from which each contains all 15 monomials we can eliminate five from these nine variables. We select the first four variables $e_{1,1}, e_{1,2}, e_{2,1}, e_{2,2}$ that appear in one monomial only (and can be straightforwardly eliminated) and the fifth variable as $e_{1,3}$ which appears in two monomials.

We reorder monomials contained in 6 equations putting monomials containing $e_{1,1}, e_{1,2}, e_{2,1}, e_{2,2}$ and $e_{1,3}$ at the beginning. The reordered monomial vector will be $X = (e_{1,1}, e_{1,2}, e_{2,1}, e_{2,2}, e_{1,3}\lambda, e_{1,3}, e_{2,3}\lambda, e_{3,1}\lambda, e_{3,2}\lambda, \lambda^2, e_{2,3}, e_{3,1}, e_{3,2}, \lambda, 1)^T$.

We can rewrite 6 equations from the epipolar constraint in a matrix form $\mathtt{M}\,X = 0$, where $\mathtt{M}$ is the coefficient matrix. After performing the Gauss-Jordan (G-J) elimination on matrix $\mathtt{M}$, we obtain 6 equations of the form

$$f_i = LT(f_i) + g_i(e_{2,3}, e_{3,1}, e_{3,2}, \lambda) = 0, \tag{7}$$

where $LT(f_i) = e_{1,1}, e_{1,2}, e_{2,1}, e_{2,2}, e_{1,3}\lambda$, resp. $e_{1,3}$ for $i = 1, 2, 3, 4, 5$ resp. 6 and $g_i(e_{2,3}, e_{3,1}, e_{3,2}, \lambda)$ are 2^{nd}order polynomials in 4 variables $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$. So, five variables $e_{1,1}, e_{1,2}, e_{1,3}, e_{2,1}, e_{2,2}$ can be expressed as functions of the other four variables $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$.

$$\begin{aligned} e_{1,1} &= -g_1(e_{2,3}, e_{3,1}, e_{3,2}, \lambda) \\ e_{1,2} &= -g_2(e_{2,3}, e_{3,1}, e_{3,2}, \lambda) \\ e_{1,3} &= -g_6(e_{2,3}, e_{3,1}, e_{3,2}, \lambda) \\ e_{2,1} &= -g_3(e_{2,3}, e_{3,1}, e_{3,2}, \lambda) \\ e_{2,2} &= -g_4(e_{2,3}, e_{3,1}, e_{3,2}, \lambda). \end{aligned} \tag{8}$$

We can substitute these expressions to the remaining equation from the epipolar constraint and to the singularity and trace constraint for $\mathtt{E}$. In this way we obtain 11 polynomial equations in 4 unknowns (one of degree 3, four of degree 5 and six of degree 6)

One equation from the epipolar constraint

$$\lambda(-g_6(e_{2,3}, e_{3,1}, e_{3,2}, \lambda)) + g_5(e_{2,3}, e_{3,1}, e_{3,2}, \lambda) = 0 \quad (9)$$

one equation from the singularity constraint

$$\det(\mathtt{E}) = 0, \quad (10)$$

and 9 equations from the trace constraint

$$2\left(\mathtt{EE}^T\right)\mathtt{E} - trace(\mathtt{EE}^T)\mathtt{E} = 0, \quad (11)$$

with

$$E = \begin{pmatrix} -g_1 & -g_2 & -g_6 \\ -g_3 & -g_4 & e_{2,3} \\ e_{3,1} & e_{3,2} & 1 \end{pmatrix}. \quad (12)$$

Using these 11 equations we create the action matrix for the polynomial $f = \lambda$ as described next.

3.1.2 Computing B and the number of solutions

For both problems we use algebraic geometry software Macaulay 2, which can compute in finite fields, to solve the polynomial equations for many random coefficients from $\mathbb{Z}_p$, to compute the number of solutions, the Gröbner basis, and the basis B [26]. Here the exact arithmetic can be used and numbers can be represented in a simple and efficient way. It speeds up computations, minimizes memory requirements and especially avoid numerical instability. Computing with coefficients represented as floating point approximations may lead to numerical instability when computing the Gröbner basis and action matrix, since it may be difficult to determine when coefficients become zero in the reduction.

If the basis B remains stable for many different random coefficients from $\mathbb{Z}_p$, i.e. if it consists of the same monomials, it is generically equivalent to the basis of the original system of polynomial equations. Also, the way of obtaining polynomials that are necessary to create the action matrix is always the same and for general data the generated polynomials differ only in their coefficients. Once this is known, action matrix can be created for rational coefficients or even their floating point approximation.

Using Macaulay 2 to solve the above system of 11 polynomial equations for many random coefficients from $\mathbb{Z}_p$, we compute the number of solutions, the Gröbner basis, and the basis B for this problem.

In this way we have found that this problem has 52 solutions. To create the action matrix, we used the graded reverse lexicographic ordering $e_{2,3} > e_{3,1} > e_{3,2} > \lambda$ and got the basis $B = (e_{3,1}^2 e_{3,2}^2, e_{2,3} e_{3,2}^3, e_{3,1} e_{3,2}^3, e_{3,2}^4, e_{2,3}^3\lambda, e_{3,1}^3\lambda, e_{2,3}^2 e_{3,2}\lambda, e_{2,3} e_{3,1} e_{3,2}\lambda, e_{3,1}^2 e_{3,2}\lambda, e_{2,3} e_{3,2}^2\lambda, e_{3,1} e_{3,2}^2\lambda, e_{3,2}^3\lambda, e_{3,1}^2\lambda^2, e_{3,1} e_{3,2}\lambda^2, e_{3,2}^2\lambda^2, e_{3,1}\lambda^3, e_{3,2}\lambda^3, \lambda^4, e_{2,3}^3, e_{2,3}^2 e_{3,1}, e_{2,3} e_{3,1}^2, e_{3,1}^3, e_{2,3}^2 e_{3,2}, e_{2,3} e_{3,1} e_{3,2}, e_{3,1}^2 e_{3,2}, e_{2,3} e_{3,2}^2, e_{3,1} e_{3,2}^2, e_{3,2}^3, e_{2,3}^2\lambda, e_{2,3} e_{3,1}\lambda, e_{3,1}^2\lambda, e_{2,3} e_{3,2}\lambda, e_{3,1} e_{3,2}\lambda, e_{3,2}^2\lambda, e_{3,1}\lambda^2, e_{3,2}\lambda^2, \lambda^3, e_{2,3}^2, e_{2,3} e_{3,1}, e_{3,1}^2, e_{2,3} e_{3,2}, e_{3,1} e_{3,2}, e_{3,2}^2, e_{2,3}\lambda, e_{3,1}\lambda, e_{3,2}\lambda, \lambda^2, e_{2,3}, e_{3,1}, e_{3,2}, \lambda, 1)$ of the algebra $A = \mathbb{C}\left[e_{2,3}, e_{3,1}, e_{3,2}, \lambda\right]/I$.

3.1.3 Constructing action matrix

We construct the action matrix $\mathtt{M}_\lambda$ for multiplication by polynomial $f = \lambda$. The method described in Section 2 calls for generating polynomials $q_i = \lambda\mathbf{x}^{\alpha(i)} + \sum_{j=1}^{N} c_{ji}\mathbf{x}^{\alpha(j)} \in I$. As described in Section 2, we will do this by systematically generating higher order polynomials from I. This will be done by adding monomial multiples of initial polynomial equations. We stop when all necessary polynomials q_i are obtained.

The basic steps of generating the polynomials necessary for constructing the action matrix for the calibrated problem are as follows:

1. We begin with one 3^{rd}, four 5^{th} and six 6^{th} degree polynomials. In the first step we multiply the one 3^{rd} degree polynomial with all four variables $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$. This polynomial together with its multiples can be represented by 5×35 matrix with rank 5 which we simplify by G-J elimination.
2. In the next step we add $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$ multiples of four new 4^{th} degree polynomials to the existing 5 polynomials. Together with 5^{th} degree polynomial from $\det(F)$ we obtain 22 polynomials which can be represented by a 22×75 matrix with rank 16. We again perform G-J elimination.
3. We obtain one new 4^{th} degree polynomial and 10 new 5^{th} degree polynomials. In this step we add $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$ multiples of this 4^{th} degree polynomial to the already generated 16 polynomials. In this way we obtain 20 polynomials representable by a 20×103 matrix with rank 20. We simplify it by G-J elimination.
4. Now we have four new 5^{th} degree polynomials. In this step we add $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$ multiples of all 14 5^{th} degree polynomials (4 new and 10 generated in the previous step) to already generated 20 polynomials. Together with three 5^{th} degree and six 6^{th} degree polynomials from the trace constraint we obtain 85 polynomials representable by a 85×180 matrix, which has rank 59. We simplify it by G-J elimination.
5. After G-J elimination we obtain one new 4^{th} degree polynomial, five new 5^{th} degree polynomials and 33 new 6^{th} degree polynomials. We add $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$

multiples of the one new 4^{th} and five new 5^{th} degree polynomials to already generated 59 polynomials. Obtained 83 polynomials can be represented by a 83×180 matrix, which has rank 82. We again perform G-J elimination.

6. Now we have 12 new 5^{th} degree, 11 new 6^{th} degree polynomials. Again by adding $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$ multiples of these 12 new 5^{th} degree polynomials to the generated 82 polynomials we obtain 130 polynomials which can be represented by a 130×182 matrix with rank 119. We perform G-J elimination on this matrix.
7. In the last step we add $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$ multiples of seven new generated 4^{th} degree polynomials to previously generated 119 polynomials. Obtained 147 polynomials can be represented by a 147×182 matrix, which has rank 130. We perform the last G-J elimination.
8. All polynomials needed for constructing the action matrix have been generated. Action matrix M_λ has been constructed.

3.1.4 Solving equations using eigenvectors

The eigenvectors of $\mathtt{M}_\lambda$ give solutions for $e_{2,3}, e_{3,1}, e_{3,2}, \lambda$. Using a backsubstitution, we obtain solutions also for $e_{1,1}, e_{1,2}, e_{1,3}, e_{2,1}, e_{2,2}$. In this way we obtain 52 (in general complex) solutions.

3.2. Uncalibrated cameras with different radial distortion

To get solutions to the minimal problem of estimating one parameter radial distortion model from point correspondences in two uncalibrated image with different radial distortion λ_1 and λ_2 in each image, we have to use equations from the epipolar constraint for 9 point correspondences

$$\mathbf{p}_{u_i}^\top(\lambda_1)\,\mathtt{F}\,\mathbf{p}'_{u_i}(\lambda_2) = 0, \quad i = 1, \ldots, 9 \tag{13}$$

$$\mathtt{F} = \begin{pmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{pmatrix} \tag{14}$$

and the singularity of the fundamental matrix $\mathtt{F}$

$$\det(\mathtt{F}) = 0. \tag{15}$$

We obtain 10 equations in 11 unknowns. Assuming $f_{3,3} \neq 0$, we can set $f_{3,3} = 1$.

3.2.1 Eliminating variables

The epipolar constraint gives 9 equations with 16 monomials $(f_{3,1}\lambda_1, f_{3,2}\lambda_1, f_{1,3}\lambda_2, f_{2,3}\lambda_2, \lambda_1\lambda_2, f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}, f_{3,1}, f_{3,2}, \lambda_1, \lambda_2, 1)$ and 10 variables $(f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}, f_{3,1}, f_{3,2}, \lambda_1, \lambda_2)$.

Using similar elimination method like in the calibrated case we can eliminate 6 from these 10 variables. All these variables can be eliminated at once.

We have again four variables which appear in one monomial only $(f_{1,1}, f_{1,2}, f_{2,1}, f_{2,2})$ and four variables which appear in two monomials $(f_{1,3}, f_{2,3}, f_{3,1}, f_{3,2})$. Since we have 9 equations from the epipolar constraint we can use these equations to eliminate 6 variables, four variables which appear in one monomial only and two of the variables which appear in two monomials. In this solution we have selected $f_{1,3}$ and $f_{2,3}$.

We reorder monomials contained in 9 equations and put monomials containing $f_{1,1}, f_{1,2}, f_{2,1}, f_{2,2}, f_{1,3}$ and $f_{2,3}$ at the beginning. Reordered monomial vector becomes $X = (f_{1,1}, f_{1,2}, f_{2,1}, f_{2,2}, f_{1,3}\lambda_2, f_{1,3}, f_{2,3}\lambda_2, f_{2,3}, f_{3,1}\lambda_1, f_{3,2}\lambda_1, \lambda_1\lambda_2, f_{3,1}, f_{3,2}, \lambda_1, \lambda_2, 1)^T$.

We rewrite 9 equations from the epipolar constraint in a matrix form $\mathtt{M}X = 0$. After performing G-J elimination we obtain 9 equations of the form

$$f_i = LT(f_i) + g_i(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) = 0, \tag{16}$$

where $LT(f_i) = f_{1,1}, f_{1,2}, f_{2,1}, f_{2,2}, f_{1,3}\lambda_2, f_{1,3}, f_{2,3}\lambda_2, f_{2,3}$ resp. $f_{3,1}\lambda_1$ for $i = 1, 2, 3, 4, 5, 6, 7, 8$ resp. 9 and $g_i(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2)$ are 2^{nd}order polynomials in four variables $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$. So we can express 6 variables, $f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}$ as the functions of other four variables $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$.

$$\begin{aligned} f_{1,1} &= -g_1(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) \\ f_{1,2} &= -g_2(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) \\ f_{1,3} &= -g_6(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) \\ f_{2,1} &= -g_3(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) \\ f_{2,2} &= -g_4(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) \\ f_{2,3} &= -g_8(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2), \end{aligned} \tag{17}$$

Substituting these expressions to other three equations from the epipolar constraint and also to the singularity constraint for F gives 4 polynomial equations in 4 unknowns (one of 2^{nd} degree, two of 3^{rd} degree and one of 5^{th} degree)

$$\lambda_2(-g_6(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2)) + g_5(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) = 0$$

$$\lambda_2(-g_8(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2)) + g_7(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) = 0$$

$$f_{3,1}\lambda_1 + g_9(f_{3,1}, f_{3,2}, \lambda_1, \lambda_2) = 0$$

$$\det\begin{pmatrix} -g_1 & -g_2 & -g_6 \\ -g_3 & -g_4 & -g_8 \\ f_{3,1} & f_{3,2} & 1 \end{pmatrix} = 0. \tag{18}$$

Using these four equations we create the action matrix for the polynomial $f = \lambda_2$.

3.2.2 Computing B and the number of solutions

Using Macaulay 2 we have found that this problem has 24 solutions. To create the action matrix, we use the graded reverse lexicographic ordering $f_{3,1} > f_{3,2} > \lambda_1 > \lambda_2$. With this ordering, we get the basis $B = (\lambda_2^4, f_{3,1}f_{3,2}^2, f_{3,2}^3, f_{3,2}\lambda_1^2, \lambda_1^3, f_{3,2}^2\lambda_2, \lambda_1^2\lambda_2, f_{3,1}\lambda_2^2, f_{3,2}\lambda_2^2, \lambda_2^3, f_{3,1}^2, f_{3,1}f_{3,2}, f_{3,2}^2, f_{3,2}\lambda_1, \lambda_1^2, f_{3,1}\lambda_2, f_{3,2}\lambda_2, \lambda_1\lambda_2, \lambda_2^2, f_{3,1}, f_{3,2}, \lambda_1, \lambda_2, 1)$ of the algebra $A = \mathbb{C}\left[f_{3,1}, f_{3,2}, \lambda_1, \lambda_2\right]/I$.

3.2.3 Constructing action matrix

In this problem we construct the action matrix $\mathtt{M}_{\lambda_2}$ for multiplication by polynomial $f = \lambda_2$. We again generate polynomials $q_i = \lambda_2 \mathbf{x}^{\alpha(i)} + \sum_{j=1}^{N} c_{ji}\mathbf{x}^{\alpha(j)} \in I$ by systematically generating higher order polynomials from I.

The basic steps of generating the polynomials necessary for constructing the action matrix for this problem are as follows:

1. We begin with one 2^{nd}, two 3^{rd} and one 5^{th} degree polynomials. In the first step we multiply the 2^{nd} and the 3^{rd} degree polynomials with all four variables $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$. These three polynomials and their multiples can be represented by 37 monomials and a 15×37 matrix with rank 15 which we simplify by G-J elimination.
2. We obtain five new 3^{rd} degree polynomials and 7 new 4^{th} degree polynomials. In this step we add $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$ multiples of these five 3^{rd} degree polynomials to already generated 15 polynomials. So we obtain 35 polynomials representable by a 35×63 matrix, which has rank 29. We simplify it by G-J elimination.
3. After G-J elimination we obtain two new 3^{rd} degree polynomials and 12 new 4^{th} degree polynomials. We again add $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$ multiples of two new 3^{rd} degree polynomials to already generated 29 polynomials. Obtained 37 polynomials can be represented by a 37×64 matrix, which has rank 34. We again perform G-J elimination.
4. We obtain five new 4^{th} degree polynomials. In this step we add $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$ multiples of all 24 4^{th} degree polynomials (5 new and 19 generated in previous steps) to already generated 34 polynomials. Together with 5^{th} degree polynomial from $\det(F)$ we obtain 131 polynomials representable by a 131×119 matrix, which has rank 80. We simplify it by G-J elimination.
5. Now we have one new 3^{rd} degree, one new 4^{th} degree and 44 new 5^{th} degree polynomials. Again by adding $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$ multiples of the one new 3^{rd} degree and the one new 4^{th} degree polynomial to the generated 80 polynomials we obtain 88 polynomials which can be represented by a 88×125 matrix with rank 86. We perform G-J elimination on this matrix.
6. In the next step we add $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$ multiples of the one new 3^{rd} degree and three new 4^{th} degree polynomials to the generated 86 polynomials. We obtain 102 polynomials which can be represented by a 102×125 matrix with rank 95 which is simplified by G-J elimination.
7. In the last step we add $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$ multiples of five new 4^{th} degree polynomials to previously generated 95 polynomials. Obtained 115 polynomials can be represented by a 115×125 matrix, which has rank 101. We perform last G-J elimination.
8. Action matrix M_{λ_2} has been constructed.

3.3. Solving equations using eigenvectors

The eigenvectors of $\mathtt{M}_{\lambda_2}$ give solutions for $f_{3,1}, f_{3,2}, \lambda_1, \lambda_2$. Using a backsubstitution for $f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}$ we obtain 24 (in general complex) solutions.

4. Experiments

The first calibrated problem is more difficult than the second different distortions problem. It also took much more time (1700 s compared to 30 s) to compute. It would be impractical in present form and therefore we only demonstrate that we obtained correct results for this calibrated problem and experiments with noise and outliers are left for future. The different distortions problem, on the other hand, took “only” 30 s to compute and thus we present experimental results on simulated data and an example on real images.

In both cases we get more roots, 52 in calibrated case and 24 in different distortions case. In general, we also obtain more than one real root. If there is more than one real root, we need to select the best root, the root which is consistent with most measurements. To do so, we treat the real roots obtained by solving the equations for one input as real roots from different inputs and use RANSAC [4, 10] or kernel voting [14] for several (many) inputs to select the best root among all generated roots. The kernel voting is done by a Gaussian kernel with fixed variance and the estimate of λ is found as the position of the largest peak. See [14] or [12] for more on kernel voting for this problem.

4.1. Tests on synthetic images

We initially studied results of our algorithms implemented in exact arithmetic using synthetic datasets. Our testing procedure for uncalibrated case with different radial distortions in each image was as follows:

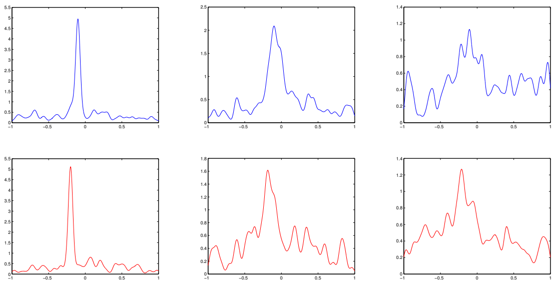

Figure 2. Kernel voting results, for ground truth $\lambda_1 = -0.1$ and $\lambda_2 = -0.2$, image size 512×512 and (Left) noiseless data and 100% inliers , (Center) noise level 2 pixels, (Right) noise level 1 pixel and 90% inliers. Estimated radial distortion parameters were (Left) $\lambda_1 = -0.1004$, $\lambda_2 = -0.2004$, (Center) $\lambda_1 = -0.1005$, $\lambda_2 = -0.1858$, and (Right) $\lambda_1 = -0.1048$, $\lambda_2 = -0.2121$.

1. Generate a 3D scene consisting of N (from 100 to 500) random points distributed uniformly within a cube. Project $M\%$ of the points on image planes of the two displaced cameras. These are matches. In both image planes, generate $(100 - M)\%$ random points distributed uniformly in the image. These are mismatches. Altogether, they become undistorted correspondences.
2. Apply different radial distortion to the undistorted correspondences in each image and in this way generate noiseless distorted points.
3. Add Gaussian noise of standard deviation σ to the distorted points.
4. Repeat K times (We use K from 50 to 100 though for more noisy data K form 100 to 200 gives better results).
   (a) Choose 9 point correspondences from given N correspondences randomly.
   (b) Normalize image point coordinates to $[-1, 1]$
   (c) Find up to 24 roots of the minimal solution to the autocalibration of radial distortion.
   (d) Select the real roots in the feasible interval, e.g., $-1 < \lambda < 1$ and the corresponding $\mathtt{F}$'s.
5. Use kernel voting to select the best root.

The resulting density functions for different noise levels and outlier contaminations are shown in Figure 2. Here, $K = 100$, image size was 512×512 and ground truth $\lambda_1 = -0.1$ and ground truth $\lambda_2 = -0.2$. In all cases, a good estimates, close to the true λ_1 and λ_2 were found as the position of the maximum of the root density function. Thus this method in exact arithmetic is robust to mismatches and noise.

Our testing procedure for the calibrated case was similar to the procedure for uncalibrated case with different radial distortions. For noiseless data we obtain comparably good estimates. For ground truth $\lambda = -0.3$ we have repeatedly obtained λ's between -0.291 and -0.306.

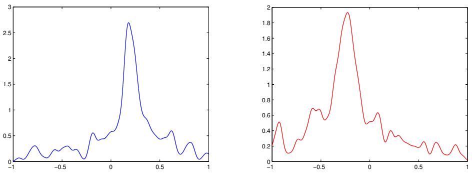

Figure 3. Distribution of real roots obtained by kernel voting for image in Figure 1. Estimated $\lambda_1 = 0.1806$ and $\lambda_2 = -0.2266$.

4.2. Tests on real images

The input images with different relatively large distortions in each image, Figure 1 (Left), were obtained by synthetically adding large distortion to (almost) perspective images. Tentative point correspondences were then found by the wide base-line matching algorithm [16]. They contained correct as well as incorrect matches. Distortion parameters λ_1 and λ_2 were estimated by Maple implementation of our 9-point algorithm for uncalibrated cameras with different radial distortions and the kernel voting method. The input (Left) and corrected (Right) images are presented in Figure 1. Figure 3 shows the distribution of real roots, for images from Figure 1, from which $\lambda_1 = 0.1806$ and $\lambda_2 = -0.2266$ were estimated as the argument of the maximum.

4.3. Conclusion

In this paper we have presented two new minimal problems for simultaneous estimation of epipolar geometry and radial distortion. We were able to demonstrate that algorithms we provided are correct and give the right solutions. We also demonstrate that the radial distortion estimation for cameras with different distortions is stable and behaves favorably in presence of noise and outliers.

We used exact rational arithmetic to be able to get working algorithms because the problems in this paper become too sensitive to rounding errors and naive approach to include only linearly independent equations before starting Gauss-Jordan elimination did not give stable results.

Our current implementation in Maple is clearly far from the optimal one. It will be the topic of future research to find whether it will be better to construct a robust approximate solver in floating point arithmetics or to use exact rational arithmetics in combination with controlled approximations.

References

[1] J. Barreto and K. Daniilidis Fundamental Matrix for Cameras with Radial Distortion. *IEEE International Conference on Computer Vision. Beijing, October 2005*

[2] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag, 1992.

[3] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer-Verlag, 2005.

[4] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.

[5] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. *CVPR 2001*, pp. 125–132.

[6] J.-C. Faugere. A new efficient algorithm for computing gröbner bases (f_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.

[7] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE PAMI*, 25(8):930–943, 2003.

[8] C. Geyer and H. Stewenius. A nine-point algorithm for estimating para-catadioptric fundamental matrices. *CVPR 2007*.

[9] R. Hartley and S. Kang. Parameter-free radial distortion correction with centre of distortion estimation. *ICCV 2005*, pp. 1834–1841.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[11] S. Kang. Catadioptric self-calibration. *CVPR 2000*, pp. I:201–208.

[12] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. *CVPR 2007*

[13] H. Li. A simple solution to the six-point two-view focal-length problem. *ECCV 2006*, pp. 200–213.

[14] H. Li and R. Hartley. A non-iterative method for correcting lens distortion from nine-point correspondences. *OMNIVIS 2005*.

[15] H. Li and R. Hartley. Five-point motion estimation made easy. *ICPR 2006*, pp. 630–633.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[17] B. Micusik and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. *CVPR 2003*, pp. 485–490.

[18] B. Micusik and T. Pajdla. Structure from motion with wide circular field of view cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(7):1135–1149, July 2006.

[19] D. Nister. An efficient solution to the five-point relative pose. *IEEE PAMI*, 26(6):756–770, 2004.

[20] R. Steele and C. Jaynes. Overconstrained linear estimation of radial distortion and multi-view geometry. *ECCV 2006*, pp. 253–264.

[21] G. Stein. Lens distortion calibration using point correspondences. *CVPR 1997*, pp. 600:602.

[22] H. Stewenius, C. Engels, and D. Nister. Recent developments on direct relative orientation. *ISPRS J. of Photogrammetry and Remote Sensing*, 60:284–294, 2006.

[23] H. Stewenius, D. Nister, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *CVPR 2005*, pp. 789–794.

[24] H. Stewenius, D. Nister, M. Oskarsson, and K. Astrom. Solutions to minimal generalized relative pose problems. *OMNIVIS 2005*.

[25] S. Thirthala and M. Pollefeys. The radial trifocal tensor: A tool for calibrating the radial distortion of wide-angle cameras. *CVPR 2005* pp. 321–328.

[26] C. Traverso. Gröbner trace algorithms. In *ISSAC*, pages 125–138, 1988.

[27] Z. Zhang. On the epipolar geometry between two images with lens distortion. In *ICPR 1996*.