Fast and Robust Numerical Solutions to Minimal Problems for Cameras with Radial Distortion

Martin Byröd,
Lund University,
Lund, Sweden,
byrod@maths.lth.se

Zuzana Kukelova,
Czech Technical University in Prague,
Prague, Czech Rep.,
kukelova@cmp.felk.cvut.cz

Klas Josephson,
Lund University,
Lund, Sweden,
klasj@maths.lth.se

Tomas Pajdla,
Czech Technical University in Prague,
Prague, Czech Rep.,
pajdla@cmp.felk.cvut.cz

Kalle Åström,
Lund University,
Lund Sweden,
kalle@maths.lth.se

Abstract

A number of minimal problems of structure from motion for cameras with radial distortion have recently been studied and solved in some cases. These problems are known to be numerically very challenging and in several cases there exist no known practical algorithm yielding solutions in floating point arithmetic. We make some crucial observations concerning the floating point implementation of Gröbner basis computations and use these new insights to formulate fast and stable algorithms for two minimal problems with radial distortion previously solved in exact rational arithmetic only: (i) simultaneous estimation of essential matrix and a common radial distortion parameter for two partially calibrated views and six image point correspondences and (ii) estimation of fundamental matrix and two different radial distortion parameters for two uncalibrated views and nine image point correspondences. We demonstrate on simulated and real experiments that these two problems can be efficiently solved in floating point arithmetic. [1]

1. Introduction

Estimating camera motion and inner calibration parameters from sequences of images is a challenging computer vision problem with a broad range of applications [11]. Typically one starts with a noisy set of tentative image point correspondences. The first step then is to make decisions about inliers and outliers and get a good initial estimate to be able to deploy a more sophisticated optimization algorithm on the set of all inliers.

Two robust and widely used techniques for this purpose are RANSAC [8] and kernel voting [14], both relying on solving a large number of instances of the underlying problem, each with a small number of point correspondences. There is thus a need to develop fast and stable algorithms for solving geometric vision problems with a minimal number of points. Typically this amounts to solving a system of polynomial equations in several variables. These problems are known to be numerically very challenging and in several cases there exist no known practical algorithm yielding solutions in floating point arithmetic.

Traditionally, minimal problems have been formulated assuming a linear pin-hole camera model with different restrictions on the inner calibration parameters etc. However, for some cameras such fish-eye lenses this can be insufficient and one might need to handle strong radial distortions already from the outset.

Solving for the fundamental matrix under radial distortion was first studied in [2], where a *non-minimal* algorithm based on 15 point correspondences was given for a pair of uncalibrated cameras. More recently, in [12, 13], a number of different *minimal* problems with radial distortion have been studied and practical solutions were given in some cases.

The state-of-the-art method for solving polynomial equations is based on calculations with Gröbner bases [16] and has many applications in computer vision, but also in other fields such as cryptology [7] and robotics[1]. In [19, 3] Gröbner bases were used to derive a fast algo-

[1] This work has been funded by the Swedish Research Council through grant no. 2005-3230 'Geometry of multi-camera systems' and grant no. 2004-4579 'Image-Based Localization and Recognition of Scenes'. Moreover, this work has been supported by grants EU FP6-IST-027787 DIRAC, MSM6840770038 DMCM III, STINT Dur IG2003-2 062 and MSMT KONTAKT 9-06-17.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

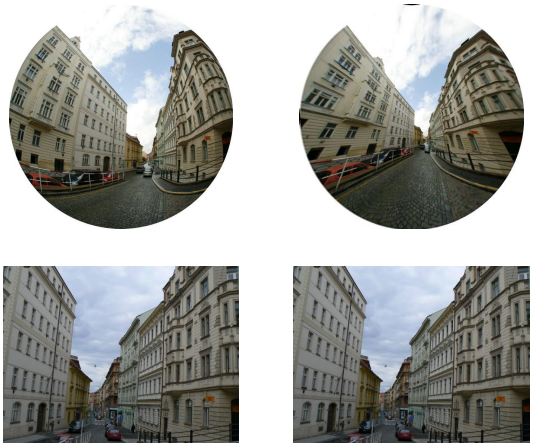

Figure 1. (Left) Input images with different radial distortions (Top) 66% cutout from omnidirectional image and (Bottom) image taken with standard perspective camera. (Right) Corrected images.

rithm for globally optimal three view triangulation under the L_2-norm.

In this paper, we further develop the techniques of numerical Gröbner basis computations. In particular we (i) note the importance of obtaining a single elimination step in the Gröbner basis computation, (ii) give guidelines for how this can be achieved and (iii) give a new simplified formulation of the Gröbner basis computation procedure based on LU factorization, which reduces the computational burden of the elimination step.

Leveraging on these new insights, we formulate fast and numerically stable algorithms for two minimal problems with radial distortion previously unsolved in floating point arithmetic: (i) simultaneous estimation of essential matrix and a common radial distortion parameter for two partially calibrated views and six image point correspondences and (ii) estimation of fundamental matrix and two different radial distortion parameters for two uncalibrated views and nine image point correspondences.

We demonstrate the speed and intrinsic numerical stability as well as robustness to noise of the proposed algorithms using both synthetic data and real images.

2. Review of Gröbner Basis Techniques for Polynomial Equation Solving

Solving systems of polynomial equations is a challenging problem in many respects and there exist no practical numerically stable algorithms for the general case. Instead, special purpose algorithms need to be developed for specific applications. The state-of-the-art tool for doing this is calculations with Gröbner bases.

Our general goal is to find the complete set of solutions to a system

$$f_1(\mathbf{x}) = 0, \ldots, f_m(\mathbf{x}) = 0, \tag{1}$$

of m polynomial equations in n variables $\mathbf{x} = (x_1, \ldots, x_n)$. The polynomials $f_1, \ldots, f_m$ generate an *ideal* I in $\mathbb{C}[\mathbf{x}]$, the ring of multivariate polynomials in $\mathbf{x}$ over the field of complex numbers defined as the set

$$I = \{g(\mathbf{x}) : g(\mathbf{x}) = \Sigma_k h_k(\mathbf{x}) f_k(\mathbf{x})\}, \tag{2}$$

where the $h_k(\mathbf{x})$ are any polynomials.

The Gröbner basis method for equation solving essentially builds on a generalization of polynomial division to the multivariate case. A concept arising in multivariate polynomial division which does not exist in the univariate case is division by a *set* of polynomials. See [4] for details. Division by an ideal as given by (2) can then be defined as division by the set of generators f_k.

The starting point now is to consider the space of all possible remainders under division by I. This space is denoted $\mathbb{C}[\mathbf{x}]/I$ and referred to as the *quotient space*. It can be seen as a generalization of the modulo rings $\mathbb{Z}_n$ to polynomials. A famous result from algebraic geometry now states that if the set of equations (1) has a finite set of zeros, then $\mathbb{C}[\mathbf{x}]/I$ will be a finite-dimensional linear space with dimension equal to the number of zeros of (1) [4].

With the space $\mathbb{C}[\mathbf{x}]/I$ in hand an elegant trick now yields the solutions to (1). Consider multiplication by one of the variables x_k. This is a linear mapping from $\mathbb{C}[\mathbf{x}]/I$ to itself and since we are in a finite-dimensional space, by selecting an appropriate linear basis, this mapping can be represented as a matrix $\mathbf{m}_{x_k}$. This matrix is known as the *action matrix* and the eigenvalues of $\mathbf{m}_{x_k}$ are exactly the values of x_k on the zeros of (1) [4]. Furthermore, the eigenvectors of $\mathbf{m}_{x_k}^T$ correspond to the vector of monomials evaluated at the zeros of (1).

The crucial step in the process is to compute the remainder arithmetic of $\mathbb{C}[\mathbf{x}]/I$. Multivariate polynomial division by I is complicated by the fact that it is not well defined for most choices of generators. Consider the operator $\mathbf{P} : \mathbb{C}[\mathbf{x}] \rightarrow \mathbb{C}[\mathbf{x}]/I$ representing division by I for some choice of generators. For $\mathbf{P}$ to be well defined we require that $\mathbf{P}(f_1(\mathbf{x}) + f_2(\mathbf{x})) = \mathbf{P}f_1(\mathbf{x}) + \mathbf{P}f_2(\mathbf{x})$ for all $f_1(\mathbf{x}), f_2(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$.

Fortunately there exist a canonical choice of generators for which $\mathbf{P}$ is well defined. This set of generators of I is known as the Gröbner basis of I and allows direct construction of the action matrix, see [3] for details. Calculating the Gröbner basis of I is therefore our main concern. In general, this is accomplished by Buchberger's algorithm which works well in exact arithmetic. However, in floating point arithmetic it very easily becomes unstable. There exist some attempts to remedy this [5, 6], but for more difficult

cases the only reliable approach (so far) is to study a particular class of equations (*e.g*. relative orientation for calibrated cameras [17], optimal three view triangulation [19], etc.) and use knowledge of what the structure of the Gröbner basis should be to design a special purpose Gröbner basis solver. This art has been developed by Stewenius and others in a number of papers [16, 12, 18]. In the following section we outline how this is done and provide new insights enabling us to solve the two problems with radial distortion treated in this paper.

3. A Matrix Version of Buchberger's Algorithm

The reason why Buchberger's algorithm breaks down in floating point arithmetic is that eliminations of monomials are performed successively and this causes round-off errors to accumulate to the point where it is completely impossible to tell whether a certain coefficient should be zero or not. The trick introduced by Faugere [5] is to write the list of equations on matrix form

$$\mathbf{C}\begin{bmatrix}\mathbf{x}^{\alpha_1}\\ \vdots \\ \mathbf{x}^{\alpha_n}\end{bmatrix} = 0. \tag{3}$$

where $\begin{bmatrix}\mathbf{x}^{\alpha_1} & \cdots & \mathbf{x}^{\alpha_n}\end{bmatrix}^T$ is a vector of monomials with the notation $\mathbf{x}^{\alpha_k} = x_1^{\alpha_{k1}} \cdots x_p^{\alpha_{kp}}$. Elimination of leading terms now translates to matrix operations and we then have access to a whole battery of techniques from numerical linear algebra allowing us to perform many eliminations at the same time with control on pivoting etc.

However, as mentioned above, the real power of this approach is brought out by combining it with knowledge about a specific problem obtained in advance with a computer algebra system such as Macaulay2 [10]. One can then get information about exactly which monomials occur in Buchberger's algorithm and the dimension of $\mathbb{C}[\mathbf{x}]/I$.

3.1. Obtaining a Single Elimination Step

With knowledge of the particular problem at hand, the ideal situation is to obtain a single big elimination step. The reason for this is that each elimination step can be ill conditioned and with errors accumulating the situation soon becomes hopeless. With a single elimination step we get maximal control over row pivoting etc. Moreover, the basis selection method introduced in [3] can further improve stability, but is only applicable when a single elimination step is possible.

In Buchberger's Algorithm, two polynomials are picked and the least common multiple of their leading terms is eliminated by multiplying them with the right monomials and then subtracting them. This is done a large number of times until convergence. We mimic this process but aim at completely separating multiplication by monomials and elimination. The steps are

1. Multiply the original set of equations with a large number of monomials yielding an expanded set of equations.
2. Stack the coefficients of these equations in an expanded coefficient matrix $\mathbf{C}_{\text{exp}}$.
3. If enough new equations were generated in the previous step, row operations on $\mathbf{C}_{\text{exp}}$ yield the elements of the Gröbner basis we need.

An important observation made independently in [3] and [12] is that not all elements of the Gröbner basis are needed. Let $\mathcal{B}$ denote a selection of basis monomials for $\mathbb{C}[\mathbf{x}]/I$. Then to construct the action matrix $\mathbf{m}_{x_k}$ we only need to calculate the elements of the ideal I with leading monomials in the set $x_k \cdot \mathcal{B} \setminus \mathcal{B}$.

Let $\mathcal{M}$ denote the complete set of monomials and let $\mathcal{R} = x_k \cdot \mathcal{B} \setminus \mathcal{B}$ denote the set of monomials that need to be reduced to $\mathbb{C}[\mathbf{x}]/I$. Finally, let $\mathcal{E}$ ($\mathcal{E}$ for excessive) denote the remaining monomials. We then have a partitioning of the monomials as $\mathcal{M} = \mathcal{E} \bigcup \mathcal{R} \bigcup \mathcal{B}$.

Now, reorder the columns of $\mathbf{C}_{\text{exp}}$ to reflect this

$$\begin{bmatrix}\mathbf{C}_{\mathcal{E}} & \mathbf{C}_{\mathcal{R}} & \mathbf{C}_{\mathcal{B}}\end{bmatrix}. \tag{4}$$

and analogously rearrange and partition the vector of monomials as $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{X}_{\mathcal{E}}^T & \boldsymbol{X}_{\mathcal{R}}^T & \boldsymbol{X}_{\mathcal{B}}^T\end{bmatrix}^T$. The $\mathcal{E}$-monomials are not in the basis and do not need to be reduced so we eliminate them performing an LU factorization of $\mathbf{C}_{\text{exp}}$ yielding the following schematic result:

$$\begin{bmatrix}\mathbf{U}_{\mathcal{E}_1} & \mathbf{C}_{\mathcal{R}_1} & \mathbf{C}_{\mathcal{B}_1} \\ 0 & \mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{B}_2}\end{bmatrix}, \tag{5}$$

where $\mathbf{U}_{\mathcal{E}_1}$ and $\mathbf{U}_{\mathcal{R}_2}$ are upper triangular. We can now discard the top rows of the coefficient matrix producing

$$\begin{bmatrix}\mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{B}_2}\end{bmatrix} \tag{6}$$

From this we see that if the sub matrix $\mathbf{U}_{\mathcal{R}_2}$ is of full rank we get precisely the polynomials from the ideal I we need by forming

$$\begin{bmatrix}\mathbf{I} & \mathbf{U}_{\mathcal{R}_2}^{-1}\mathbf{C}_{\mathcal{B}_2}\end{bmatrix}. \tag{7}$$

This is because (7) represents the equations

$$\begin{bmatrix}\mathbf{I} & \mathbf{U}_{\mathcal{R}_2}^{-1}\mathbf{C}_{\mathcal{B}_2}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}_{\mathcal{R}} \\ \boldsymbol{X}_{\mathcal{B}}\end{bmatrix} = 0 \tag{8}$$

or equivalently

$$\boldsymbol{X}_{\mathcal{R}} = -\mathbf{U}_{\mathcal{R}_2}^{-1}\mathbf{C}_{\mathcal{B}_2}\boldsymbol{X}_{\mathcal{B}}, \tag{9}$$

which means that the $\mathcal{R}$-monomials can now be expressed uniquely in terms of the $\mathcal{B}$-monomials. This is precisely what we need to compute the action matrix $\mathbf{m}_{x_k}$ in $\mathbb{C}[\mathbf{x}]/I$. In other words, the property of $\mathbf{U}_{\mathcal{R}_2}$ as being of full rank is sufficient to get the part of the remainder arithmetic of $\mathbb{C}[\mathbf{x}]/I$ that we need to compute $\mathbf{m}_{x_k}$.

4. Application to Minimal Problems with Radial Distortion

Based on the techniques described in the previous section, we are now able to provide fast and stable algorithms for two previously intractable minimal problems with radial distortion:

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

These two problems were previously studied in [13] and found to be numerically very challenging. In [13] the authors provide solutions to these problems computed in exact rational arithmetic only. This results in very long computational times and is not usable in practical applications. Here we show that these two problems can be efficiently solved in floating point arithmetic.

4.1. Uncalibrated Case

In our solution we use the same formulation of the problem as in [13]. This formulation assume a one-parameter division model [9] given by the formula

$$\mathbf{p}_u \sim \mathbf{p}_d/(1 + \lambda r_d^2) \tag{10}$$

$\mathbf{p}_u = (x_u, y_u, 1)$, 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.

It is known that to get solutions to this minimal problem for uncalibrated cameras 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

$$\mathbf{p}_{u_i}^{\top}(\lambda_1)\,\mathtt{F}\,\mathbf{p}'_{u_i}(\lambda_2) = 0, \quad i = 1, \ldots, 9 \tag{11}$$

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

$$\det(\mathtt{F}) = 0. \tag{12}$$

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

By linear elimination, these 10 equations can be reduced to 4 equations in 4 unknowns (one of 2^{nd} degree, two of 3^{rd} degree and one of 5^{th} degree). For more details see [13] where it was shown that this problem has 24 solutions.

The numerical solver is constructed starting with the four remaining equations in the four unknowns $f_{3,1}$, $f_{3,2}$, λ_1 and λ_2. The first step is to expand the number of equations, as outlined in Section 3, by multiplying them by a handcrafted set of monomials in the four unknowns yielding 393 equations in 390 monomials. See Section 4.1.1 for details.

We now stack the coefficients of the equations in a matrix $\mathbf{C}$ as in (3). Following this, we order the monomials as in (4). The sets $\mathcal{E}$ and $\mathcal{R}$ depend on which variable is used to create the action matrix. For this problem $f_{3,1}$ was used as action variable. The classical method is thereafter to choose the linear basis $\mathcal{B}$ of $\mathbb{C}[\mathbf{x}]/I$ to be the 24 lowest monomials (w.r.t. some monomial order). This is enough to get a solution to the problem, but as mentioned in Section 3 we can use the method introduced in [3] to select a basis of linear combinations of monomials from a larger set and thereby improve numerical stability. Empirically, we have found that the linear basis can be selected from the set of all monomials up to degree four excluding the monomial λ_1^4. The set $\mathcal{R}$ then consists of monomials of degree five that are reached when the monomials of degree four are multiplied with $f_{3,1}$. $\mathcal{E}$ is the remaining set of 285 monomials.

Putting the part of $\mathbf{C}$ corresponding to $\mathcal{E}$ and $\mathcal{R}$ on triangular form by means of an LU decomposition now produces what is illustrated in (5). We can then remove all equations that include excessive monomials and still have enough information to construct the action matrix.

Finally, we make the choice of representatives for $\mathbb{C}[\mathbf{x}]/I$ by the method in [3] and do the last elimination to get the part of the Gröbner basis we need to construct the action matrix.

4.1.1 Details on the Expansion Step for the Uncalibrated Case

We have found in experiments that to construct the necessary elements of the Gröbner basis, we need to generate polynomials up to total degree eight. Thus, the 2^{nd} degree polynomial has to be multiplied with all monomials up to degree six and corresponding numbers for the 3^{rd} and 5^{th} degree polynomials.

Further investigations has shown that not exactly all monomials up to degree eight are needed, so in the implementation, the 2^{nd} degree polynomial was only multiplied with monomials up to degree five and each variable not higher than four, further on was λ_1 not multiplied with higher degree than two. For the other polynomials it was possible to limit the degree of each individual variable to one lower than the total degree.

These multiplications yield 393 equations in 390 monomials. Without the last fine tuning of the degrees, the number of equations and monomials will be larger but all extra monomials will be in the set $\mathcal{E}$ and will make no real differences to the solver except slightly longer computation times.

4.2. Calibrated Case

To solve the minimal problem for calibrated cameras, we make use of the epipolar constraint for 6 point correspondences

$$\mathbf{p}_{u_i}^{\top}(\lambda)\,\mathtt{E}\,\mathbf{p}'_{u_i}(\lambda) = 0, \quad i = 1,\ldots,6, \tag{13}$$

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

$$\det(\mathtt{E}) = 0 \tag{14}$$

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

$$2\left(\mathtt{E}\mathtt{E}^T\right)\mathtt{E} - trace(\mathtt{E}\mathtt{E}^T)\mathtt{E} = 0. \tag{15}$$

Again assuming $e_{3,3} \neq 0$, we can set $e_{3,3} = 1$ and obtain 16 equations in 9 unknowns. Using a similar method as for the uncalibrated case, these equations can be rewritten as 11 polynomial equations in 4 unknowns (one of 3^{rd} degree, four of 5^{th} degree and six of 6^{th} degree). In [13] it was shown that this problem has 52 solutions.

The numerical solution of this problem largely follows that of the uncalibrated version. In the first expansion, all equations are multiplied with monomials to reach degree eight. This gives 356 equations in 378 monomials. As in the uncalibrated case it is possible to reduce the number of monomials by fine tuning the degrees we need to go to, in this case yielding 320 equations in 363 monomials.

The next step is to reorder the monomials as in equation (4). Once again, the linear basis of $\mathbb{C}[\mathbf{x}]/I$ can be constructed from the monomials of degree four and lower. $\mathcal{R}$ will then consist of those monomials of degree five that are reached when the degree four monomials are multiplied with the variable $e_{3,1}$, which is used as action variable.

As before $\mathbf{C}$ is transformed to triangular form by LU decomposition and after that we only consider those equations that do not include any of the monomials in $\mathcal{E}$. Now $\mathbf{C}$ holds all necessary information to choose representatives in $\mathbb{C}[\mathbf{x}]/I$ by the method of [3] and create the action matrix with respect to multiplication by $e_{3,1}$.

5. Experiments

We have tested the algorithms for the uncalibrated and calibrated minimal problems on synthetic images with various levels of noise, outliers and radial distortions as well as on real images.

Both problems are solved by finding the roots of a system of polynomial equations which means that we obtain several potentially correct answers, 52 in the calibrated case and 24 in the uncalibrated case. In general we obtain more than one real root, in which case we need to select the best one, *i.e.* 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 kernel voting [14] for several 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_1 and λ_2 in the uncalibrated case and λ in the calibrated case is found as the position of the largest peak [14, 12].

5.1. Tests on Synthetic Images

For both problems treated in this paper, the same synthetic experiments were carried out to evaluate the quality of the solvers.

In all our simulated experiments we generate our synthetic data using the following procedure:

1. Generate a 3D scene consisting of 1000 points distributed randomly 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, true as well as false matches.
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.

Uncalibrated case

In the first two experiments we study the robustness of our algorithm for the uncalibrated case to Gaussian noise added to the distorted points.

The first experiment investigates the estimation error of λ as a function of noise. The ground truth radial distortions parameters were $\lambda_1 = -0.2$, $\lambda_2 = -0.3$ in the first case and $\lambda_1 = -0.01$, $\lambda_2 = -0.7$ in the second case, see Figure 2. The noise varied from 0 to 2 pixels. For each noise level relative errors for 2000 λ's (estimated as closest values to the ground truth value from all solutions) were computed. The results in Figure 2 for the estimated $\overline{\lambda_1}$ (Left) and $\overline{\lambda_2}$ (Right) are presented by the Matlab function *boxplot* which shows values 25% to 75% quantile as a blue box with red horizontal line at median. The red crosses show data beyond 1.5 times the interquartile range.

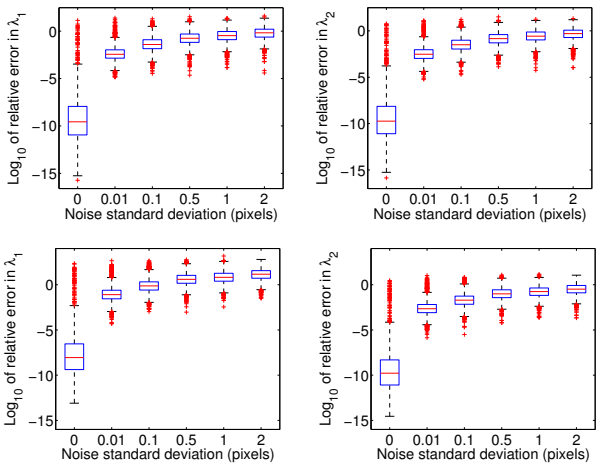

Figure 2. Uncalibrated case: Relative errors of (Left) $\overline{\lambda_1}$ and (Right) $\overline{\lambda_2}$ as a function of noise. Ground truth (Top) $\lambda_1 = -0.2$, $\lambda_2 = -0.3$ and (Bottom) $\lambda_1 = -0.01$, $\lambda_2 = -0.7$. Blue boxes contain values from 25% to 75% quantile.

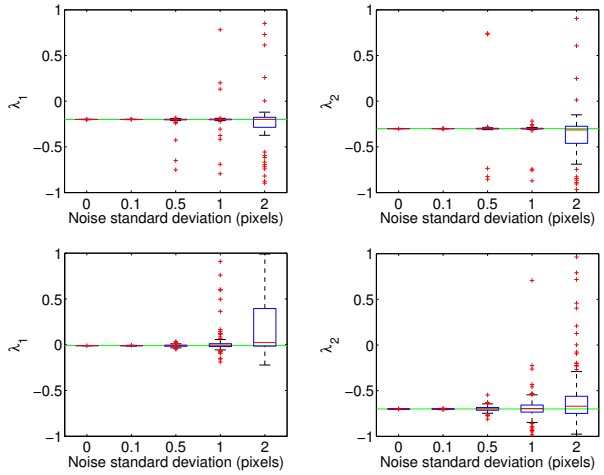

Figure 3. Uncalibrated case, kernel voting: Estimated (Left) $\overline{\lambda_1}$ and (Right) $\overline{\lambda_2}$ as a function of noise, (Top) ground truth $\lambda_1 = -0.2$, $\lambda_2 = -0.3$ (green lines), 90% of inliers and 100 samples in kernel voting and (Bottom) ground truth $\lambda_1 = -0.01$, $\lambda_2 = -0.7$, 100% of inliers and 50 samples in kernel voting.

For noiseless data we obtain very accurate estimates of radial distortion parameters even for very different λ's. For larger noises the $\log_{10}$ relative errors are much higher (mostly around 10^{-1}). However obtained λ's are still satisfactory and mostly differ from the ground truth value in the second decimal place. The main point though is not to use a minimal point set to get a good estimate, but to repeatedly draw minimal configurations from a larger set of potential matches and then use *e.g.* kernel voting to get a more reliable estimate. Finally, the result can be further enhanced using the obtained estimate as a good starting guess in a large scale bundle adjustment. The effect of kernel voting is studied in the second experiment.

In this experiment we did not select the root closest to the ground truth value for each run of the algorithm, instead we used kernel voting to select the best λ's among all generated roots from several runs. The ground truth radial distortion parameters were as in the previous experiment ($\lambda_1 = -0.2$, $\lambda_2 = -0.3$ in the first case and $\lambda_1 = -0.01$, $\lambda_2 = -0.7$ in the second case) and the level of noise varied from 0 to 2 pixels. Moreover, in the first case there were 10% of outliers in the image (M=90).

The testing procedure was as follows:

1. Repeat K times (We use K from 50 to 100 though for more noisy data K from 100 to 200 gives better results).
   (a) Randomly choose 9 point correspondences from a set of N potential correspondences (6 point correspondences for the calibrated case).
   (b) Normalize image point coordinates to $[-1, 1]$.
   (c) Find 24 roots using our algorithm.
   (d) Select the real roots in the feasible interval, *i.e.* $-1 < \lambda_1, \lambda_2 < 1$ and the corresponding F's.
2. Use kernel voting to select the best root.

Figure 3 shows λ's computed using our algorithm for the uncalibrated case as a function of noise. In the first case with outliers Figure 3 (Top) 100 λ's were estimated using kernel voting for roots computed from 100 ($K = 100$) 9-tuples of correspondences randomly drawn for each noise level. In the second case Figure 3 (Bottom) 200 λ's were estimated using kernel voting for roots computed from 50 ($K = 50$) 9-tuples of correspondences. This means that for each noise level our algorithm ran 10,000 times in both cases. The results are again presented by the Matlab function *boxplot*.

The median values for $\overline{\lambda_1}$ and $\overline{\lambda_2}$ are very close to the ground truth value for all noise levels from 0 to 2 pixels and also for very different radial distortion parameters Figure 2 (Bottom) and 10% of outliers Figure 2 (Top).

Calibrated case

The same synthetic experiments were carried out for the calibrated solver.

The results of the first experiment which shows relative errors of the estimated $\overline{\lambda}$ as a function of noise are shown in Figure 4. The ground truth radial distortion was $\lambda = -0.3$. For noiseless data we again obtain very precise estimates of radial distortion parameter λ. For larger noise levels the $\log_{10}$ relative errors are slightly larger than for the uncalibrated case. However, using kernel voting we can still ob-

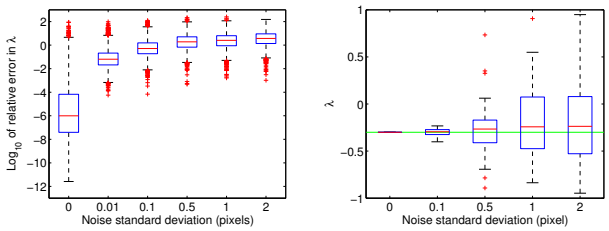

Figure 4. Calibrated case: (Left) relative errors of $\overline{\lambda}$ as a function of noise, ground truth $\lambda = -0.3$. (Right) kernel voting: Estimated $\overline{\lambda}$ using kernel voting for roots computed from 200 6-tuples of correspondences randomly drawn for each noise level. Ground truth $\lambda = -0.3$ (green line).

tain good estimates. This is shown by our second experiment.

In this experiment λ was estimated 50 times using kernel voting for roots computed from 200 6-tuples of correspondences randomly drawn for each noise level, Figure 4. The median values for $\overline{\lambda}$ are again very close to the ground truth value $\lambda = -0.3$ for all noise levels from 0 to 2 pixels. However the variances of this for the calibrated case are larger, especially for higher noise levels, than the variances for the uncalibrated case. This means that for good estimates of λ this algorithm requires more samples in the kernel voting procedure than in the uncalibrated case.

5.2. Time Consumption

To evaluate the speed of the new algorithm a reasonably optimized version of the algorithm for the uncalibrated case was implemented. The implementation was done in Matlab so rewriting the algorithm in a compiled language such as C should reduce the execution time further.

The algorithm was run 10,000 times and the time consumption was measured using the Matlab profiler. The experiments were performed on an Intel Core 2 CPU 2.13 GHz machine with 2 GB of memory. The estimated average execution time for solving one instance of the uncalibrated problem was 16 milliseconds. The corresponding time for the calibrated case was 17 milliseconds.

These results are to be compared with the execution times given for the same problem in [13], where solutions were computed in exact rational arithmetic. There, the processing time for one problem instance was 30 s for the uncalibrated case and 1700 s for the calibrated case.

5.3. Tests on Real Images

We have tested our algorithm for uncalibrated cameras with different radial distortions on several different sets of images. In the first experiment the input images with different relatively large distortions in each image, Figure 5 (Left), were obtained as 60% cutouts from fish-eye images taken with two different cameras with different radial distortions.

Figure 5. Real data, 60% cutouts from omnidirectional images. (Left) Input images with different radial distortions for camera 1 (Top) and camera 2 (Bottom). (Right) Corrected images.

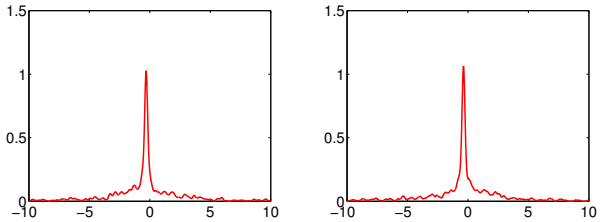

Figure 6. Distribution of real roots obtained by kernel voting for images in Figure 5. Estimated $\lambda_1 = -0.301250$ and $\lambda_2 = -0.368125$.

Tentative point matches were then found by the wide baseline matching algorithm [15]. They contained correct as well as incorrect matches. Distortion parameters λ_1 and λ_2 were estimated using our algorithm for uncalibrated cameras with different radial distortions and the kernel voting method for 100 samples. The input (Left) and corrected (Right) images are presented in Figure 5. Figure 6 shows the distribution of real roots for images from Figure 5, from which $\lambda_1 = -0.301250$ and $\lambda_2 = -0.368125$ were estimated as the argument of the maximum. The peaks from kernel voting are sharp and the λ's are estimated accurately.

In the second experiment we tested our algorithm on images with significantly different distortions. The left image Figure 1 (Left), was obtained as a 66% cutout from a fish-eye image and the right image was taken with a standard perspective camera. Since these images had a rather large difference in radial distortion, the tentative point correspondences contained a larger number of mismatches. Distortion parameters λ_1 and λ_2 were again estimated using our algorithm for uncalibrated cameras with different radial distortions and the kernel voting method. The input (Left) and

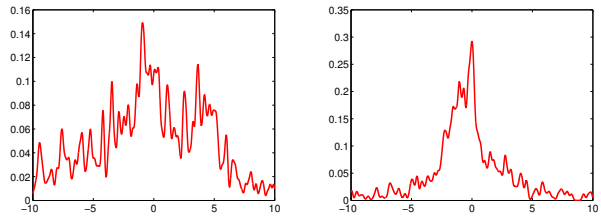

Figure 7. Distribution of real roots obtained by kernel voting for images in Figure 1. Estimated $\lambda_1 = -0.925625$ and $\lambda_2 = 0.002500$.

corrected (Right) images are presented in Figure 1. Figure 7 shows the distribution of real roots for these images from which $\lambda_1 = -0.925625$ and $\lambda_2 = 0.002500$ were estimated. As can be seen the peaks obtained by kernel voting are not so sharp but still sufficient to get good estimates of the λ's even from only 100 samples.

6. Conclusions

In this paper we have given fast and robust algorithms for two minimal problems previously unsolved in floating point arithmetic. The two problems of simultaneously solving for relative pose and radial distortion were, due to numerical problems, previously solved in exact rational arithmetic only, yielding them to time consuming to be of practical value. With the floating point algorithm presented in this paper we have reduced the computation time from minutes to milliseconds. Moreover, we have verified that this is done without loss of numerical precision by extensive experiments both on synthetic and real images.

In the experiments we have also demonstrated that the radial distortion estimation is robust both to outliers and noise when kernel voting is used over several runs. Finally we have shown that large differences in distortion between the two images can be handled.

References

[1] A. Almadi, A. Dhingra, and D. Kohli. A gröbner-sylvester hybrid method for closed-form displacement analysis of mechanisms. *Journal of Mechanical Design*, 122(4):431 – 438, 12 2000.

[2] J. Barreto and K. Daniilidis. Fundamental matrix for cameras with radial distortion. In *IEEE International Conference on Computer Vision*, Beijing, China, 2005.

[3] M. Byröd, K. Josephson, and K. Åström. Improving numerical accuracy of gröbner basis polynomial equation solver. In *International Conference on Computer Vision*, 2007.

[4] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, 2007.

[5] J.-C. Faugère. A new efficient algorithm for computing grobner bases (f4). *Journal of pure and applied algebra*, 139:61–88, 6 1999.

[6] J.-C. Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In *ISSAC '02: Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, New York, NY, USA, 2002. ACM Press.

[7] J.-C. Faugère and A. Joux. Algebraic cryptanalysis of hidden field equation (hfe) cryptosystems using gröbner bases. In *CRYPTO 2003*, pages 44–60, 2003.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–95, 1981.

[9] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of Computer Vision and Pattern Recognition Conference (CPVR)*, pages 125–132, 2001.

[10] D. Grayson and M. Stillman. Macaulay 2. Available at http://www.math.uiuc.edu/Macaulay2/, 1993-2002. An open source computer algebra software.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[12] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. In *Proceedings of Computer Vision and Pattern Recognition Conference (CPVR)*. IEEE Press, 2007.

[13] Z. Kukelova and T. Pajdla. Two minimal problems for cameras with radial distortion. In *Proceedings of The Seventh Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS)*, 2007.

[14] H. Li and R. Hartley. A non-iterative method for lens distortion correction from point matches. In *Workshop on Omnidirectional Vision*, Beijing China, Oct. 2005.

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

[16] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, Apr. 2005.

[17] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, June 2006.

[18] H. Stewénius, D. Nistér, M. Oskarsson, and K. Åström. Solutions to minimal generalized relative pose problems. In *Workshop on Omnidirectional Vision*, Beijing China, Oct. 2005.

[19] H. Stewénius, F. Schaffalitzky, and D. Nistér. How hard is three-view triangulation really? In *Proc. 10th Int. Conf. on Computer Vision*, pages 686–693, Beijing, China, 2005.