Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision

Zuzana Kukelova, *Member*, *IEEE*, Martin Bujnak, *Member*, *IEEE*, and Tomas Pajdla, *Member*, *IEEE*

Abstract—We present a method for solving systems of polynomial equations appearing in computer vision. This method is based on polynomial eigenvalue solvers and is more straightforward and easier to implement than the state-of-the-art Gröbner basis method since eigenvalue problems are well studied, easy to understand, and efficient and robust algorithms for solving these problems are available. We provide a characterization of problems that can be efficiently solved as polynomial eigenvalue problems (PEPs) and present a resultant-based method for transforming a system of polynomial equations to a polynomial eigenvalue problem. We propose techniques that can be used to reduce the size of the computed polynomial eigenvalue problems. To show the applicability of the proposed polynomial eigenvalue method, we present the polynomial eigenvalue solutions to several important minimal relative pose problems.

Index Terms—Structure from motion, relative camera pose, minimal problems, polynomial eigenvalue problems.

1 Introduction

Problems of estimating relative or absolute camera pose [12] from image correspondences can be formulated as minimal problems and solved from a minimal number of image points [9]. These minimal solutions [29], [38], [36], [9], [2], [3] are used in many applications such as 3D reconstruction [34], [33] and structure from motion since they are very effective as hypothesis generators in RANSAC paradigm [9] or can be used for initializing the bundle adjustment [12].

Many new minimal problems [36], [37], [38], [11], [15], [2], [3], [13], [18], [4] which have been solved recently lead to nontrivial systems of polynomial equations. A popular method for solving such systems is based on polynomial ideal theory and Gröbner bases [6]. The Gröbner basis method was used to solve almost all previously mentioned minimal problems including the well-known 5-pt relative pose problem [38], the 6-pt equal focal length problem [36], the 6-pt problem for one fully calibrated and one up to focal length calibrated camera [3], or the problem of estimating relative pose and one parameter radial distortion from 8-pt correspondences [15].

The Gröbner basis approach is general but not always straightforward and often cannot be easily used to create new solvers or to reimplement the existing ones. This is mainly because the existing general Gröbner basis algorithms [6] cannot be directly used to create efficient solvers for computer vision problems and therefore special algorithms for concrete problems have to be designed to achieve numerical robustness and computational efficiency. An automatic generator of Gröbner basis solvers proposed in [16] can be used as a black-box and helps considerably in constructing new solvers. However, additional expert knowledge is often necessary when dealing with more difficult problems to generate a useful solution.

In this paper, we present an alternative method for solving systems of polynomial equations appearing in computer vision based on polynomial eigenvalue solvers [1]. This method is in some sense more straightforward and easier to implement than the Gröbner basis method since eigenvalue problems are well studied, easy to understand, and efficient and robust algorithms for solving these problems [1] can be directly used to solve concrete computer vision problems.

The polynomial eigenvalue method was previously used to solve several problems in computer vision, like the problem of autocalibration of one-parameter radial distortion from 9 point correspondences [10], or to estimate paracatadioptric camera model from image matches [28].

Motivated by these examples, we provide here a characterization of problems that can be efficiently solved as polynomial eigenvalue problems (PEPs) and present a resultant-based method for transforming a system of polynomial equations to a polynomial eigenvalue problem. The resultant-based method presented in this paper is not as general as the Gröbner basis method, but is very simple and straightforward and can be applied to many systems of polynomial equations. We also propose techniques for reducing the size of polynomial eigenvalue problems and suggest how transforming of systems of polynomial equation to polynomial eigenvalue problems can be done in general.

To show the applicability of our approach, we present polynomial eigenvalue solutions to several minimal relative pose problems. We show that the 5-pt relative pose

- Z. Kukelova and T. Pajdla are with the Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague, Karlovo namesti 13, 121-35 Praha 2, Czech Republic. E-mail: {kukelova, pajdla}@cmp.felk.cvut.cz.
- M. Bujnak can be reached at Bzovicka 24, 851-07 Bratislava, Slovakia. E-mail: martin@solvergenerator.com.

Manuscript received 25 Dec. 2010; revised 10 Aug. 2011; accepted 18 Oct. 2011; published online 30 Nov. 2011.
Recommended for acceptance by A. Fitzgibbon.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference. IEEECS Log Number TPAMI-2010-12-0984.
Digital Object Identifier no. 10.1109/TPAMI.2011.230.

problem, the 6-pt equal focal length problem, the 6-pt problem for one calibrated and one up to focal length calibrated camera, and the problem of estimating relative pose and one parameter radial distortion from 8 point correspondences can be solved robustly and efficiently as polynomial eigenvalue problems. These solutions are fast and more stable than previous solutions [29], [38], [36], [15]. They are in some sense also more straightforward and easier to implement since these solvers only require collecting some coefficient matrices and then calling an existing efficient eigenvalue solver.

This paper extends [17], [3], [15], [19]. The main contribution is in proposing a resultant-based method for transforming a large class of systems of polynomial equations to a polynomial eigenvalue problem, presenting two techniques for reducing the size of this polynomial eigenvalue problem, and providing more efficient polynomial eigenvalue solutions to the problems presented in [17], [3], and [15].

The paper is organized as follows: First, we introduce the polynomial eigenvalue problems and show how they can be transformed to generalized eigenvalue problems (GEPs) and solved. Then, we provide the resultant-based method for transforming systems of polynomial equations to polynomial eigenvalue problems. Methods for reducing the size of the polynomial eigenvalue problems are described in Section 2.3. In Sections 3.1 and 3.2, we formulate the relative pose problems and summarize previous solutions to these problems. In Section 4, we provide polynomial eigenvalue solutions to these problems. The final section is dedicated to experiments. In Appendix A, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.228, we present an example of a system of polynomial equations and how can it be transformed to a polynomial eigenvalue problem.

2 Polynomial Eigenvalue Problems

Polynomial eigenvalue problems are problems of the form

$$\mathtt{C}(\lambda)\mathbf{v} = 0, \tag{1}$$

where $\mathbf{v}$ is a vector of monomials in all variables except for λ and $\mathtt{C}(\lambda)$ is a matrix polynomial in variable λ defined as

$$\mathtt{C}(\lambda) \equiv \lambda^l \mathtt{C}_l + \lambda^{l-1}\mathtt{C}_{l-1} + \cdots + \lambda \mathtt{C}_1 + \mathtt{C}_0, \tag{2}$$

with $n \times n$ coefficient matrices $\mathtt{C}_j$ [1].

We next describe how these problems can be solved by transforming them to the generalized eigenvalue problems.

2.1 Transformation to the Standard Generalized Eigenvalue Problem

Polynomial eigenvalue problems (1) can be transformed to the standard generalized eigenvalue problems:

$$\mathtt{A}\,\mathbf{y} = \lambda\,\mathtt{B}\,\mathbf{y}. \tag{3}$$

GEPs (3) are well-studied problems and there are many efficient numerical algorithms for solving them [1]. A very useful method for dense and small or moderate-sized GEPs is the QZ algorithm [1], with time complexity $O(n^3)$ and memory complexity $O(n^2)$. Different algorithms are usually used for large scale GEPs. A common approach for large scale GEPs is to reduce them to standard eigenvalue problems and then apply iterative methods.

Algorithms for solving GEPs and standard eigenvalue problems are available in almost all mathematical software and libraries, for example in LAPACK, ARPACK, or MATLAB, which provides the polyeig function for solving polynomial eigenvalue problems (1) of arbitrary degree (including their transformation to the generalized eigenvalue problems (3)).

2.1.1 Quadratic Eigenvalue Problems (QEPs)

To see how a PEP (1) can be transformed to a GEP (3) let us first consider an important class of polynomial eigenvalue problems, the quadratic eigenvalue problems of the form

$$(\lambda^2\mathtt{C}_2 + \lambda\mathtt{C}_1 + \mathtt{C}_0)\mathbf{v} = 0, \tag{4}$$

where $\mathtt{C}_2$, $\mathtt{C}_1$, and $\mathtt{C}_0$ are coefficient matrices of size $n \times n$, λ is a variable called an eigenvalue, and $\mathbf{v}$ is a vector of monomials in all variables except λ called an eigenvector.

QEP (4) can be transformed to the generalized eigenvalue problem (3) with

$$\mathtt{A} = \begin{pmatrix} 0 & \mathtt{I} \\ -\mathtt{C}_0 & -\mathtt{C}_1 \end{pmatrix}, \quad \mathtt{B} = \begin{pmatrix} \mathtt{I} & 0 \\ 0 & \mathtt{C}_2 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{v} \\ \lambda\mathbf{v} \end{pmatrix}. \tag{5}$$

Here, 0 and $\mathtt{I}$ are $n \times n$ null and identity matrices, respectively. GEP (3) with matrices (5) gives equations $\lambda\mathbf{v} = \lambda\mathbf{v}$ and $-\mathtt{C}_0\mathbf{v} - \lambda\mathtt{C}_1\mathbf{v} = \lambda^2\mathtt{C}_2\mathbf{v}$, which is equivalent to (4). Note that this GEP (5) has $2\,n$ eigenvalues and therefore by solving it we obtain $2\,n$ solutions to the QEP (4).

2.1.2 Higher Order Eigenvalue Problems

Higher order PEPs of degree l,

$$(\lambda^l\mathtt{C}_l + \lambda^{l-1}\mathtt{C}_{l-1} + \cdots + \lambda\mathtt{C}_1 + \mathtt{C}_0)\mathbf{v} = 0, \tag{6}$$

can be also transformed to the generalized eigenvalue problem (3). Here,

$$\begin{aligned} \mathtt{A} &= \begin{pmatrix} 0 & \mathtt{I} & 0 & \dots & 0 \\ 0 & 0 & \mathtt{I} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -\mathtt{C}_0 & -\mathtt{C}_1 & -\mathtt{C}_2 & \dots & -\mathtt{C}_{l-1} \end{pmatrix}, \\ \mathtt{B} &= \begin{pmatrix} \mathtt{I} & & & \\ & \dots & & \\ & & \mathtt{I} & \\ & & & \mathtt{C}_l \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{v} \\ \lambda\mathbf{v} \\ \dots \\ \lambda^{l-1}\mathbf{v} \end{pmatrix}. \end{aligned} \tag{7}$$

For higher order PEPs, one has to work with larger matrices with $n\,l$ eigenvalues. Therefore, for larger values of n and l convergence problems when solving these problems may appear [1].

Note that if the leading matrix $\mathtt{C}_l$ is nonsingular and well conditioned, then we can consider a monic matrix polynomial

$$\overline{\mathtt{C}(\lambda)} = \mathtt{C}_l^{-1}\mathtt{C}(\lambda), \tag{8}$$

with coefficient matrices $\overline{\mathtt{C}}_i = \mathtt{C}_l^{-1}\mathtt{C}_i$, $i = 0 \dots l-1$. Then, the PEP (6) can be transformed directly to the eigenvalue problem

$$\mathtt{A}\mathbf{y} = \lambda\mathbf{y}, \tag{9}$$

where

$$\mathtt{A} = \begin{pmatrix} 0 & \mathtt{I} & 0 & \dots & 0 \\ 0 & 0 & \mathtt{I} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -\overline{\mathtt{C}}_0 & -\overline{\mathtt{C}}_1 & -\overline{\mathtt{C}}_2 & \dots & -\overline{\mathtt{C}}_{l-1} \end{pmatrix}. \quad (10)$$

In this case, the matrix $\mathtt{A}$ (10) is sometimes called the block companion matrix.

It happens quite often that the leading matrix $\mathtt{C}_l$ is singular, but the last matrix $\mathtt{C}_0$ is regular and well conditioned. Then, either the described method which transforms PEP (6) to the GEP (7) or the transformation $\beta = 1/\lambda$ can be used. The transformation $\beta = 1/\lambda$ reduces the problem to finding eigenvalues of the matrix

$$\mathtt{A} = \begin{pmatrix} 0 & \mathtt{I} & 0 & \dots & 0 \\ 0 & 0 & \mathtt{I} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -\mathtt{C}_0^{-1}\mathtt{C}_l & -\mathtt{C}_0^{-1}\mathtt{C}_{l-1} & -\mathtt{C}_0^{-1}\mathtt{C}_{l-2} & \dots & -\mathtt{C}_0^{-1}\mathtt{C}_1 \end{pmatrix}. \quad (11)$$

In some cases, other linear rational transformations which improve the conditioning of the leading matrix can be used [1].

Now, we know how the polynomial eigenvalue problems (1) can be transformed to generalized (3) and classical eigenvalue problems (9) and solved using standard numerical methods. We will next describe a resultant-based method which can be used to transform a system of polynomial equations to a polynomial eigenvalue problem.

2.2 Transformation of Systems of Polynomial Equations to a PEP

Consider a system of equations

$$f_1(\mathbf{x}) = \dots = f_m(\mathbf{x}) = 0, \quad (12)$$

which is given by a set of m polynomials $\mathtt{F} = \{f_1, \dots, f_m | \ f_i \in \mathbb{C}[x_1, \dots, x_n]\}$ in n variables $\mathbf{x} = (x_1, \dots, x_n)$ over the field of complex numbers. Let this system have a finite number of solutions.

If we are lucky, like in the case of three from the four minimal relative pose problems considered in this paper, then for some x_j, let say x_1, (12) can directly be rewritten to a polynomial eigenvalue problem:

$$\mathtt{C}(x_1)\mathbf{v} = 0, \quad (13)$$

where $\mathtt{C}(x_1)$ is a matrix polynomial with square $m \times m$ coefficient matrices and $\mathbf{v}$ is a vector of s monomials in variables $x_2, \dots, x_n$, i.e., monomials of the form $\mathbf{x}^\alpha = x_2^{\alpha_2} x_3^{\alpha_3} \dots x_n^{\alpha_n}$. In this case, the number of monomials s is equal to the number of equations m, i.e., $s = m$.

Unfortunately, not all systems of polynomial equations (12) can be directly transformed to a PEP (13) for some x_j. This happens when, after rewriting the system to the form (13) we remain with fewer equations than monomials in these equations, i.e., $s > m$, and therefore we do not have square coefficient matrices $\mathtt{C}_i$. In such a case, we have to generate new equations as polynomial combinations of initial equations (12). This has to be done in a special way to get as many equations as monomials after rewriting the system to the form (13). Then, we can treat each monomial as a new variable and look at this system as at a linear one. Therefore, we sometimes say that we "linearize" our system of equations.

Here, we present one method which can be used to "linearize" a system of polynomial equations (12) and is based on multipolynomial resultants [6], [25], [26]. This method constructs resultant matrices, whose determinants express nontrivial multiples of the resultant polynomial [6] and which "linearize" a nonlinear polynomial system (12) in terms of matrix polynomials. This method takes a system of nonlinear polynomial equations (12) and reduces it to a system of the form (13). After constructing resultant matrices, we directly obtain a polynomial eigenvalue formulation of our problem and we can solve it using the methods described in Section 2.1.

Next, we describe the resultant-based method [6] for transforming a system of polynomial equations to a PEP (1). It is based on the Macaulay formulation of the resultant, which does not work for all systems of polynomial equations. Therefore, we next describe a modification of this method which enlarges its applicability and improves its numerical stability. Finally, we suggest a generalization of this method which can be applied to all systems of polynomial equations.

2.2.1 Resultant-Based Method

The resultant-based method for solving systems of polynomial equations, which reduces the problem to an eigenvalue problem, is well studied [6], [25], [26], [40]. The method was originally developed for a system of n polynomial equations in n unknowns, respectively, n homogeneous polynomial equations in $n+1$ unknowns. However, it can also be applied to $m \geq n$ general polynomial equations (12) in n unknowns.

First consider a system of n polynomial equations in n unknowns:

$$f_1(x_1, \dots, x_n) = \dots = f_n(x_1, \dots, x_n) = 0. \quad (14)$$

Assume that we want to formulate these equations as a PEP (1) for x_1. "Hide" x_1 in the coefficient field, i.e., consider these equations as n equations in $n-1$ variables $x_2, \dots, x_n$ and coefficients from $\mathbb{C}[x_1]$:

$$f_1, \dots, f_n \in (\mathbb{C}[x_1])[x_2, \dots, x_n]. \quad (15)$$

Let the degrees of these equations in variables $x_2, \dots, x_n$ be $d_1, d_2, \dots, d_n$, respectively. Now consider a system of n homogeneous polynomial equations in n unknowns:

$$\begin{gathered} F_1(x_2, \dots, x_{n+1}) = \dots = F_n(x_2, \dots, x_{n+1}) = 0, \\ F_1, \dots, F_n \in (\mathbb{C}[x_1])[x_2, \dots, x_n, x_{n+1}], \end{gathered} \quad (16)$$

which we obtain from the system (15) by homogenizing it using a new variable x_{n+1}, i.e., $F_i = x_{n+1}^{d_i} \, f_i(\frac{x_2}{x_{n+1}}, \dots, \frac{x_n}{x_{n+1}})$. Set

$$d = \sum_{i=1}^{n}(d_i - 1) + 1 = \sum_{i=1}^{n} d_i - n + 1. \quad (17)$$

For instance, when $(d_1, d_2, d_3) = (2, 2, 1)$, then $d = 3$.

Now take the set of all monomials $\mathbf{x}^\alpha = x_2^{\alpha_2} x_3^{\alpha_3} \dots x_n^{\alpha_n} x_{n+1}^{\alpha_{n+1}}$ in variables $x_2, \dots, x_{n+1}$ of total degree d, i.e., $|\alpha| = \sum_{i=2}^{n+1} \alpha_i = d$, and partition it into n subsets:

$$\begin{aligned} S_1 &= \{\mathbf{x}^\alpha : |\alpha| = d,\ x_2^{d_1} | \mathbf{x}^\alpha\}, \\ S_2 &= \{\mathbf{x}^\alpha : |\alpha| = d,\ x_2^{d_1} \not| \mathbf{x}^\alpha \text{ but } x_3^{d_2} \mid \mathbf{x}^\alpha\}, \\ &\dots \\ S_n &= \{\mathbf{x}^\alpha : |\alpha| = d,\ x_2^{d_1}, \dots, x_n^{d_{n-1}} \not|; \mathbf{x}^\alpha \text{ but } x_{n+1}^{d_n} \mid \mathbf{x}^\alpha\}, \end{aligned} \tag{18}$$

where $x_j^{d_i} \mid \mathbf{x}^\alpha$ means that $x_j^{d_i}$ divides monomial $\mathbf{x}^\alpha$ and $x_j^{d_i} \not| \mathbf{x}^\alpha$ means that it does not.

Note that every monomial of total degree d in variables $x_2, \dots, x_{n+1}$ lies in one of these sets and these sets are mutually disjoint. Moreover, if $\mathbf{x}^\alpha \in S_i$, then $x_{i+1}^{d_i} \mid \mathbf{x}^\alpha$ and $\mathbf{x}^\alpha / x_{i+1}^{d_i}$ is a monomial of total degree $d - d_i$.

We can now create a system of equations that generalizes system (16)

$$\begin{aligned} \mathbf{x}^\alpha / x_2^{d_1}\ F_1 = 0 \quad &\text{for all } \mathbf{x}^\alpha \in S_1 \\ &\dots \\ \mathbf{x}^\alpha / x_{n+1}^{d_n}\ F_n = 0 \quad &\text{for all } \mathbf{x}^\alpha \in S_n. \end{aligned} \tag{19}$$

This system has some special properties. Since F_i are homogeneous polynomials of total degree d_i, polynomials $\mathbf{x}^\alpha / x_{i+1}^{d_i}\ F_i$ are of total degree d for all $i = 1, \dots n$ and therefore can be written as a linear combination of monomials of degree d in n variables $x_2, \dots, x_{n+1}$. There exist $\binom{n+d-1}{d}$ monomials of degree d in n variables.

The total number of equations in the system (19) is equal to the number of elements in sets $S_1, \dots S_n$, which is also $\binom{n+d-1}{d}$ because these sets contain all monomials of degree d in variables $x_2, \dots x_{n+1}$. Thus, the system of polynomial equations (19) consists of $\binom{n+d-1}{d}$ homogeneous equations in $s \leq \binom{n+d-1}{d}$ monomials (all of degree d) in variables $x_2, \dots, x_{n+1}$. Note that the coefficients of these equations are polynomials in x_1.

We can next dehomogenize equations (19) by setting $x_{n+1} = 1$. This dehomogenization does not reduce the number of monomials in these equations. This is because there are no monomials of the form $\mathbf{x}^\alpha = x_2^{\alpha_2} x_3^{\alpha_3} \dots x_n^{\alpha_n} x_{n+1}^{\alpha_{n+1}}$ among monomials of total degree d which differ only in the power α_{n+1}.

Therefore, we obtain a system of $\binom{n+d-1}{d}$ equations in $s \leq \binom{n+d-1}{d}$ monomials up to degree d in $n-1$ variables $x_2, \dots, x_n$. Note that the number of monomials up to degree d in $n-1$ variables is $\sum_{k=0}^{d} \binom{n+k-2}{k} = \binom{n+d-1}{d}$, which is exactly the number of monomials of degree d in n variables.

The system obtained after dehomogenization is equivalent to the initial system (14), i.e., has the same solutions, and can be written as

$$\mathtt{C}(x_1)\mathbf{v} = 0, \tag{20}$$

where $\mathtt{C}(x_1)$ is a matrix polynomial and $\mathbf{v}$ is a vector of $s \leq \binom{n+d-1}{d}$ monomials in variables $x_2, \dots, x_n$.

For many systems of polynomial equations, we are able to choose $s \leq \binom{n+d-1}{d}$ linearly independent polynomials (including all initial polynomials (14)) from the generated $\binom{n+d-1}{d}$ polynomials (19). In that case the coefficient matrices $\mathtt{C}_\mathrm{j}$ in the matrix polynomial $\mathtt{C}(x_1)$ are square and the formulation (20) is directly a polynomial eigenvalue formulation of our system of polynomial equations (14).

Unfortunately, it may sometimes happen that between $\binom{n+d-1}{d}$ polynomials (19) there are less than s linearly independent polynomials or that, after rewriting these polynomials into the form (20), all coefficient matrices $\mathtt{C}_\mathrm{j}$ are close to singular. This may happen because the presented Macaulay resultant-based method is not designed for general polynomials but for dense homogeneous ones, i.e., for polynomials whose coefficients are all nonzero and generic, and it constructs a minimal set of equations that is necessary to obtain square matrices for these specific polynomials [6].

Therefore, we will use here a small modification of this Macaulay resultant-based method. This modified method is very simple and differs from the standard method only in the form of the sets S_i (18) and produces a higher number of polynomial equations.

2.2.2 Modified Resultant-Based Method

We again consider all monomials $\mathbf{x}^\alpha = x_2^{\alpha_2} \dots x_{n+1}^{\alpha_{n+1}}$ in variables $x_2, \dots, x_{n+1}$ of total degree d; however, here the sets S_i have the form

$$\begin{aligned} \overline{S_1} &= \{\mathbf{x}^\alpha : |\alpha| = d,\ x_2^{d_1} \mid \mathbf{x}^\alpha\}, \\ \overline{S_2} &= \{\mathbf{x}^\alpha : |\alpha| = d,\ x_3^{d_2} \mid \mathbf{x}^\alpha\}, \\ &\dots \\ \overline{S_n} &= \{\mathbf{x}^\alpha : |\alpha| = d,\ x_{n+1}^{d_n} \mid \mathbf{x}^\alpha\}. \end{aligned} \tag{21}$$

This means that in the extended set of polynomial equations:

$$\begin{aligned} \mathbf{x}^\alpha / x_2^{d_1}\ F_1 = 0 \ &\text{for all } \mathbf{x}^\alpha \in \overline{S_1}, \\ &\dots \\ \mathbf{x}^\alpha / x_{n+1}^{d_n}\ F_n = 0 \ &\text{for all } \mathbf{x}^\alpha \in \overline{S_n}, \end{aligned} \tag{22}$$

we multiply each homogeneous polynomial F_i of degree d_i, $i = 1, \dots, n$, with all monomials of degree $d - d_i$. This is because the set of monomials

$$\{\mathbf{x}^\alpha / x_{i+1}^{d_i},\ \ x_{i+1}^{d_i} \in \overline{S_i}\}, \tag{23}$$

contains all monomials of degree $d - d_i$ in variables $x_2, \dots, x_{n+1}$. Therefore, the extended system of polynomial equations (22) consists of all possible homogeneous polynomials of degree d which can be obtained from the initial homogeneous polynomials F_i (16) by multiplying them by monomials.

After creating this extended system of polynomial equations, the method continues as the previously described standard resultant-based method, i.e., we dehomogenize all equations from (22) and rewrite them to the form

$$\overline{\mathtt{C}(x_1)}\overline{\mathbf{v}} = 0, \tag{24}$$

where $\overline{\mathtt{C}(x_1)}$ is a matrix polynomial and $\overline{\mathbf{v}}$ is a vector of $\overline{s} \leq \binom{n+d-1}{d}$ monomials up to degree d in variables $x_2, \dots, x_n$.

Since in this case we generate more polynomial equations, in fact all possible of degree d which can be generated in the presented way, we are able to choose $\overline{s} \leq \binom{n+d-1}{d}$ linearly independent polynomials (including all initial polynomials (14)) more frequently from them. Moreover, when we have more than $\overline{s}$ linearly independent polynomials, we can select polynomials from them to obtain well-conditioned coefficient matrices $\mathtt{C}_j$ and we can in this way improve numerical stability of the polynomial eigenvalue formulation (24).

Of course this modified resultant-based method is still not completely general and cannot be used to transform all systems of polynomial equations to a PEP (1), but it is conceptually very simple and sufficient for many problems.

In general it can be proven that any system of polynomial equations can be transformed to a PEP (1). One way to prove this is using the Shape Lemma [6]. According to this lemma, in the ideal generated by initial polynomials [6] there exists a basis of the form $g_i = x_i + h_i(x_n)$, $i = 1, \dots, n$, where h_i are polynomials in x_n. After hiding x_n in this basis, we obtain a polynomial eigenvalue formulation of the initial problem. Using this formulation would be, however, very inefficient since for obtaining polynomials from this basis a huge number of new polynomials usually needs to be generated. Nevertheless, the existence of this basis simply shows that there exists at least one polynomial eigenvalue formulation of any system of polynomial equations with a finite number of solutions.

To obtain a more efficient polynomial eigenvalue formulation of some system of polynomial equations, it is sufficient to systematically generate all polynomials from the ideal, not only monomial multiples of the initial polynomials, like in the presented method, and stop when we already have sufficiently many polynomials for the formulation.

We can systematically generate new polynomials from the ideal, for example, by multiplying all new generated polynomials or, in the first step, all initial polynomials of degree $< d$ by all individual variables and reducing them each time by the Gauss-Jordan (G-J) elimination. After each G-J elimination we can check if we already have a polynomial eigenvalue formulation. If no new polynomials of degree $< d$ were generated and no polynomial eigenvalue formulation was obtained, then we increase the degree d. By checking if we already have a polynomial eigenvalue formulation we mean checking if there exists i polynomials containing $j < i$ monomials in the set of already generated polynomials.

In the presented approach, we were considering system (14) of n equations in n unknowns. However, this method can be easily extended to a system (12) of $m \geq n$ equations in n unknowns.

In the case of the modified resultant-based method it is sufficient to multiply each homogeneous polynomial F_i of degree d_i with all monomials of degree $d - d_i$, which is independent of the number of polynomials.

For the standard resultant-based method presented in Section 2.2.1, all what we need to do is to select n equations from the initial m equations with largest degrees d_i. Then, we can apply the presented method to these n equations in n unknowns. In this way, we will generate $\binom{n+d-1}{d}$ equations in $s \leq \binom{n+d-1}{d}$ monomials up to degree d in $n-1$ variables $x_2, \dots, x_n$ and with polynomial coefficients in x_1. Adding the remaining $m - n$ original equations may increase the number of monomials. It is because some monomials contained in these equations do not have to be contained in the generated and the selected equations. However, this number will not be greater then $\binom{n+d-1}{d}$ because the degree of these $m - n$ equations is smaller than or equal to the degree of the selected equations and therefore also smaller than d. Moreover, we can add also all multiples of these remaining equations up to degree d. In both cases, the resulting system of equations will contain only monomials up to degree d in variables $x_2, \dots, x_n$ and there are at most $\binom{n+d-1}{d}$ such monomials.

2.2.3 Problem Relaxation

Note that the polynomial eigenvalue formulation (13) is a relaxed formulation of the original problem of finding all solutions to the system of polynomial equations (12). This is because polynomial eigenvalue formulation (13) does not consider potential dependencies in the monomial vector $\mathbf{v}$ and solves for general eigenvalues x_1 and general eigenvectors $\mathbf{v}$. However, after transforming the system of polynomial equations (12) to PEP (13), coordinates of the vector $\mathbf{v}$ are in general dependent, e.g., $\mathbf{v} = (x_3^2, x_2x_3, x_2^2, x_3, x_2, 1)$, and therefore $\mathbf{v}(1) = \mathbf{v}(4)^2$, $\mathbf{v}(2) = \mathbf{v}(4)\mathbf{v}(5)$, etc.

This implies that after solving PEP (13) one has to check which of the computed eigenpairs $(x_1, \mathbf{v})$ satisfy the original polynomial equations (12). This can be done either by testing all monomial dependencies in $\mathbf{v}$ or by substituting the solutions to the original equations and checking if they are satisfied.

2.3 Reducing the Size of the Polynomial Eigevalue Problem

2.3.1 Removing Unnecessary Polynomials

The modified resultant method presented in Section 2.2.2 usually does not lead to the smallest polynomial eigenvalue formulation of the initial system of polynomial equations (12) and, for larger systems with larger degrees, it may generate large polynomial eigenvalue problems which are not practical. Therefore, we present here a method of removing unnecessary polynomials from the generated polynomial eigenvalue formulation.

Assume that we have the polynomial eigenvalue formulation (24) of the system (14) of n polynomial equations in n unknowns $x_1, \dots, x_n$, and that the vector $\overline{\mathbf{v}}$ in this polynomial eigenvalue formulation (24) consist of $\overline{s}$ monomials and the coefficient matrices $\overline{\mathtt{C_j}}$ in the matrix polynomial $\overline{\mathtt{C(x_1)}}$ have size $\overline{s} \times \overline{s}$. Then, we can remove unnecessary polynomials using the following procedure:

$i \leftarrow 1$
while $i \leq \overline{s} - n$ **do**
  if there exist i monomials $\mathbf{x}^\alpha$, between monomials of the vector $\overline{\mathbf{v}}$, which are contained only in $k \leq i$ polynomials of (24), excluding the initial polynomials, from our $\overline{s}$ polynomials, **then**
    remove these k polynomials
    $\overline{s} \leftarrow \overline{s} - k$
    $i \leftarrow 1$
  else
    $i \leftarrow i + 1$
  end if
end while

In each cycle of this algorithm, we remove more or as many monomials as polynomials from (24), or we remove nothing. Therefore, after each cycle the coefficient matrices $\overline{\mathtt{C_j}}$ are square or contain more rows than columns and can be therefore transformed to square matrices by omitting some rows. Moreover, there remain at least initial polynomial equations at the end of this algorithm. Therefore, using this procedure, we obtain a polynomial eigenvalue formulation of the initial system of polynomial equations (14) which is not larger than the starting formulation (24).

This algorithm may have higher impact when it is performed on an eliminated system of polynomial equations, i.e., a system of polynomial equations eliminated by the G-J elimination. For example, the G-J elimination may help in cases when the system contains some monomials which, after "hiding" a variable in the coefficient field, e.g., x_1, have only numerical (constant) coefficients, i.e., coefficients not containing x_1. Such monomials may then be directly eliminated, not depending on in how many polynomials they appear. This is because the application of the G-J elimination on all polynomials with suitable ordered monomials, before "hiding" x_1, may eliminate these monomials from all equations except for one equation. Therefore, such monomials will be contained in only one polynomial and will be eliminated using the presented algorithm.

2.3.2 Removing Zero Eigenvalues

It often happens that matrices $\mathtt{C}_j$ in the polynomial eigenvalue formulation (6) contain zero columns. A zero column in a matrix $\mathtt{C}_i$ means that the monomial corresponding to this column and to λ^i does not appear in the considered system of equations. Moreover, it also happens quite often that this monomial does not appear in the system for all λ^j, where $j > i$. Then, the column corresponding to this monomial is zero in all matrices $\mathtt{C}_j$ for $j > i$ including the highest order matrix $\mathtt{C}_l$, which is in this case singular. We will call such monomials which produce zero columns in matrices $\mathtt{C}_j$ "zero" monomials.

In the case of singular matrix $\mathtt{C}_l$, the transformation of the PEP to the eigenvalue problem (9) will produce matrix $\mathtt{A}$ in the form (11). This matrix will contain for each such "zero" monomial a zero column, a column in the block corresponding to $-\mathtt{C}_0^{-1}\mathtt{C}_l$. This zero column will result in a zero eigenvalue which is, in this case, not the solution to our original problem. Therefore, we can remove this column together with the corresponding row from the matrix $\mathtt{A}$ and in this way remove this zero eigenvalue.

Removing the row will remove 1 from the column corresponding to the same monomial as the "zero" monomial in the $-\mathtt{C}_0^{-1}\mathtt{C}_l$ block but in this case in the block $-\mathtt{C}_0^{-1}\mathtt{C}_{l-1}$. This means that if this "zero" monomial multiplied by λ^{l-1} does not appear in the system of polynomial equations we again have the zero column resulting in the zero eigenvalue. Therefore, we can also remove this zero column and the corresponding row of the matrix $\mathtt{A}$. This can be repeated until the first matrix $\mathtt{C}_i$ which contains, for this "zero" monomial, the zero column.

In this way, we can often remove most of the "parasitic" zero eigenvalues and therefore solve a considerably smaller eigenvalue problem, which may significantly improve the computational efficiency and the stability of the solution. This is because the eigenvalue computation is the most time-consuming part of final solvers.

3 Relative Pose Problems

To show the applicability of the previously described polynomial eigenvalue method, we present here polynomial eigenvalue solutions to four important minimal relative pose problems.

3.1 Problems Formulations

Consider a pair of cameras $\mathtt{P}$ and $\mathtt{P}'$. It is known [12] that in the case of fully calibrated cameras points x_j and x'_j, which are projections of 3D point X_j, are geometrically constrained by the epipolar geometry constraint

$$\mathbf{x}_j'^T \mathtt{E}\, \mathbf{x}_j = 0, \tag{25}$$

where $\mathtt{E}$ is a 3×3 rank-2 essential matrix with two equal singular values. These constraints on $\mathtt{E}$ can be written as

$$\det(\mathtt{E}) = 0, \tag{26}$$

$$2\, \mathtt{E}\, \mathtt{E}^\top \mathtt{E} - trace(\mathtt{E}\, \mathtt{E}^\top)\mathtt{E} = 0. \tag{27}$$

Equation (26) is called the rank constraint and (27) the trace constraint.

The usual way [29], [38] to compute the essential matrix is to linearize relation (25) into the form $\mathtt{M}\, X = 0$, where vector X contains nine elements of the matrix $\mathtt{E}$ and $\mathtt{M}$ contains products of image measurements. Essential matrix $\mathtt{E}$ is then constructed as a linear combination of the conveniently reshaped null space vectors of the matrix $\mathtt{M}$. The dimension of the null space depends on the number of point correspondences used. Additional constraints (26) and (27) are used to determine the coefficients in the linear combination of the null space vectors or to project an approximate solution to the space of correct essential matrices.

Fundamental matrix $\mathtt{F}$ describes uncalibrated cameras similarly as essential matrix $\mathtt{E}$ does for calibrated cameras (25), i.e.,

$$\mathbf{x}_j'^T \mathtt{F}\, \mathbf{x}_j = 0, \tag{28}$$

and can be computed in an analogous way.

In this paper, we also consider a camera pair with unknown but equal focal length f and a camera pair with one fully calibrated and one up to focal length calibrated camera. In both cases, all other calibration parameters are assumed to be known. In such a case the calibration matrix $\mathtt{K}$ [12] is a diagonal matrix $diag([f\ f\ 1])$.

Therefore, for two cameras with unknown, but equal focal length, the essential matrix $\mathtt{E} = \mathtt{K}^\top\, \mathtt{F}\, \mathtt{K} = \mathtt{K}\, \mathtt{F}\, \mathtt{K}$ since $\mathtt{K}$ is diagonal. Since $\mathtt{K}$ is regular, we have

$$\det(\mathtt{F}) = 0, \tag{29}$$

$$2\, \mathtt{F}\, \mathtt{Q}\mathtt{F}^\top \mathtt{Q}\mathtt{F} - trace(\mathtt{F}\, \mathtt{Q}\mathtt{F}^\top \mathtt{Q})\, \mathtt{F} = 0. \tag{30}$$

Equation (30) is obtained by substituting the expression for the essential matrix into the trace constraint (27), applying the substitution $\mathtt{Q} = \mathtt{K}\,\mathtt{K}$, and multiplying (27) by $\mathtt{K}^{-1}$ from left and right. Note that the calibration matrix can be written as $\mathtt{K} \simeq diag([1\ 1\ 1/f])$ to simplify equations.

For the case with one fully calibrated and one up to the focal length calibrated camera, we have $\mathtt{E} = \mathtt{K}\,\mathtt{F}$ and we obtain constraints

$$\det(\mathtt{F}) = 0, \tag{31}$$

$$2\,\mathtt{F}\,\mathtt{F}^\top \mathtt{Q}\mathtt{F} - trace(\mathtt{F}\mathtt{F}^\top \mathtt{Q})\,\mathtt{F} = 0. \tag{32}$$

Equation (32) is obtained similarly as (30).

Since almost all consumer camera lenses, in particular wide angle lenses, suffer from some radial distortion, we also consider here the problem of estimating relative pose of two uncalibrated cameras with one parameter radial distortion.

Here, we assume the one-parameter division model proposed by Fitzgibon [10], which is given by the formula

$$\mathbf{p}_u \sim \mathbf{p}_d/(1 + kr_d^2), \tag{33}$$

where k is the distortion parameter, $\mathbf{p}_u = [x_u, y_u, 1]^\top$, respectively, $\mathbf{p}_d = [x_d, y_d, 1]^\top$, are the corresponding undistorted, respectively, 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 is in the center of the image, i.e., $r_d^2 = x_d^2 + y_d^2$.

Since the epipolar constraint (28) contains undistorted image points $\mathbf{x}_i$, but we measure distorted ones, we need to put the relation (33) into (28). Let $\overline{\mathbf{x}_i} = [\overline{x}_i, \overline{y}_i, 1]$ be the ith measured distorted image point and let $\overline{r}_i^2 = \overline{x}_i^2 + \overline{y}_i^2$. Then, $\mathbf{x}_i = [\overline{x}_i, \overline{y}_i, 1 + k\overline{r}_i^2]$ and the epipolar constraint for the uncalibrated cameras has the form

$$[\overline{x}_i', \overline{y}_i', 1 + k\overline{r}_i^{2\prime}]\mathtt{F}\,[\overline{x}_i, \overline{y}_i, 1 + k\overline{r}_i^2]^\top = 0. \tag{34}$$

In this problem, we also have the singularity constraint on the fundamental matrix, i.e., the equation

$$\det(\mathtt{F}) = 0. \tag{35}$$

These are the basic equations which define all four relative pose problems considered in this paper and which we will use to formulate these problems as polynomial eigenvalue problems and solve them using the method presented in Section 2. Let us first review previous solutions to all the considered problems.

3.2 Previous Solutions

3.2.1 Five-Point Problem

The 5-pt calibrated relative pose problem was already studied by Kruppa [14], who has shown that it has at most 11 solutions. Maybank and Faugeras [8] then sharpened Kruppa's result by showing that there are at most 10 solutions. Recently, Nister et al. [30] have shown that the problem really requires solving a 10 degree polynomial.

The state-of-the-art methods of Nister [29] and Stewénius et al. [38], which obtain the solutions as the roots of a 10th-degree polynomial, are currently the most efficient and robust implementations for solving the 5-pt relative pose problem.

In both methods [29], [38], the five linear epipolar constraints were used to parameterize the essential matrix as a linear combination of a basis $\mathtt{E}_1, \mathtt{E}_2, \mathtt{E}_3, \mathtt{E}_4$ of the space of all compatible essential matrices:

$$\mathtt{E} = x\,\mathtt{E}_1 + y\,\mathtt{E}_2 + z\,\mathtt{E}_3 + \mathtt{E}_4. \tag{36}$$

Then, the rank constraint (26) and the trace constraint (27) were used to build 10 third-order polynomial equations in three unknowns and 20 monomials. These equations can be written in a matrix form:

$$\mathtt{M}\,X = 0, \tag{37}$$

with a coefficient matrix $\mathtt{M}$ reduced by the G-J elimination and the vector of all monomials X.

The method [29] used relations between polynomials (37) to derive three new equations. The new equations were arranged into a 3×3 matrix equation $\mathtt{A}(z)\,Z = 0$ with matrix $\mathtt{A}(z)$ containing polynomial coefficients in z and Z containing the monomials in x and y. The solutions were obtained using the hidden variable resultant method [6] by solving the 10th-degree polynomial $\det(\mathtt{A}(\mathtt{z}))$, finding Z as a solution to a homogeneous linear system, and constructing $\mathtt{E}$ from (36). Note that this hidden variable formulation is, in fact, a polynomial eigenvalue formulation, solved, however, using polynomial determinants.

The method [38] follows another classical approach to solving systems of polynomial equations. First, a Gröbner basis [6] of the ideal generated by equations (37) is found. Then, a multiplication matrix [35] is constructed. Finally, the solutions are obtained by computing the eigenvalues and the eigenvectors of the multiplication matrix [6]. This approach turned out to lead to a particularly simple procedure for the 5-pt problem since the particular Gröbner basis used and the 10×10 multiplication matrix can be constructed directly from the reduced coefficient matrix $\mathtt{M}$.

Another technique based on the hidden variable resultant for solving the 5-pt relative pose problem was proposed in [22]. This technique is somewhat easier to understand than [38], but is far less efficient and Maple was used in [22] to evaluate large determinants.

The first polynomial eigenvalue solution to this problem was proposed in [17]. This solution is similar to the solution presented in this paper; however, since the reduction of the size of the PEP was not used in [17], the solution [17] was rather inefficient and resulted in eigenvalue computation of a 30×30 matrix.

3.2.2 Six-Point Equal Focal Length Problem

The problem of estimating relative camera pose for two cameras with unknown, but equal, focal length from minimal number of 6 point correspondences has 15 solutions [36].

The first minimal solution to this problem proposed by Stewénius et al. [36] is based on the Gröbner basis techniques and is similar to the Stewénius solution to the 5-pt problem [38]. Using the linear epipolar constraints, the fundamental matrix is parameterized by two unknowns as

$$\mathtt{F} = x\,\mathtt{F}_1 + y\,\mathtt{F}_2 + \mathtt{F}_3. \tag{38}$$

Using the rank constraint for the fundamental matrix (29) and the trace constraint for the essential matrix (30) then brings 10 third and fifth-order polynomial equations in three unknowns x, y, and $w = f^{-2}$, where f is the unknown focal length.

The Gröbner basis solver [36] starts with these 10 polynomial equations, which can be represented by a 10×33 matrix $\mathtt{M}$. Since this matrix does not contain all necessary polynomials for creating a multiplication matrix, several new polynomials, monomial multiples of the initial polynomials, are added.

The resulting solver therefore consists of three G-J eliminations of three matrices of size 12×33, 16×33, and 18×33. The eigenvectors of the multiplication matrix provide the solutions to the three unknowns x, y, and $w = f^{-2}$.

Another Gröbner basis solver to this problem was proposed in [5]. This solver uses only one G-J elimination of a 34×50 matrix and uses a special technique for improving the numerical stability of Gröbner basis solvers by selecting a suitable basis of the quotient ring modulo the ideal generated by the Gröbner basis. In this paper, it was shown that this solver gives more accurate results than the original solver [36].

A Gröbner basis solver with single G-J elimination of a 31×46 matrix, which was generated using the automatic generator, was presented in [16]. This solver is also more accurate than the original solver proposed by Stewénius [36].

Yet another solution based on the hidden variable resultant method was proposed in [21], but it has similar problems with efficiency as the hidden variable solution to the 5-pt problem [22].

Like in the 5-pt relative pose problem, the first polynomial eigenvalue solution to this problem was proposed in [17] and we will discuss it in more detail in Section 4.2.

3.2.3 Six-Point One Calibrated Camera Problem

The problem of estimating relative camera pose for one fully calibrated and one up to the focal length calibrated camera from a minimal number of 6 point correspondences was solved only recently in [3].

This problem was previously studied in [39], where a nonminimal solution was proposed. First, the 7 point algorithm [12] was used for computing the fundamental matrix and then the focal length was estimated in a closed-form solution using Kruppa equations.

In [3], two new minimal solutions to this problem were proposed. The first solution is based on the Gröbner basis techniques and is similar to the Stewénius' solution to the 6-pt equal focal length problem [36]. The fundamental matrix is again parameterized by two unknowns, as in (38). Then, the rank constraint for the fundamental matrix (31) and the trace constraint for the essential matrix (32) are used. This gives 10 third and fourth-order polynomial equations in three unknowns x, y, and $w = f^{-2}$ and 20 monomials. The final solver in this case consists of one G-J elimination of a 21×30 coefficient matrix $\mathtt{M}$ which contains coefficients arising from concrete image measurements and the eigenvalue and eigenvector computation of a 9×9 multiplication matrix.

The second solution to this problem, which was presented in [3], was based on the polynomial eigenvalue formulation and we will discussed it in more detail in Section 4.3.

3.2.4 Eight-Point Radial Distortion Problem

The first nonminimal solution to the problem of simultaneous estimation of the epipolar geometry and the one parameter radial distortion division model was proposed by Fitzgibbon [10]. In this solution, the algebraic constraint $\det \mathtt{F} = 0$ on the fundamental matrix has not been used and therefore 9 point correspondences were necessary to solve the problem. Thanks to neglecting this algebraic constraint, the resulting nine equations from the epipolar constraint (34) could be directly formulated as a quadratic eigenvalue problem and quite easily solved using standard numerical methods.

Li and Hartley [20] solved the same problem from 9 point correspondences using the hidden variable resultant technique. The quality of the result was comparable to [10]; however, the hidden variable technique was considerably slower than the polynomial eigenvalue technique used in [10] since in [20] Maple was used to evaluate large determinants.

The first minimal solution to the problem of estimating epipolar geometry and one parameter division model was proposed in [15]. This solution used $\det \mathtt{F} = 0$ and therefore the minimal number of 8 point correspondences were sufficient to solve it. In this solution, the initial nine polynomial equations (28) and (35) in nine unknowns were first transformed to three polynomial equations in three unknowns. Then, these equations were solved using the Gröbner basis method [6]. The final solver consists of three G-J eliminations of the 8×22, 11×30, and 36×50 matrices and the eigenvalue computation of the 16×16 matrix.

The improved version of solver [15], which was generated using the automatic generator of Gröbner basis solvers and consists of only one elimination of the 32×48 was proposed in [16].

The polynomial eigenvalue solution to this problem was proposed in [19]. This solution is in fact equivalent to the solution presented in this paper. In Section 4.4, we will show how this solution can be easily obtained using the modified resultant-based method presented in Section 2.2.2 and in this way illustrate the usefulness of our method.

4 Polynomial Eigenvalue Solution to the Relative Pose Problems

In this section, we describe our solutions to the four relative pose problems. We show that the 5-pt relative pose problem, the 6-pt equal focal length problem, the 6-pt problem for one fully and one up to focal length calibrated camera, and the 8-pt problem for estimating relative pose and one parameter radial distortion can be formulated as the polynomial eigenvalue problems (1) of degree three, two, one, and four.

4.1 Five-Point Problem

To obtain the polynomial eigenvalue solution to the 5-pt relative pose problem, we use the same formulation as it was used in [29] and [38].

In this formulation, we first use linear equations from the epipolar constraint (25) for 5 point correspondences to parametrize the essential matrix with three unknowns x, y, and z (36). Using this parameterization, the rank (26) and the trace constraints (27) lead to 10 third-order polynomial equations in three unknowns and 20 monomials and can be written in the matrix form

$$\mathtt{M}\,X = 0, \tag{39}$$

where $\mathtt{M}$ is a 10×20 coefficient matrix and $X = (x^3, yx^2, y^2x, y^3, zx^2, zyx, zy^2, z^2x, z^2y, z^3, x^2, yx, y^2, zx, zy, z^2, x, y, z, 1)^\top$ is the vector of all monomials. There are all monomials in all three unknowns up to degree three. So we can use any unknown to play the role of λ in (1). For example, taking $\lambda = z$, these 10 equations (39) can be rewritten as

$$(z^3\mathtt{C}_3 + z^2\mathtt{C}_2 + z\,\mathtt{C}_1 + \mathtt{C}_0)\mathbf{v} = 0, \tag{40}$$

where $\mathbf{v}$ is a 10×1 vector of monomials, $\mathbf{v} = (x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1)^\top$ and $\mathtt{C}_3$, $\mathtt{C}_2$, $\mathtt{C}_1$, and $\mathtt{C}_0$ are 10×10 coefficient matrices:

$$\mathtt{C}_3 \equiv (0\;0\;0\;0\;0\;0\;0\;0\;0\;m_{10}),$$
$$\mathtt{C}_2 \equiv (0\;0\;0\;0\;0\;0\;0\;m_8\;m_9\;m_{16}),$$
$$\mathtt{C}_1 \equiv (0\;0\;0\;0\;m_5\;m_6\;m_7\;m_{14}\;m_{15}\;m_{19}),$$

and

$$\mathtt{C}_0 \equiv (m_1\;m_2\;m_3\;m_4 m_{11}\;m_{12}\;m_{13}\;m_{17}\;m_{18}\;m_{20}),$$

where m_j is the jth column from the coefficient matrix $\mathtt{M}$.

Since $\mathtt{C}_3$, $\mathtt{C}_2$, $\mathtt{C}_1$, and $\mathtt{C}_0$ are known square matrices, the formulation (40) is a cubic PEP and can be solved using standard efficient algorithms presented in Section 2.

In this case, the rank of the matrix $\mathtt{C}_3$ is only one, while the matrix $\mathtt{C}_0$ is regular. Therefore, we use the transformation $\beta = 1/z$ and reduce the cubic PEP (40) to the problem of finding the eigenvalues of the 30×30 matrix:

$$\mathtt{A} = \begin{pmatrix} 0 & \mathtt{I} & 0 \\ 0 & 0 & \mathtt{I} \\ -\mathtt{C}_0^{-1}\mathtt{C}_3 & -\mathtt{C}_0^{-1}\mathtt{C}_2 & -\mathtt{C}_0^{-1}\mathtt{C}_1 \end{pmatrix}. \tag{41}$$

After solving this eigenvalue problem we obtain 30 eigenvalues, solutions for $\beta = 1/z$, and 30 corresponding eigenvectors $\mathbf{v}$ from which we extract solutions for x and y.

However, there are 20 zero eigenvalues among these 30 eigenvalues. These zero eigenvalues can be easily eliminated since they correspond to the 20 zero columns of the matrices $-\mathtt{C}_0^{-1}\mathtt{C}_3$, $-\mathtt{C}_0^{-1}\mathtt{C}_2$, and $-\mathtt{C}_0^{-1}\mathtt{C}_1$, as described in Section 2.3.2. Therefore, to solve the 5-pt relative pose problem, it is sufficient to find the eigenvalues and the eigenvectors of the 10×10 matrix, which we obtain from the matrix (41) by removing columns corresponding to the zero columns of the matrices $-\mathtt{C}_0^{-1}\mathtt{C}_3$, $-\mathtt{C}_0^{-1}\mathtt{C}_2$, and $-\mathtt{C}_0^{-1}\mathtt{C}_1$ and removing the corresponding rows of this matrix (41).

Note that this 10×10 matrix is in fact the multiplication matrix used in the Gröbner basis solver [38], which was obtained directly from the polynomial eigenvalue formulation without any knowledge about the Gröbner bases or the properties of these multiplication matrices. The size of this matrix equals the dimension of the problem, which was proven to be 10 [30].

4.2 Six-Point Equal Focal Length Problem

Our polynomial eigenvalue solution to the 6-pt equal focal length problem starts with the parameterization of the fundamental matrix with two unknowns x and y (38) which is obtained from the epipolar constraint (28) for 6 point correspondences. Substituting this parameterization into the rank constraint for the fundamental matrix (29) and the trace constraint for the essential matrix (30) gives 10 third and fifth-order polynomial equations in three unknowns x, y, and $w = f^{-2}$, where f is the unknown focal length. This formulation is the same as the one used in [36], [21] and can be again written in a matrix form

$$\mathtt{M}\,X = 0, \tag{42}$$

where $\mathtt{M}$ is a 10×30 coefficient matrix and

$$X = (w^2x^3, w^2yx^2, w^2y^2x, w^2y^3, wx^3, wyx^2, wy^2x, wy^3, w^2x^2, w^2yx, w^2y^2, x^3, yx^2, y^2x, y^3, wx^2, wyx, wy^2, w^2x, w^2y, x^2, yx, y^2, wx, wy, w^2, x, y, w, 1)^\top$$

is a vector of 30 monomials. Variables x and y appear in degree three and w only in degree two. Therefore, we have selected $\lambda = w$ in the PEP formulation (1). Then, these 10 equations can be rewritten as

$$(w^2\mathtt{C}_2 + w\,\mathtt{C}_1 + \mathtt{C}_0)\mathbf{v} = 0, \tag{43}$$

where $\mathbf{v}$ is a 10×1 vector of monomials, $\mathbf{v} = (x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1)^\top$ and $\mathtt{C}_2$, $\mathtt{C}_1$, and $\mathtt{C}_0$ are 10×10 coefficient matrices.

$$\mathtt{C}_2 \equiv (m_1\;m_2\;m_3\;m_4\;m_9\;m_{10}\;m_{11}\;m_{19}\;m_{20}\;m_{26}),$$
$$\mathtt{C}_1 \equiv (m_5\;m_6\;m_7\;m_8\;m_{16}\;m_{17}\;m_{18}\;m_{24}\;m_{25}\;m_{29}),$$

and

$$\mathtt{C}_0 \equiv (m_{12}\;m_{13}\;m_{14}\;m_{15}\;m_{21}\;m_{22}\;m_{23} m_{27}\;m_{28}\;m_{30}),$$

where m_j is the jth column from the coefficient matrix $\mathtt{M}$ (42).

The formulation (43) is directly the QEP, which can again be solved using the methods presented in Section 2. After solving this QEP (43) by transforming it to a GEP, we obtain 20 eigenvalues, solutions for $w = f^{-2}$, and 20 corresponding eigenvectors $\mathbf{v}$ from which we extract solutions for x and y. To do this we normalize solutions for $\mathbf{v}$ to have the last coordinate 1 and extract values from $\mathbf{v}$ that correspond to x and y, in this case $\mathbf{v}(8)$ and $\mathbf{v}(9)$. Then, we use (38) to compute $\mathtt{F}$.

Again, there are zero eigenvalues, solutions to $1/w$, like in the 5-pt case. However, in this case, these eigenvalues cannot be so easily eliminated since here we do not have zero columns corresponding to these eigenvalues. Therefore, this polynomial eigenvalue solver delivers 20 solutions, which is more than the number of solutions of the original problem. This is caused by the fact that we are solving a relaxed version of the original problem as described in Section 2.2.3. Therefore, the solution contains not only all vectors $\mathbf{v}$ that (within limits of numerical accuracy) satisfy the constraints induced by the problem, i.e., $\mathbf{v}(1) = \mathbf{v}(8)^3$, but also additional vectors $\mathbf{v}$ that do not satisfy them. Such vectors $\mathbf{v}$ need to be eliminated, e.g., by verifying the monomial dependences as described in Section 2.2.3.

4.3 Six-Point One Calibrated Camera Problem

The polynomial eigenvalue solver to the third problem, the 6-pt relative pose problem for one fully and one up to the focal length calibrated camera, uses the same parameterization of the fundamental matrix as the previous 6-pt problem. In this case, this parameterization is substituted into (31) and (32), as described in Section 3.1, and results in 10 equations in three unknowns x, y, and $w = f^{-2}$. These equations can be rewritten into the matrix form

$$\mathtt{M}\, X = 0, \tag{44}$$

where $\mathtt{M}$ is a 10×20 coefficient matrix and

$$X = (wx^3, \mathrm{w}yx^2, wy^2x, wy^3, x^3, yx^2, y^2x, y^3, wx^2, wyx, wy^2, x^2, yx, y^2, wx, wy, x, y, w, 1)^\top$$

is a vector of 20 monomials.

Unknowns x and y appear in degree three, but w appears only in degree one. Therefore, we can select $\lambda = w$ and rewrite these 10 equations as

$$(w\, \mathtt{C}_1 + \mathtt{C}_0)\mathbf{v} = 0, \tag{45}$$

where $\mathbf{v} = (x^3, yx^2, y^2x, y^3, x^2, yx, y^2, x, y, 1)^\top$ is a 10×1 vector of monomials and $\mathtt{C}_1$, $\mathtt{C}_0$ are 10×10 coefficient matrices such that

$$\mathtt{C}_1 \equiv (m_1\ m_2\ m_3\ m_4\ m_9\ m_{10}\ m_{11}\ m_{15} m_{16}\ m_{19}),$$
$$\mathtt{C}_0 \equiv (m_5\ m_6\ m_7\ m_8\ m_{12}\ m_{13}\ m_{14}\ m_{17}\ m_{18} m_{20}),$$

where m_j is the jth column from the coefficient matrix $\mathtt{M}$.

The formulation (45) is directly the generalized eigenvalue problem (3), respectively, a polynomial eigenvalue problem of degree one with the regular matrix $\mathtt{C}_0$, and can be easily solved by finding the eigenvalues of the matrix $-\mathtt{C}_0^{-1}\mathtt{C}_1$.

After solving (45), we obtain 10 eigenvalues, solutions for $w = f^{-2}$, and 10 corresponding eigenvectors $\mathbf{v}$ from which we extract solutions for x and y. Then, we use (38) to get solutions for $\mathtt{F}$.

4.4 Eight-Point Radial Distortion Problem

The solver to the problem of estimating the relative pose of two uncalibrated cameras together with the one parameter division model from 8 point correspondences starts with nine equations in nine variables, i.e., eight equations from the epipolar constraint (34) and one from the singularity constraint on $\mathtt{F}$ (35).

In the first step we simplify these equations by eliminating some variables. We use the same elimination method as was used in [15].

Let the elements of the fundamental matrix be $f_{i,j}$. Assuming $f_{3,3} = 1$, the epipolar constraint (34) gives eight equations with 15 monomials $(f_{1,3}k, f_{2,3}k, f_{3,1}k, f_{3,2}k, k^2, f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}, f_{3,1}, f_{3,2}, k, 1)$ and 9 variables $(f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}, f_{3,1}, f_{3,2}, k)$. These monomials can be reordered such that monomials containing $f_{1,1}$, $f_{1,2}$, $f_{2,1}$, $f_{2,2}$, $f_{1,3}$ and $f_{2,3}$ are at the beginning. Reordered monomial vector will be $X = (f_{1,1}, f_{1,2}, f_{2,1}, f_{2,2}, f_{1,3}k, f_{1,3}, f_{2,3}k, f_{2,3}, f_{3,1}k, f_{3,2}k, k^2, f_{3,1}, f_{3,2}, k, 1)^\top$. Then, eight equations from the epipolar constraint can be written in a matrix form $\mathtt{M}\, X = 0$, where $\mathtt{M}$ is the coefficient matrix and X is this reordered monomial vector.

After computing the G-J elimination $\mathtt{M}'$ of $\mathtt{M}$ we obtain eight equations $\mathtt{M}'X = 0$ of the form

$$p_i = LT(p_i) + g_i(f_{3,1}, f_{3,2}, k) = 0, \tag{46}$$

where $LT(p_i)$ is the leading term of the polynomial p_i [6] which is, in our case, $LT(p_i) = f_{1,1}, f_{1,2}, f_{2,1}, f_{2,2}, f_{1,3}k, f_{1,3}, f_{2,3}k$ and $f_{2,3}$ for $i = 1, 2, 3, 4, 5, 6, 7, 8$. Polynomials $g_i(f_{3,1}, f_{3,2}, k)$ are second-order polynomials in three variables $f_{3,1}$, $f_{3,2}$, k. Next, we can express six variables, $f_{1,1}, f_{1,2}, f_{1,3}, f_{2,1}, f_{2,2}, f_{2,3}$ as the following functions of the remaining three variables $f_{3,1}, f_{3,2}, k$:

$$f_{1,1} = -g_1(f_{3,1}, f_{3,2}, k), \tag{47}$$

$$f_{1,2} = -g_2(f_{3,1}, f_{3,2}, k), \tag{48}$$

$$f_{1,3} = -g_6(f_{3,1}, f_{3,2}, k), \tag{49}$$

$$f_{2,1} = -g_3(f_{3,1}, f_{3,2}, k), \tag{50}$$

$$f_{2,2} = -g_4(f_{3,1}, f_{3,2}, k), \tag{51}$$

$$f_{2,3} = -g_8(f_{3,1}, f_{3,2}, k). \tag{52}$$

We can substitute expressions (47)-(52) into the remaining two equations p_5 and p_7 from the epipolar constraint (46) and also into the singularity constraint (35) for $\mathtt{F}$. In this way, we obtain three polynomial equations in three unknowns (two third-order polynomials and one fifth-order polynomial)

$$k(-g_6(f_{3,1}, f_{3,2}, k)) + g_5(f_{3,1}, f_{3,2}, k) = 0, \tag{53}$$

$$k(-g_8(f_{3,1}, f_{3,2}, k)) + g_7(f_{3,1}, f_{3,2}, k) = 0, \tag{54}$$

$$\det\begin{pmatrix} -g_1 & -g_2 & -g_6 \\ -g_3 & -g_4 & -g_8 \\ f_{3,1} & f_{3,2} & 1 \end{pmatrix} = 0. \tag{55}$$

If we take the unknown radial distortion parameter k to play the role of λ in PEP (1), we can rewrite these three equations as

$$(k^4\mathtt{C}_3 + k^3\mathtt{C}_3 + k^2\mathtt{C}_2 + k\mathtt{C}_1 + \mathtt{C}_0)\mathbf{v} = 0, \tag{56}$$

where $\mathbf{v} = (f_{3,1}^3, f_{3,1}^2 f_{3,2}, f_{3,1} f_{3,2}^2, f_{3,2}^3, f_{3,1}^2, f_{3,1}f_{3,2}, f_{3,2}^2, f_{3,1}, f_{3,2}, 1)^\top$ is a 10×1 vector of monomials and $\mathtt{C}_4$, $\mathtt{C}_3$, $\mathtt{C}_2$, $\mathtt{C}_1$, and $\mathtt{C}_0$ are 3×10 coefficient matrices.

Polynomial eigenvalue formulation (1) requires having square coefficient matrices $\mathtt{C}_\mathrm{j}$. Unfortunately, in this case, we do not have square coefficient matrices. We only have three equations and 10 monomials in the vector $\mathbf{v}$. Therefore, we will use the method of transforming the system of polynomial equations to the PEP described in Section 2.2.2. The standard resultant-based method from Section 2.2.1 also produces a polynomial eigenvalue formulation of this problem but the modified method, Section 2.2.2, improves the numerical stability of the final solver a little bit.

Equations (53) and (54) are equations of degree one when considered as polynomials form $(\mathbb{C}[k])(f_{3,1}, f_{3,2})$, while (55) is an equation of degree three. Therefore, the degree d (17), defined in this method as $\mathrm{d} = \sum_{i=1}^{n}(d_i - 1) + 1$, is equal to three. This means that the sets $\overline{S_i}$ (21) will consist of monomials

$$\overline{S_1} = \{f_{3,1}^3, f_{3,1}^2 f_{3,2}, f_{3,1} f_{3,2}^2, f_{3,1}^2, f_{3,1} f_{3,2}, f_{3,1}\},$$
$$\overline{S_2} = \{f_{3,2}^3, f_{3,1}^2 f_{3,2}, f_{3,1} f_{3,2}^2, f_{3,2}^2, f_{3,1} f_{3,2}, f_{3,2}\},$$
$$\overline{S_3} = \{1\},$$

after the dehomogenization (see Section 2.2.2).

The extended system of (22), which is equivalent to the initial system of (53)-(55), will therefore consist of 13 equations, i.e., (53) and (54) multiplied with the monomials $f_{3,1}^2, f_{3,1}f_{3,2}, f_{3,2}^2, f_{3,1}, f_{3,2}, 1$ and (55), and will contain only 10 monomials. Therefore, this system can be rewritten to the polynomial eigenvalue form (56) with coefficient matrices $\mathtt{C_j}$ of size 13×10.

Since we have more equations than monomials, we can select 10 equations from them such that the resulting system will have a small condition number. The resulting formulation will be a PEP of degree four.

After solving this PEP (56), we obtain 40 solutions for λ and 40 corresponding eigenvectors $\mathbf{v}$ from which we extract solutions to $f_{3,1}$ and $f_{3,2}$.

Also in this case, the eigenvalue problem results in several zero eigenvalues. Eleven of these zero eigenvalues correspond to zero columns of the coefficient matrices $\mathtt{C}_j$ and can be removed using the method describe in Section 2.3.2. This means that we reduce the original 40×40 eigenvalue problem to a 29×29 eigenvalue problem. We still obtain more solutions than the number of solutions to the original system of polynomial equations, which is in this case 16. Therefore, we need to eliminate these "parasitic" solutions by, e.g., verifying monomial dependencies, see Section 2.2.3.

After finding the solutions to k, $f_{3,1}$, and $f_{3,2}$, we can use (47)-(52) to get solutions for the fundamental matrix $\mathtt{F}$.

5 Experiments

In this section, we evaluate our solutions and compare them with the existing state-of-the-art methods. Since all methods are algebraically equivalent and solvers differ only in the way of solving problems, we have evaluated them on synthetic noise-free data only. We aimed at studying the numerical stability and speed of the algorithms. The properties of the individual problems in different configurations and under different noise contaminations are studied in [29], [38], [36], [3], [15], and [19].

5.1 Numerical Stability

In all our experiments, the scenes were generated using 3D points randomly distributed in a 3D cube. Each 3D point was projected by a camera with random feasible orientation and position and random or fixed focal length. For the radial distortion problem, the radial distortion using the division model [10] was added to all image points.

For the calibrated 5-pt problem, we extracted camera relative rotations and translations from estimated essential

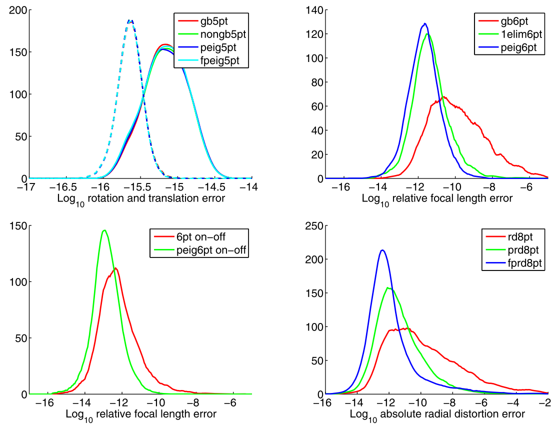

Fig. 1. Numerical stability of the algorithms considered.

matrices. From the four possible choices of rotations and translations, we selected the one where all 3D points were in front of the canonical camera pair [12]. Let $\mathtt{R}$ be an estimated camera relative rotation and $\mathtt{R}_{gt}$ the corresponding ground-truth rotation. The rotation error is measured as the angle in the angle axis representation of the relative rotation $\mathtt{R}\,\mathtt{R}_{gt}^{-1}$ and the translation error as an angle between ground-truth and estimated translation vectors. Fig. 1 top-left compares results of different 5-pt solvers: gb5pt denotes Stewenius Gröbner basis solver [38], nongb5pt—Nister's [29], peig5pt—the polyeig solver presented in [17], and fpeig5pt denotes the fast polyeig solver, with removed zero eigenvalues proposed in this work. The rotation error is displayed with solid line and the translation error with dashed line. The numerical stability of all solvers is very good.

In evaluations of both 6-pt problems, we focused on the value of estimated focal length. We measured the relative focal length error $(f - f_{gt})/f_{gt}$, where f is the estimated focal length and f_{gt} denotes the ground truth. Fig. 1 top-right compares the 6-pt solvers for a pair of cameras with a constant focal length. In this figure, gb6pt denotes the Stewenius Gröbner basis solver [36], 1elim6pt—the Gröbner basis solution with single elimination created using the automatic generator [16], and peig6pt denotes the polyeig solver presented in this paper. The polynomial eigenvalue solver outperforms both Gröbner basis solvers.

Solvers for a pair of an internally calibrated camera and a camera with unknown focal length are compared in Fig. 1 bottom-left. Again, the polynomial eigenvalue solver (peig6pt on-off) is a little bit more stable compared to the Gröbner basis solver (6pt on-off) described in [3].

Finally, the bottom-right plot in Fig. 1 compares 8-pt solvers for two cameras with unknown constant radial distortion. Here, the Gröbner basis solver [15] (rd8pt) provides less stable solutions compared to both polyeig solvers. Moreover, the polyeig solver with reduced zero eigenvalues (fprd8pt) outperforms the not-reduced solver both in precision and speed.

5.2 Computational Complexity

We have implemented all solvers presented in this paper in C++. In all our implementations, we have decided to transform the resulting polynomial eigenvalue problems to the standard eigenvalue problems (11), as described in Section 2.1, and use standard numerical eigenvalue methods to solve them.

All presented solvers are easy to implement since in each solver only a few matrices are filled with appropriate coefficients extracted from the input equations. Then, matrix (11), which is the input to some standard numerical eigenvalue algorithm, is constructed using these matrices. We have implemented our own eigenvalue solver according to [32]; however, LAPACK eigenvalue routine or other well-known algorithms can also be used [1].

We next compare the computational complexity of the solvers presented in this paper with the state-of-the-art methods.

In the case of the 5-pt relative pose problem, the presented polynomial eigenvalue method requires computing the inverse of one 10×10 matrix and then computing eigenvalues of the matrix of the same size. This is equivalent to the number and type of the operations used in the Gröbner basis solver [38]. The difference is only in the method used for obtaining the resulting 10×10 matrix.

For this 5-pt problem, we have found that it is more efficient to directly compute the characteristic polynomial of the 10×10 matrix, using the Faddeev-Leverrier method [7], and to find solutions by computing the roots of this 10th-order characteristic polynomial using the Sturm sequences [23] than to use standard eigenvalue algorithms.

Compared to the state-of-the-art solver from [29], the polyeig solver presented in this paper is a little bit less efficient. This is because the solver from [29] requires computing the inverse of one 10×10 matrix and then creates the 10th-order polynomial by computing one 3×3 polynomial determinant. This is more efficient than the computation of this polynomial used in our case because the characteristic polynomial formula [7] requires computing trace of a 10×10 matrix to power 10.

Comparing the speed of the solvers, our solver runs on Intel i7 Q720 notebook about 31 μs while the only available implementation of the solver from [38] runs about 244 μs. However, since the Gröbner basis solver [38] consists of the same operations as our proposed solver, the same time as ours can be achieved. The optimized version of the solver from [29] should be even faster; however, it is not publicly available.

In the case of the 6-pt equal focal length problem, the polynomial eigenvalue method requires computing the inverse of one 10×10 matrix and then computing eigenvalues of the 20×20 matrix. This solver runs on the same hardware as the 5-pt solver, about 182 μs. The best available implementation of the Gröbner basis solver from [16] requires performing G-J elimination of a 31×46 matrix and to compute eigenvalues of the 15×15 matrix and runs about 650 μs.

The polynomial eigenvalue solver for the 6-pt problem for one calibrated and one up to focal length calibrated camera requires computing the inverse of one 10×10 matrix and then eigenvalues of the 10×10 matrix. The solver runs about 30 μs. The Gröbner basis solver from [3] requires performing G-J elimination of a 21×30 matrix and computing eigenvalues of the 9×9 matrix. Its implementation from [3] runs about 259 μs.

Finally, the polynomial eigenvalue solution to the 8-pt radial distortion problem requires computing the inverse of one 10×10 matrix and eigenvalues of the 29×29 matrix. This solver runs about 685 μs. The best available implementation of the Gröbner basis solver from [16] requires performing G-J elimination of a 32×48 matrix and computing eigenvalues of the 16×16 matrix and runs about 640 μs; however, this time can still be reduced by a better implementation of this solver.

Note that all mentioned Gröbner basis solvers [38], [16], [3] are partially implemented in MATLAB and therefore their running times can be further improved. However, the problem is that most of these solvers are quite complicated to understand and without some knowledge of algebraic geometry they cannot be easily reimplemented.

About 90 percent of the time of all presented polynomial eigenvalue solvers is spent in the eigenvalue and eigenvector computation. This will also be the most consuming part in cleverly reimplemented Gröbner basis solvers [38], [16], [3].

6 Conclusion

In this paper, we have presented the polynomial eigenvalue method for solving systems of polynomial equations appearing in computer vision. Compared to the state-of-the-art Gröbner basis method, the presented method is more straightforward and easier to implement since eigenvalue problems are well studied, easy to understand, and efficient and robust algorithms for solving these problems are available. We have shown this by presenting very simple, fast, stable, and easy to implement solutions to four important minimal relative pose problems.

Moreover, we have characterized the problems that can be efficiently solved as polynomial eigenvalue problems and presented a resultant-based method for transforming a system of polynomial equations to a polynomial eigenvalue problem. Finally, we have proposed some useful techniques for reducing the size of the computed polynomial eigenvalue problems.

Acknowledgments

This work has been supported by EC project FP7-SPACE-241523 PRoViScout and by The Czech Government under the research program MSM6840770038.

References

[1] Z. Bai, J. Demmel, J. Dongorra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems.* SIAM, 2000.

[2] M. Bujnak, Z. Kukelova, and T. Pajdla, "A General Solution to the P4P Problem for Camera with Unknown Focal Length," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[3] M. Bujnak, Z. Kukelova, and T. Pajdla, "3D Reconstruction from Image Collections with a Single Known Focal Length," *Proc. 12th IEEE Int'l Conf. Computer Vision,* 2009.

[4] M. Bujnak, Z. Kukelova, and T. Pajdla, "New Efficient Solution to the Absolute Pose Problem for Camera with Unknown Focal Length and Radial Distortion," *Proc. 10th Asian Conf. Computer Vision,* 2010.

[5] M. Byröd, K. Josephson, and K. Åström, "Improving Numerical Accuracy of Gröbner Basis Polynomial Equation Solver," *Proc. 11th IEEE Int'l Conf. Computer Vision,* 2007.

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

[7] D.K. Faddeev and W.N. Faddeeva, *Computational Methods of Linear Algebra.* Freeman, 1963.

[8] O. Faugeras and S. Maybank, "Motion from Point Matches: Multiplicity of Solutions," *Int'l J. Computer Vision,* vol. 4, no. 3, pp. 225-246, 1990.

[9] 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,* vol. 24, no. 6, pp. 381-395, 1981.

[10] A. Fitzgibbon, "Simultaneous Linear Estimation of Multiple View Geometry and Lens Distortion," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2001.

[11] C. Geyer and H. Stewenius, "A Nine-Point Algorithm for Estimating Para-Catadioptric Fundamental Matrices," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

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

[13] K. Josephson, M. Byröd, and K. Aström, "Pose Estimation with Radial Distortion and Unknown Focal Length," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[14] E. Kruppa, "Zur Ermittlung eines Objektes aus Zwei Perspektiven mit Innerer Orientierung," Sitz.-Ber. Akad.Wiss., Wien, Math. Naturw. Kl., Abt. IIa., vol. 122, pp. 1939-1948, 1918.

[15] Z. Kukelova and T. Pajdla, "A Minimal Solution to the Autocalibration of Radial Distortion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[16] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic Generator of Minimal Problem Solvers," *Proc. 10th European Conf. Computer Vision,* 2008.

[17] Z. Kukelova, M. Bujnak, and T. Pajdla, "Polynomial Eigenvalue Solutions to the 5-Pt and 6-Pt Relative Pose Problems," *Proc. 19th British Machine Vision Conf.,* Sept. 2008.

[18] Z. Kukelova, M. Byröd, K. Josephson, T. Pajdla, and K. Åström, "Fast and Robust Numerical Solutions to Minimal Problems for Cameras with Radial Distortion," *Computer Vision and Image Understanding,* vol. 114, no. 2, pp. 234-244, Feb. 2010.

[19] Z. Kukelova and T. Pajdla, "A Minimal Solution to Radial Distortion Autocalibration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 12, pp. 2410-2422, Dec. 2011.

[20] H. Li and R. Hartley, "A Non-Iterative Method for Correcting Lens Distortion from Nine-Point Correspondences," *Proc. Omni-Vision,* 2005.

[21] H. Li, "A Simple Solution to the Six-Point Two-View Focal-Length Problem," *Proc. European Conf. Computer Vision,* pp. 200-213, 2006.

[22] H. Li and R. Hartley, "Five-Point Motion Estimation Made Easy," *Proc. 18th Int'l Conf. Pattern Recognition,* pp. 630-633, 2006.

[23] D. Hook and P. McAree, "Using Sturm Sequences to Bracket Real Roots of Polynomial Equations," *Graphic Gems I,* pp. 416-423, Academic Press, 1990.

[24] X. Li, C. Wu, C. Zach, S. Lazebnik, and J. Frahm, "Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs," *Proc. 10th European Conf. Computer Vision,* 2008.

[25] D. Manocha and J.F. Canny, "Multipolynomial Resultant Algorithms," *J. Symbolic Computation,* vol. 15, no. 2, pp. 99-122, 1993.

[26] D. Manocha, "Solving Systems of Polynomial Equations," *IEEE Computer Graphics and Applications,* vol. 14, no. 2, pp. 46-55, Mar. 1994.

[27] D. Martinec and T. Pajdla, "Robust Rotation and Translation Estimation in Multiview Reconstruction," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[28] B. Micusik and T. Pajdla, "Estimation of Omnidirectional Camera Model from Epipolar Geometry," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 485-490, 2003.

[29] D. Nister, "An Efficient Solution to the Five-Point Relative Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 756-770, June 2004.

[30] D. Nister, R. Hartley, and H. Stewenius, "Using Galois Theory to Prove that Structure from Motion Algorithms Are Optimal," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[31] J. Philip, "A Non-Iterative Algorithm for Determining All Essential Matrices Corresponding to Five Point Pairs," *Photogrammetric Record,* vol. 15, no. 88, pp. 589-599, 1996.

[32] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C++: The Art of Scientific Computing.* Cambridge Univ. Press, Feb. 2002.

[33] N. Snavely, S.M. Seitz, and R.S. Szeliski, "Photo Tourism: Exploring Image Collections in 3D," *Proc. Siggraph,* pp. 835-846, 2006.

[34] N. Snavely, S. Seitz, and R. Szeliski, "Skeletal Graphs for Efficient Structure from Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[35] H.J. Stetter, *Numerical Polynomial Algebra.* SIAM, 2004.

[36] H. Stewénius, D. Nister, F. Kahl, and F. Schaffalitzky, "A Minimal Solution for Relative Pose with Unknown Focal Length," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 789-794, 2005.

[37] H. Stewenius, D. Nister, M. Oskarsson, and K. Astrom, "Solutions to Minimal Generalized Relative Pose Problems," *Proc. Workshop Omnidirectional Vision,* 2005.

[38] H. Stewenius, C. Engels, and D. Nister, "Recent Developments on Direct Relative Orientation," *ISPRS J. Photogrammetry and Remote Sensing,* vol. 60, pp. 284-294, 2006.

[39] M. Urbanek, R. Horaud, and P. Sturm, "Combining Off- and On-Line Calibration of a Digital Camera," *Proc. Third Int'l Conf. 3-D Digital Imaging and Modeling,* 2001.

[40] A. Wallack, I.Z. Emiris, and D. Manocha, "MARS: A Maple/Matlab/C Resultant-Based Solver," *Proc. Int'l Symp. Symbolic and Algebraic Computation,* pp. 244-251, 1998.

Zuzana Kukelova received the master's degree in mathematics from the Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, in 2005. Currently, she is working toward the PhD degree at the Center for Machine Perception, which is a part of the Department of Cybernetics at the Czech Technical University in Prague. She is interested in algebraic geometry and computer vision, where she focuses on minimal problems and methods for solving systems of polynomial equations. She is a member of the IEEE.

Martin Bujnak received the master's degree in computer graphics and parallel processing from the Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, in 2005. Currently, he is working toward the PhD degree at the Center for Machine Perception, which is a part of the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague. In his research, he focuses on camera calibration, pose estimation, and structure from motion problems. He is a member of the IEEE.

Tomas Pajdla received the MSc and PhD degrees from the Czech Technical University, in Prague. He works in geometry and algebra of computer vision and robotics, with emphasis on nonclassical cameras, 3D reconstruction, and industrial vision. He contributed to introducing epipolar geometry of panoramic cameras, noncentral camera models generated by linear mappings, generalized epipolar geometries, and to developing solvers for minimal problems in structure from motion. He is a member of the IEEE.