

Sampling Algebraic Varieties for Robust Camera Autocalibration

Danda Pani Paudel¹ and Luc Van Gool^{1,2}

¹ Computer Vision Lab, ETH Zürich, Switzerland

² VISICS, ESAT/PSI, KU Leuven, Belgium

{paudel, vangool}@vision.ee.ethz.ch

Abstract. This paper addresses the problem of robustly autocalibrating a moving camera with constant intrinsics. The proposed calibration method uses the Branch-and-Bound (BnB) search paradigm to maximize the consensus of the polynomials. These polynomials are parameterized by the entries of, either the Dual Image of Absolute Conic (DIAC) or the Plane-at-Infinity (PaI). During the BnB search, we exploit the theory of sampling algebraic varieties, to test the positivity of any polynomial within a parameter's interval, i.e. outliers with certainty. The search process explores the space of exact parameters (i.e the entries of DIAC or PaI), benefits from the solution of a local method, and converges to the solution satisfied by the largest number of polynomials. Given many polynomials on the sought parameters (with possibly overwhelmingly many from outlier measurements), their consensus for calibration is searched for two cases: simplified Kruppa's equations and Modulus constraints, expressed in DIAC and PaI, resp. Our approach yields outstanding results in terms of robustness and optimality.

1 Introduction

Estimating the intrinsics of a moving camera is difficult mainly due the non-linear nature of the problem. Furthermore, it also demands the camera motion to be rich and diverse (with sufficiently large translations and rotations), such that the degenerate motions for autocalibration – so-called Critical Motion Sequences (CMS) – are avoided. Yet, when the camera undergoes a large motion, establishing a good set of correspondences across multiple views becomes very challenging mainly due to the change in viewpoints or occlusions. Failure to establish such correspondences leads to an inaccurate estimation of camera motion, thus demanding a robust method for camera autocalibration. Although the problem of accurate motion estimation may be partially handled by carefully capturing the image sets, such solutions are not always possible especially when the image acquisitions cannot be controlled as desired (e.g. remote cameras, holidays pictures). Furthermore, camera motion estimation is expected to be riddled with inaccuracies and outliers, mainly due to the presence of repetitive patterns, changes in illumination, and partial/no scene overlaps.

Camera autocalibration has obtained significant attention since the seminal work by *Maybank et Faugeras* [1]. Existing approaches can be broadly divided into three categories: (i) direct estimation of a ubiquitous conic which encodes the camera intrinsics: the so-called Dual Image of the Absolute Conic (DIAC) [2–4], (ii) stratified estimation

of the Plane-at-Infinity (PaI) followed by a linear retrieval of the DIAC [5–9], (iii) joint estimation of both PaI and DIAC in the form of the so-called Dual Image of the Absolute Quadric (DIAQ) [10–12]. On the one hand, most of these methods are only locally optimal (requiring a good initialization of the sought intrinsics) and susceptible to even a small number of outliers. On the other hand, globally optimal methods [9, 11, 12] only focus on the optimality while discarding the criteria for robustness towards outliers.

A relatively robust and globally optimal method [13] performs the interval analysis within the framework of Branch-and-Bound (BnB) to solve the modified robust cost function proposed in [14], which was originally derived in [2]. Unfortunately, due to the inefficiency of the interval analysis technique, the method has proved computationally very expensive and applicable only to rather short image sequences. Furthermore, this method also suffers from the same problem as [14] for high numbers of outliers. To the best of our knowledge, there exist no efficient minimal solver for full camera calibration that offers a practical way of conducting robust estimation within the framework of RANSAC. Furthermore, RANSAC-based methods would still be non-deterministic and fail to provide meaningful solutions in the presence of many outliers. Existing minimal solvers such as [15–17] make strong assumptions on an unknown focal length with known aspect ratio. A detailed study on Minimal Conditions for camera autocalibration is provided in [18]. Note that a variety of methods for solving systems of nonlinear polynomial equations exist. While some are based on Gröbner bases or homotopy continuation [19], others use Sum-of-Squares (SoS) polynomial optimization [20–22]. Yet, such methods are dedicated to solving outlier-free nonlinear systems and dealing with outliers is carried out through RANSAC.

In this work, we address the problem of robustly autocalibrating a moving camera with constant intrinsics. The proposed method uses the BnB search paradigm to solve both direct and stratified calibrations, namely simplified Kruppa’s equations [4] and Modulus constraints [7], parameterized by DIAC and PaI, respectively. Although, *Gurdjos et al.* [23] suggest that the solutions to Kruppa’s equations or Modulus constraints may suffer from artificial CMS (due to the failure of enforcing the Absolute Conic to lie on the PaI), we argue that the artificial CMS are unlikely to happen for cameras under large motion. In such cases, robustness towards outliers is rather more important to deal with. In fact, the joint estimation of the DIAC and PaI, in the form of rank-3 DAQ, not only makes the problem more challenging but also demands a high quality projective reconstruction, thus making it less suitable for robust camera calibration.

During the BnB search, we explore the DIAC or PaI parameter space. As in [24, 25], we rely on establishing optimistic and pessimistic sets of inlier assignments for pruning branches whose most optimistic sets are worse than the best pessimistic one. For any branch, we obtain the pessimistic inlier set using a local refinement method that guarantees the solution to lie within the sought interval. To estimate the optimistic inlier set, we exploit the theory of sampling algebraic varieties for testing the positivity/negativity of polynomials on the given varieties. In this regard, polynomials to be tested, say interval polynomials, are derived from the parameters’ current intervals, whereas varieties are represented by the polynomials from either the Kruppa’s equations or Modulus constraints. The interval polynomials are quadratic in nature and designed such that they are always negative within the considered interval while being positive elsewhere. If the

interval polynomial is positive/negative on any variety, the measurement corresponding to that variety is an outlier, with certainty, for that particular interval. The optimistic inliers are then estimated by discarding such outliers for the total measurements.

The major contributions of this paper can be summarized as: (i) For the first time, we introduce the theory of sampling algebraic varieties to detect the outlier polynomials with certainty, within the considered parameters' intervals. (ii) Based on the rigorous theory of algebraic geometry, we devise an efficient and optimal BnB-based search method to maximize the consensus of the polynomials sharing common real roots. Furthermore, we also provide the common root as the solution to the given polynomial system with many outliers. (iii) The proposed method has been tested on two challenging autocalibration problems, demonstrating outstanding results both in terms of robustness and optimality.

2 Sampling Varieties Theory

Consider the ring $\mathbb{R}[x] := \mathbb{R}[x_1, \dots, x_n]$ of multivariate polynomials and an algebraic variety $\mathcal{V} \subseteq \mathbb{C}^n$ defined such that $\mathcal{V} := \{x \in \mathbb{C}^n : h_i(x) = 0, \text{ for } i = 1, \dots, m\}$. For a given polynomial $p(x) \in \mathbb{R}[x]$, we are interested to know whether,

$$p(x) \geq 0, \text{ for all } x \in \mathcal{V} \cap \mathbb{R}^n. \quad (1)$$

The decision problem in (1) is NP-hard. However, there are some relaxation methods based on sum of squares (SoS) [26, 25]. Recall that a polynomial $f(x) \in \mathbb{R}[x]$ is SoS, if there exist polynomials $f_i(x) \in \mathbb{R}[x]$ such that $f(x) = \sum_i (f_i(x))^2$. Given a bound $d \in \mathbb{N}$, it is straightforward to observe that (1) must hold true, if there exists a SoS polynomial $f(x)$ of degree $\leq 2d$ such that,

$$p(z) = f(z), \text{ for all } z \in \mathcal{V}. \quad (2)$$

We refer to such $f(x)$ as a d-SoS certificate. Now, we are interested to know the answer to the following problem.

Problem 1. Given a bound $d \in \mathbb{N}$, a polynomial $p(x)$ and a variety \mathcal{V} , does there exist a d-SoS certificate?

An affirmative answer to Problem 1 is a sufficient condition for (1) to be true. To answer the Problem 1, we rely on the theory of sampling varieties, originally developed in [27]. This theory uses a generic set of samples $\mathcal{Z} = \{z_1, \dots, z_S\} \subseteq \mathcal{V}$, while specializing the condition in (2) to \mathcal{Z} . Intuitively, if (2) is satisfied for a sufficiently large sample set $\mathcal{Z} \subseteq \mathcal{V}$, it must also be true for all $z \in \mathcal{V}$, as long as both the number of variables and the degree of $p(x)$ are bounded. Then one may be interested to know what is the smallest size of \mathcal{Z} required to conclude that (2) is indeed true.

Unfortunately, there is not an easy way to find the minimal size of \mathcal{Z} . But the good news is that, for a given \mathcal{Z} , one can test whether it is sufficient to find a d-SoS certificate. Using the method proposed in [27], such certificate can be obtained in two steps; (i) pre-certificate computation, (ii) poisedness test.

Definition 1 (Pre-certificate). For a variety $\mathcal{V} \subseteq \mathbb{C}^n$, a non negative polynomial $p(x) \in \mathbb{R}[x]$ on $\mathcal{V} \cap \mathbb{R}^n$, and a bound $d \in \mathbb{N}$, a sampling d -SoS pre-certificate is a pair $(f(x), \mathcal{Z})$, where $f(x)$ is a SoS polynomial of degree $\leq 2d$ and $\mathcal{Z} = \{z_1, \dots, z_S\} \subseteq \mathcal{V}$ a sample set, such that,

$$p(z_s) = f(z_s), \text{ for } s = 1, \dots, S. \quad (3)$$

If $f(x)$ is a d -SoS certificate, the pre-certificate is correct.

Finding a pre-certificate with (3), is a Semi-Definite Programming (SDP) problem.

$$\begin{aligned} \text{find } & \mathbf{Q} \in \mathcal{S}^N, \quad \mathbf{Q} \succeq 0, \\ \text{s.t. } & p(z_s) = u(z_s)^\top \mathbf{Q} u(z_s) \quad \text{for } s = 1, \dots, S. \end{aligned} \quad (4)$$

Where, $u(x) \in \mathbb{R}[x]^N$ is the vector of all monomials of degree at most d , and \mathcal{S}^N denotes the space of $N \times N$ real symmetric matrices. Recall that, a polynomial $f(x) \in \mathbb{R}[x]$ is d -SoS if and only if, $f(x) = u(x)^\top \mathbf{Q} u(x)$ for a positive definite matrix \mathbf{Q} (denoted by $\mathbf{Q} \succeq 0$). Therefore, if there exists any matrix $\mathbf{Q} \succeq 0$ that satisfies (4), the pre-certificate (defined in Definition 1) is correct. Although the pre-certificate obtained in this manner does not necessarily guarantee that (2) is satisfied, it can be used for the certainty, if the set \mathcal{Z} is poised.

Definition 2 (Poisedness). Let $\mathcal{L} \subseteq \mathbb{R}[\mathcal{V}]$ be a linear subspace. For a given set of samples $\mathcal{Z} \subseteq \mathcal{V}$, $(\mathcal{L}, \mathcal{Z})$ is called poised if the only polynomial $p(x) \in \mathcal{L}$ with $p(z) = 0$ for all $z \in \mathcal{Z}$, is the zero polynomial. Furthermore, for any finite dimensional \mathcal{L} there is a finite set \mathcal{Z} such that $(\mathcal{L}, \mathcal{Z})$ is poised [27].

Let $\mathcal{L}_d \subseteq \mathbb{R}[\mathcal{V}]$ and $\mathcal{L}_{2d} \subseteq \mathbb{R}[\mathcal{V}]$ are the linear spaces spanned by the entries of $u(x)$ and $u(x)u(x)^\top$, respectively. The polynomial $f(x) = u(x)^\top \mathbf{Q} u(x) \in \mathcal{L}_{2d}$. There exist a simple strategy to test if the given pair $(\mathcal{L}, \mathcal{Z})$ is poised or not. The test is summarized in Algorithm 1. Please, refer [27] for the details about the poisedness test. Now, the fol-

Algorithm 1 [*testFlag*] = *poisednessTest*($u(x), \mathcal{Z}$)

1. Form vector $u_2(x) = \text{vec}(u(x)u(x)^\top)$ spanning \mathcal{L}_{2d} .
 2. Build a matrix \hat{U}_2 with columns $u_2(z)$ for $z \in \mathcal{Z} \cup \bar{\mathcal{Z}}$.
 3. If \hat{U}_2 has full column rank, return “false”. Else, return “true”.
-

lowing theorem guarantees the correctness of pre-certificate for a good set of samples, i.e. when $(\mathcal{L}_{2d}, \mathcal{Z})$ is poised.

Theorem 1 (Poisedness implies correctness [27]). For any given set of samples \mathcal{Z} , its pre-certificate $(f(x), \mathcal{Z})$ is correct, if $(\mathcal{L}_{2d}, \mathcal{Z})$ is poised.

To summarize, the decision Problem of (1) can be answered with the help of a generic sample set \mathcal{Z} . For a general variety \mathcal{V} , one can use numerical algebraic geometry tools (such as Bertini [28] and PHCpack [29]) to compute such sample sets. The answer to the Problem (1) is affirmative, if there exists $\mathbf{Q} \succeq 0$ satisfying (3) and \mathcal{Z}

passes the poisedness test of Algorithm 1. In case of failure, the poisedness test of \mathcal{Z} can be ensured by adding more samples. Recall from Definition 2, for any finite dimensional \mathcal{L} there is a finite set \mathcal{Z} that passes the poisedness test. Needless to say that if $p(z_s)$ is negative for any z_s , the test for Problem of (1) is not really necessary. More importantly, if the pre-certificate cannot be obtained even for the smallest \mathcal{Z} that passes the poisedness test, the answer to the Problem (1) is undetermined.

3 Consensus within a System of Polynomials

Consider a set of measurements $\{\mathcal{M}_i\}_{i=1}^m$ such that each \mathcal{M}_i can be expressed as a set polynomials $\mathcal{P}_i = \{h_{ij}(x)\}_{j=1}^l$, on the unknown parameter $x \in \mathbb{R}^n$. Let the variety $\mathcal{V}_i := \{x \in \mathbb{R}^n : h_{ij}(x) = 0, \text{ for } j = 1, \dots, l\}$, where $h_{ij}(x) = 0, \text{ for } j = 1, \dots, l$ are the polynomials obtained from the measurement \mathcal{M}_i . Ideally, we are interested to find an $x \in \cap_{i=1}^m \mathcal{V}_i$. However, in the presence of noise and outliers, such x may not even exist. Therefore, we wish to solve the following Problem.

Problem 2. Given a set $\mathcal{S} = \{\mathcal{P}_i\}_{i=1}^m$ and a threshold ϵ ,

$$\begin{aligned} & \max_{x, \zeta \subseteq \mathcal{S}} |\zeta|, \\ & \text{subject to } \mathbf{d}(x, \mathcal{V}_i) \leq \epsilon, \quad \forall \mathcal{P}_i \in \zeta, \end{aligned} \quad (5)$$

for a sample x to variety \mathcal{V} distance defined by,

$$\mathbf{d}(x, \mathcal{V}) = \min_{y \in \mathcal{V}} \|x - y\|. \quad (6)$$

This problem, however, is difficult to solve due to its non-convex and NP-hard combinatorial nature. In this work, we approach this problem using the BnB algorithmic paradigm. Our BnB search is performed by branching on the space of parameters $x \in \mathbb{R}^n$. Every branch is represented by an interval of parameters x in the form of two vectors $[\underline{x}, \bar{x}]$ such that $\underline{x} \leq \bar{x}$, for the entry-wise inequality.

3.1 Polynomials within an Interval

The key idea of this paper is an effective way of estimating the optimistic number of inliers/outliers measurements, represented by a set of polynomials, for each branch by answering the following problem.

Problem 3. For any given measurement \mathcal{M}_i and a parameter's interval $[\underline{x}, \bar{x}]$, does there exist a vector $x \in [\underline{x}, \bar{x}]$ such that $\mathbf{d}(x, \mathcal{V}_i) \leq \epsilon$?

In other words, we would like to know whether all the polynomials $h_{ij} \in \mathcal{P}_i$ share at least one common root within the given interval representing a branch, with an ϵ tolerance. If this question is answered affirmatively, then the measurement \mathcal{M}_i is a potential inlier within the considered interval. This problem however, is difficult to answer unless the following proposition is considered.

Proposition 1. *For the given threshold ϵ and parameter's interval $[\underline{x}, \bar{x}]$, let us define a polynomial $p_b(x)$ representing the interval bounds:*

$$p_b(x) = \left\| x - \frac{x + \bar{x}}{2} \right\|^2 - \left(\left\| \frac{\bar{x} - x}{2} \right\| + \epsilon \right)^2. \quad (7)$$

For any measurement \mathcal{M}_i , there exists no $x \in [\underline{x}, \bar{x}]$ such that $\mathbf{d}(x, \mathcal{V}_i) \leq \epsilon$, if $p_b(x) \geq 0$ for all $x \in \mathcal{V}_i \cap \mathbb{R}^n$.

Proof. Observe that $p_b(x) \geq 0$ is the interior of a sphere with center $\frac{x + \bar{x}}{2}$ and radius $\left\| \frac{\bar{x} - x}{2} \right\| + \epsilon$, which includes all $x \in [\underline{x}, \bar{x}]$ with ϵ tolerance. Therefore, if $p_b(x) \geq 0$ for all $x \in \mathcal{V}_i \cap \mathbb{R}^n$, there exists no $x \in [\underline{x}, \bar{x}]$ such that $\mathbf{d}(x, \mathcal{V}_i) \leq \epsilon$. \square

Proposition 1 allows us to answer the Problem 3 similar to the decision problem of (1). The ability to answer the Problem 3 allows us to reason about whether the measurement \mathcal{M}_i is an inlier or outlier. Recall that a measurement \mathcal{M}_i is an inlier if $\mathbf{d}(x, \mathcal{V}_i) \leq \epsilon$. Alternatively, there exists no $x \in [\underline{x}, \bar{x}]$ such that $\mathbf{d}(x, \mathcal{V}_i) \leq \epsilon$, then the measurement \mathcal{M}_i is definitely an outlier within the considered bounds. Otherwise, it is a potential inlier. For interval bounds $[\underline{x}, \bar{x}]$ and a threshold ϵ , we summarize the method to test whether the given measurement \mathcal{M}_i is an outlier, in Algorithm 2.

Algorithm 2 [*testFlag*] = `OutlierTest`($[\underline{x}, \bar{x}], \mathcal{M}_i, \epsilon$)

1. Construct $p_b(x)$ using (7), for bounds $[\underline{x}, \bar{x}]$ and threshold ϵ .
 2. For given \mathcal{M}_i , construct \mathcal{P}_i to define its variety \mathcal{V}_i .
 3. Test if $p_b(x) \geq 0$, for all $x \in \mathcal{V}_i \cap \mathbb{R}^n$, as in (1) in two steps:
 - (i) Compute the pre-certificate using (4).
 - (ii) Verify the pre-certificate using Algorithm 1.
 4. If the test holds true, return “true”. Else, return “false”.
-

Now we are interested to know whether Algorithm 2 often misses the outliers. In other words, even if there exists no $x \in [\underline{x}, \bar{x}]$ with $\mathbf{d}(x, \mathcal{V}_i) \leq \epsilon$, could it be possible that Algorithm 2 fails to provide the outlier certificate for the measurement \mathcal{M}_i ? In fact, this may often happen, especially when the interval gap is large. However, when the gap shrinks towards zero during the BnB search, such occurrences are less and less likely to happen. This can be inferred from the extreme condition of $\epsilon = 0$ and $\hat{x} = \underline{x} = \bar{x}$. In such case, the polynomial $p_b(x)$ becomes an SoS, of the form $p_b(x) = \|x - \hat{x}\|^2$. The d-SoS certificate $f(x)$ of $p_b(x)$, in (2), is $p_b(x)$ itself.

3.2 The BnB Algorithm

The goal of the BnB algorithm is to estimate parameters $x \in \mathbb{R}^n$ that yield the largest number of inlier measurements. We start with a set $\mathcal{S} = \{\mathcal{P}_i\}_{i=1}^m$, where the set of polynomials \mathcal{P}_i defines a variety \mathcal{V}_i for each measurement \mathcal{M}_i . During BnB search, a dynamic search tree, whose nodes are parameter intervals, is built to explore the space

of parameters. Given measurements and intervals, the BnB algorithm (see Algorithm 3) requires the estimation of both optimistic and pessimistic numbers of inliers.

Optimistic number of inliers: We estimate the optimistic number of inliers by exploiting the outlier test method of the Algorithm 2. Since any measurement that passes the test is definitely an outlier for the considered interval, the optimistic inliers are obtained by simply discarding such outliers from the total measurements.

Pessimistic number of inliers: A local refinement method is used to obtain a pessimistic number of inliers for each node. The local method iteratively refines \mathbf{x} while starting from the mid-values of the parameter intervals. To ensure that the final solution still represents the same node, it searches the optimal solution within the investigated intervals. Given a set of potential inliers $\mathcal{I} \subseteq \mathcal{S}$, the algorithm iteratively updates the parameters to:

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}]} \sum_{\substack{h_{ij}(\mathbf{x}) \in \mathcal{P}_i, \\ \mathcal{P}_i \in \mathcal{I}}} \|h_{ij}(\mathbf{x})\|^2. \quad (8)$$

The pessimistic set of inliers are only those measurements which satisfy the condition $\mathbf{d}(\mathbf{x}^*, \mathcal{V}) \leq \epsilon$. For the given measurement \mathcal{M}_i , following two steps are performed:

- (i) If there exists $\mathbf{z} \in \mathcal{Z}_i$ (sample set \mathcal{Z}_i of \mathcal{V}_i) such that $\mathbf{d}(\mathbf{x}^*, \mathbf{z}) \leq \epsilon$, \mathcal{M}_i is an inlier.
- (ii) The entries of \mathbf{x}^* are used to solve the polynomial system $\mathcal{P}_i = \{h_{ij}(\mathbf{x})\}_{j=1}^l$ on l variables (by replacing the others). If the solution is ϵ close to \mathbf{x}^* , \mathcal{M}_i is an inlier. This step is carried out only if $|h_{ij}(\mathbf{x})| < \eta$ for all $h_{ij}(\mathbf{x}) \in \mathcal{P}_i$ and threshold η .

At any instant of the BnB search, we keep track of the biggest set of inliers obtained so far. Let us call $nOpti$ and $nPess$ the number of optimistic and pessimistic inliers, respectively. Similarly, $bPess$ for the maximum of $nPess$ among all nodes. Any node whose $nOpti$ is worse than $bPess$ is rejected. Otherwise, the node is further branched for its parameter with the largest interval, resulting in two new nodes to be processed. The node corresponding to the $bPess$ is processed first. The algorithm terminates when no node has an $nOpti$ that is better than $bPess$.

Algorithm 3 [$flag, bPess$] = processNode($[\underline{\mathbf{x}}, \bar{\mathbf{x}}], \mathcal{S}, \epsilon, bPess$)

1. Count $nOpti$ using Algorithm 2 for each $\mathcal{P}_i \in \mathcal{S}$ of \mathcal{M}_i .
 2. If $nOpti \leq bPess$, set $flag = \text{"false"}$ (for pruning), return.
 3. Count $nPess$ using local method (8) followed by steps (i) and (ii).
 4. If $bPess < nPess$, then $bPess \leftarrow nPess$, $flag = \text{"true"}$.
-

4 Polynomials for Camera Autocalibration

We consider that a set of m image pairs $\{\mathcal{I}_i, \mathcal{I}'_i\}_{i=1}^m$ are captured by uncalibrated cameras. For each pair, both 3D scene points and cameras are reconstructed up to a projective ambiguity, from the point correspondences between images [30]. Without loss of generality, we choose the world frame such that it coincides with the camera coordinate

frame of the first image. In this case, the projection matrix of the first image becomes $[1_{3 \times 3} | 0_{3 \times 1}]$. Let the projection matrix of the second image be $M_i = [H_i | e_i]_{3 \times 4}$. Given a set of measurements $\mathcal{S} = \{M_i\}_{i=1}^m$, with possibly many outliers, we wish to calibrate cameras by using the method presented in Section 3. To do so, we exploit the formulations of the following two classical camera autocalibration methods.

4.1 Simplified Kruppa's Equations on DIAC

The Kruppa's equations for camera autocalibration rely on an omnipresent line conic lying on the PaI – the so-called the Dual Absolute Conic (DAC). The projection of DAC on the image \mathcal{I}'_i , known as Dual Image of Absolute Conic (DIAC), can be expressed in the form of the camera intrinsics K_i of the camera capturing \mathcal{I}'_i , as,

$$\omega_i = K_i K_i^T. \quad (9)$$

In this work, we assume that the camera intrinsics are constant across all images such that $K = K_i$, thus leading to a unique DIAC ω for all image pairs. The simplified Kruppa's equations [4] allow us to express such ω in the form of polynomials, with the help of Fundamental matrices $F_i = [e_i]_{\times} H_i$.

Let $F_i = U_i D_i V_i$ be the singular value decomposition, with $D = \text{diag}([r_i, s_i, 0])$. For $U_i = [u_{i1} | u_{i2} | u_{i3}]$ and $V_i = [v_{i1} | v_{i2} | v_{i3}]$, two independent polynomials of simplified Kruppa's equations are of the form:

$$\begin{aligned} h_{i1}(\omega) &= (r_i s_i v_{i1}^T \omega v_{i2})(u_{i2}^T \omega u_{i2}) + (r_i^2 v_{i1}^T \omega v_{i1})(u_{i1}^T \omega u_{i2}), \\ h_{i2}(\omega) &= (r_i s_i v_{i1}^T \omega v_{i2})(u_{i1}^T \omega u_{i1}) + (s_i^2 v_{i2}^T \omega v_{i2})(u_{i1}^T \omega u_{i2}). \end{aligned} \quad (10)$$

Given a set of projection matrices $\mathcal{S} = \{M_i\}_{i=1}^m$ as measurements, the task of robust camera autocalibration is to estimate ω that maximizes the consensus of these measurements. Note that ω is a 3×3 matrix with $\omega = \omega^T$ and $\omega_{(3,3)} = 1$. Therefore, one can linearly parameterize ω using only a vector $x \in \mathbb{R}^5$. Let $h_{i1}(x)$ and $h_{i2}(x)$ be the equivalent representations of $h_{i1}(\omega)$ and $h_{i2}(\omega)$ of (10), respectively. For each measurement $M_i \in \mathcal{S}$, we derive two polynomials $\mathcal{P}_i = \{h_{i1}(x), h_{i2}(x)\}$ defining a variety $\mathcal{V}_i := \{x \in \mathbb{R}^5 : h_{i1}(x) = 0, h_{i2}(x) = 0\}$. Now, we cast the task of autocalibration as Problem 2, for a set $\mathcal{S} = \{\mathcal{P}_i\}_{i=1}^m$ and a threshold ϵ , to estimate ω parameterized by $x \in \mathbb{R}^5$. This problem is solved by using our solution proposed in Section 3 for both ω and the largest inlier set ζ , simultaneously. The intrinsics K are then recovered by performing the Cholesky decomposition on ω .

4.2 Modulus Constraints on PaI

Modulus constraints [7] allow us to derive polynomials parameterized by the coordinates of the Plane-at-Infinity (PaI) – the plane supporting the Absolute Conic – with the help of the homography between two images induced by any arbitrary plane. The estimation of the PaI is a necessary condition to upgrade projective reconstructions to

affine. Once the upgrade is performed, the task of camera calibration boils down to a linear problem [30]. Such methods fall into the category of stratified autocalibration.

In this work, we assume that all the projection matrices M_i are registered to a common coordinate frame. In practice, this can be achieved in several ways: by joint projective factorization [31, 32]; by registering projection matrices using projective homographies [30]; or simply by choosing a fixed image as the reference for all others. Without loss of generality, we assume that M_i are obtained using the latter. Under such circumstances, there exists a unique Plane-at-Infinity, say Π_∞ , common to all the views [5]. Let $\Pi_\infty = (\pi_\infty^\top, 1)^\top \in \mathbb{R}^4$ be the coordinate vector of Π_∞ . For $M_i = [H_i | e_i]$, the homography between pairs \mathcal{I}_i and \mathcal{I}'_i via Π_∞ can be expressed as,

$$H_{i\infty} = H_i - e_i \pi_\infty^\top. \quad (11)$$

Given M_i , the theory of Modulus constraints relies on the fact that the homography $H_{i\infty}$ must have all three eigenvalues with the same moduli, to recover the unknown π_∞ . In [7], *Pollefeys et Van Gool* have shown that the moduli constraint can indeed be expressed as quartic polynomials, parameterize by $\pi_\infty \in \mathbb{R}^3$. At this point, we borrow four linear functions $l_0(\pi_\infty), \dots, l_3(\pi_\infty)$, from [7] (please, refer [7] for their definitions). Now, the modulus constraints are of the form:

$$h_i(\pi_\infty) = l_{i3}(\pi_\infty)(l_{i1}(\pi_\infty))^3 - l_{i0}(\pi_\infty)(l_{i2}(\pi_\infty))^3. \quad (12)$$

While searching for Π_∞ , we make use of projection matrices $\mathcal{S} = \{M_i\}_{i=1}^m$ as measurements, such that the sought solution maximizes the consensus within \mathcal{S} . For each $M_i \in \mathcal{S}$, we derive a polynomial $\mathcal{P}_i = \{h_i(\pi_\infty)\}$ defining a variety $\mathcal{V}_i := \{\pi_\infty \in \mathbb{R}^3 : h_i(\pi_\infty) = 0\}$. Now, we cast the task of finding Π_∞ as Problem 2, for a set $\mathcal{S} = \{\mathcal{P}_i\}_{i=1}^m$ and a threshold ϵ , to estimate $\pi_\infty \in \mathbb{R}^3$. This problem can be solved by using our solution of Section 3 for both π_∞ and the largest inlier set ζ , simultaneously.

5 Discussion

Although we focus on the autocalibration for cameras with constant intrinsics, the solution proposed for estimating PaI does not require this assumption. Note that, a projective reconstruction obtained from cameras with different intrinsics still share a common PaI. Therefore, once the PaI is estimated, one may exploit this information for calibrating cameras with variable focal lengths similar to [33].

The main concern of this paper is finding good initial bounds on the sought parameters (see Algo. 3). Fortunately, in the context of camera calibration, one can safely assume that a vague guess on camera intrinsics can be known, except for close to affine cameras. For affine (or close to affine) cameras, the task of autocalibration is considered to be rather a simpler problem [34], hence can be attempted accordingly.

In our experiments, we consider that the focal length lies within [1 10] interval relative to the image size, aspect ratio lies between 0.7-1.25, principal points lie around image center within a radius of $(\frac{1}{4})^{th}$ of the image size, and the skew is close to zero. With these assumptions, we derive bounds on DAIC using the interval analysis arithmetics [35]. While deriving the bounds on PaI, we assume that the distance of PaI is

smaller than 5 units from the reference camera frame, in the normalized projective reconstruction. The projective reconstruction is obtained by normalizing the reference image to 1×1 . Note that, the coordinate of PaI, π_∞ is the product of its distance d and the normal \hat{n} . Since $\|\hat{n}\| = 1$, the coordinates π_∞ can also be bounded.

6 Experiments

We conducted several experiments with nine different real datasets to test the robustness and optimality of our method. These datasets are, Fountain [36], Herz-Jesu [36], Dino4983 [37], DinoColmap [37], House [37], Courtyard [38], CherubColmap [39], Vercingetorix [32], and Watertower [32]. All reported experiments were conducted on real datasets. The projective reconstruction required for our method was obtained using [40] for Fountain and Herz-Jesu, after piece-wise factorization and registration with projective homography. The projective reconstructions for all other datasets were obtained using [32]. For the experiments with quantitative evaluations, only the outliers were synthetically generated. These synthetic outliers were added either on Fountain or Herz-jesu dataset, by arbitrarily selecting them for any set of experiments, since these two are also the datasets with known ground truth DIAC and PaI. Our algorithm is implemented in MATLAB2015a and all the optimization problems are solved using MOSEK [41]. All experiments were carried out on a 16GB RAM Pentium i7/3.40GHz.

To evaluate the calibration quality, five different error measurement metrics were defined: the RMS error on 3D reconstruction, errors in focal length Δf , principal point Δuv , skew Δs and PaI $\Delta \pi_\infty$, similarly as in [12]. The 3D reconstruction error is computed on the normalized pointsets (with mean radius of $\sqrt{3}$), error for camera intrinsics are computed for 1×1 image size, and error for PaI is computed for its normal vector.

6.1 Simplified Kruppa’s Equations

To observe the behavior of proposed algorithm with simplified Kruppa’s equations, we generated multiple image pairs with dominating number of outliers. Then, we tracked the number of pessimistic and optimistic inliers for increasing BnB iterations, along with the number of nodes remaining to be processed and the volume of the parameter space yet to be explored. For almost all our experiments, it has been observed that the algorithm converges before 1000 iteration, while limiting itself to a reasonable number of nodes across search iterations (showing only a small memory usage). One of these observations is shown in Fig. 1. Note from Fig. 1(left) that our outlier detection method of Algorithm 2 comes into play, to help punning, soon after the algorithm starts. Our algorithm also finds the optimal solution only in a few iterations, while most of the search is performed to obtain the optimality certificate. In Fig. 1 (middle), we show the number of nodes and search volume remaining for the same experiment.

We report the computation time of our method in Fig 1 (right), for two different sets of inlier pairs with increasing percentage of outliers. Note that these experiments are conducted with as high as 90% of outliers. Even in such extreme cases, our algorithm successfully results the optimal solution, with optimality certificate, within a reasonable amount of time. Here, 10 inliers with 90% outliers refers to the total of 100 image pairs.

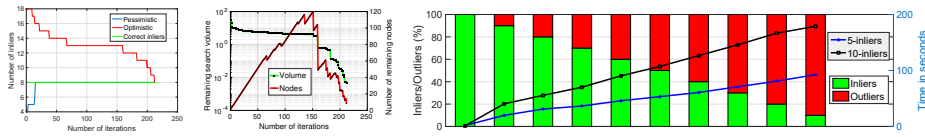


Fig. 1. Convergence graph (left) and remaining nodes/volume during BnB search (middle). Time taken for increasing % outliers with two different cases of fixed inliers (right). Experiments with Kruppa’s equations.

In Fig. 2, we report the results obtained by our method and a randomly started local method for 100 independent experiments. We use the well known Mendonça-Cipolla [14] for local method as it was designed to be robust towards noise. In each experiment, the Mendonça-Cipolla method was started at randomly picked camera intrinsics lying within the initial intervals. The results show that, unlike Mendonça-Cipolla which generates a very low number of inliers and provides very large 3D errors, our method consistently detects the same number of inliers with the same 3D error.

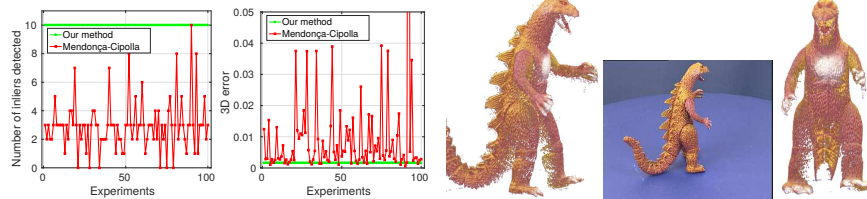


Fig. 2. Left-middle: global vs. local method for 100 experiments with 10 inliers and 20 outliers. Number of detected inliers (left) and 3D reconstruction error (middle). Right: reconstruction of Dino, obtained using our estimated camera intrinsics. One of the input images with two views of the reconstruction. Experiments with Kruppa’s equations.

We further compared the results of our method against that of one local and three global methods for autocalibration: a local and practical method from [8], a DAQ rank-3 constrained direct method from [11], a LMI-based direct method from [12] and a stratified method from [9]. All of these methods assume that the input projective reconstruction is free of outliers. Therefore, we conducted experiments with datasets with no outliers. In Tab. 1, we provide the quantitative results for calibration accuracy, by computing errors in intrinsics and PaI. For the DIAC obtained from Kruppa’s equations, we extract camera intrinsics using Choleskey decomposition, whereas the PaI is obtained linearly using the DAQ projection equation. Tab. 1 shows that our method is very competitive in terms of accuracy and generally faster in terms of speed when compared to global methods without outliers. Note that there exists no method which is globally optimal as well as robust to outliers for the task at hand.

In Tab. 2, more experiments with real data in the presence of outliers is provided. Here, we report the number of image pairs processed, and the number inlier pairs de-

ected by our method. In the same table, inliers detected by Mendonça-Cipolla method, for the same inlier threshold, is also provide for comparison, along with two other methods discussed later in this paper. For all four methods, their run time is also reported. Although the compared methods of Tab. 2 are faster, they are only locally optimal and thus do not always provide the globally optimal solutions. As expected, our method consistently detects more (or equal) number of inliers than Mendonça-Cipolla method. To qualitatively evaluate our calibration results, we provided our estimated intrinsics to the calibrated reconstruction framework of [42], whose dense reconstruction is shown in Fig 2 (right). The reconstructions for two more datasets obtained via projective-to-metric upgrade using our method are shown in Fig. 3. Figs. 2 and 3 demonstrate that our estimated inlier set provides meaningful metric reconstruction. This also ensures that the obtained inliers are indeed true inliers. Unfortunately, 3D models for all dataset of Tab. 2 are not available to compute the 3D RMS error.

Dataset	Method	Δf	Δuv	Δs	$\Delta \pi_\infty$	Time(s)	3D err.
Fountain (11-views)	Practical [8]	0.0117	0.0149	0.0037	0.0080	0.36	0.0003
	Stratified [9]	0.0777	0.0969	0.0125	0.0455	388.24	0.0083
	Rank-3 Direct [11]	0.0100	0.0147	0.0044	0.0075	5.75	0.0003
	LMI Direct [12]	0.0506	0.0269	0.0024	0.0199	156.88	0.0018
	Kruppa (ours)	2.93e-05	0.0069	3.23e-05	0.0009	1.88	0.0001
	Modulus (ours)	0.0368	0.0085	0.0033	0.0033	13.71	0.0012
Herz-jesu (8-views)	Practical [8]	0.0017	0.0113	0.0068	0.0006	0.36	0.0004
	Stratified [9]	0.7231	0.4462	0.3232	0.0960	380.72	0.0432
	Rank-3 Direct [11]	0.0026	0.0096	0.0069	0.0008	16.54	0.0003
	LMI Direct [12]	0.0138	0.0086	0.005	0.005	115.61	0.0008
	Kruppa (ours)	4.46e-05	0.0069	3.40e-05	0.0003	0.53	0.0004
	Modulus (ours)	0.0103	0.0070	0.0047	0.0009	52.12	0.0003

Table 1. Our method vs. one local [8] and three global [9, 11, 12] autocalibration methods. Two real datasets without outliers. Outlier-free data is necessary for [8], [9], [11] and [12].

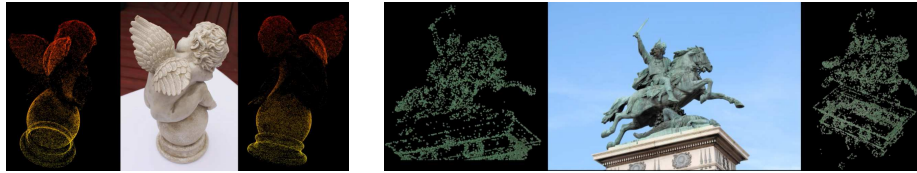


Fig. 3. Sample image and two views of 3D reconstruction of Cherubim (left) and Vercingetorix (right) obtained via projective-to-metric upgrade using our method with Kruppa's equations.

6.2 Modulus Constraints

Similar to the case of Kruppa's equations, we also monitored the optimistic/pessimistic inliers as well as the remaining volume/node with the evolution of the BnB search. As expected, we observed that the Algorithm 2 starts rejecting outliers efficiently only after

a few iterations of BnB. This is attributed to the weaker initial bounds on PaI, and the higher degree of polynomials of Modulus constraints. For an example experiment, the convergence graph and node/volume information is shown in Fig 4.

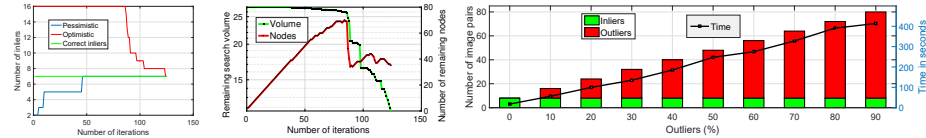


Fig. 4. Convergence graph (left) and remaining nodes/volume during BnB search (middle). Time taken for increasing % of outliers and fixed inliers (right). Our method with Modulus constraints.

In Fig 4, we also show the time taken for our method for Modulus constraints, for a fixed number of inlier image pairs and increasing number of outliers. To test the robustness, we varied the number of outliers up to 90% and compared the results against an in-house RANSAC method. Every iteration of the RANSAC, (i) randomly generates a PaI hypothesis within the initial interval of consideration; (ii) selects best three modulus constraint polynomials for given hypothesis (based on their residuals) and locally refines the hypothesis similarly for (8); (iii) collects the consensus among all the polynomials using the refined hypothesis. The maximum number of iterations for RANSAC is chosen such that it takes about the same time as our method. Fig. 5 (left) shows that our method consistently detects 8 inliers for all the experiments, while RANSAC fails to detect number of correct inliers, starting from 40% of outliers. The numbers of inliers reported are true-positive inliers. Our method does not detect any false positive inlier. In rest of the plots of Fig. 5, we show errors in focal length (second from left), PaI (third from left), and 3D reconstruction (right). Using the estimated PaI, we linearly solve the DAQ projection equation for DIAC, under the assumption of constant intrinsics. If the obtained DIAC is far from being a positive definite matrix, the intrinsics cannot be recovered. We consider such cases as calibration failure. The Missing entries for RANSAC in Fig. 5 refer to such failures.

Our results for Modulus constraints are also compared to a direct method from [12] and a stratified method from [9], in Tab. 1 alongside with our results for Kruppa’s equations. Note that both of our methods perform better than [12] and [9], in terms of time as well as accuracy. In Tab. 2, time and inliers detected by our method with Modulus constraints are provided. In the same table, inliers detected by RANSAC, for the same inlier threshold, is also provide for comparison.

From our experiments, we observed that the reconstruction obtained from [32] preserves a unique PaI valid for most of the views. However, the projection matrices do not really respect the constraint of constant intrinsics. This is not really surprising since cameras are allowed (or even expected) to have different intrinsics, during reconstruction. This makes the camera calibration with constant intrinsics assumption a difficult problem, since many polynomials derived from Kruppa’s equations are not satisfied anymore. This can be observed in Table 2, from the difference between detected inliers for Kruppa and Modulus methods.

Dataset	N	Kruppa (ours)		Mendonça-Cipolla		Modulus (ours)		Modulus-RANSAC	
		Inliers	Time(s)	Inliers	Time(s)	Inliers	Time(s)	Inliers	Time(s)
CherubColmap	20	9	28.76	3	0.49	20	118.51	7	76.92
Courtyard	43	11	81.51	11	0.36	43	29.06	43	0.92
Dino4983	11	8	104.55	6	1.02	11	65.98	11	3.09
DinoColmap	11	8	135.65	4	0.32	11	21.97	11	3.33
House	8	7	22.73	2	0.35	8	387.59	8	7.61
Vercingetorix	22	11	50.02	7	0.79	21	262.43	18	75.34
Watertower	36	22	70.81	22	0.28	36	342.44	36	25.80

Table 2. Real data with outliers. Total number of image pairs processed (N), number of inliers detected, and time take for our method with Kruppa’s equations, Mendonça-Cipolla [14], our method with Modulus constraints, and a in-house RANSAC on Modulus constraints.

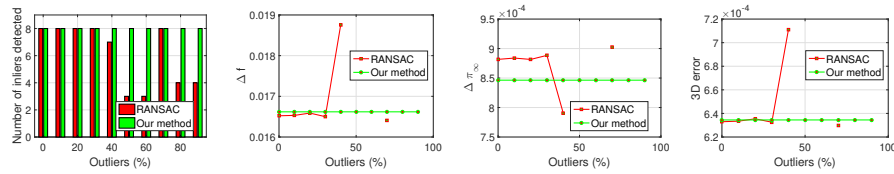


Fig. 5. Global vs. RANSAC with outliers. Left to right: number of detected inliers; focal length error; plane-at-infinity error; 3D reconstruction error. Experiments with Modulus constraints.

Our experiments highlight the robustness of our method, which is also the main focus of this paper. The scaling of our method with image pairs and outliers can be seen in Fig. 1 and Fig. 4. Since the algorithm explores the parameters space, the performance depends upon dimensionality of the problem, number of measurements, outlier ratio, and the initial bound gap of the parameters. Therefore, a trade-off between them is necessary for a better scaling.

7 Conclusion

In this paper, we presented a generic framework of consensus maximization for polynomials. The proposed framework was applied to obtain the consensus among polynomials parameterized by DIAC or PaI, which appear during camera autocalibration. We showed with several experiments that our algorithm can calibrate cameras even when an overwhelmingly high number of camera motions are incorrectly estimated. Moreover, the proposed method not only detects the inlier/outlier camera motions correctly, but also results accurate estimation of DIAC and PaI, when searched with the polynomials derived from Kruppa’s equations or Modulus constraints. Our framework has potential to be applied in many other Computer Vision problems.

Acknowledgements

This research was supported by the Swiss Commission for Technology and Innovation under the CTI project EXASOLVED, Grant No. 26253.1 PFES-ES.

References

1. Maybank, S.J., Faugeras, O.D.: A theory of self-calibration of a moving camera. *International Journal of Computer Vision* **8**(2) (1992) 123–151
2. Luong, Q., Faugeras, D.: Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision* **22** (1997) 261–289
3. Seo, Y., Heyden, A., Cipolla, R.: A linear iterative method for auto-calibration using the dac equation. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Volume 1., IEEE (2001) I–I*
4. Lourakis, M.I., Deriche, R.: Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondences to 3D measurements. PhD thesis, INRIA (1999)
5. Hartley, R.I., Hayman, E., de Agapito, L., Reid, I.: Camera calibration and the search for infinity. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. Volume 1., IEEE (1999) 510–517*
6. Nistér, D.: Untwisting a projective reconstruction. *International Journal of Computer Vision* **60**(2) (2004) 165–183
7. Pollefeys, M., Van Gool, L.: Stratified self-calibration with the modulus constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(8) (1999) 707–724
8. Gherardi, R., Fusiello, A.: Practical autocalibration. *Computer Vision–ECCV 2010 (2010) 790–801*
9. Chandraker, M., Agarwal, S., Kriegman, D., Belongie, S.: Globally optimal affine and metric upgrades in stratified autocalibration. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE (2007) 1–8*
10. Triggs, B.: Autocalibration and the absolute quadric. In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, IEEE (1997) 609–614*
11. Chandraker, M., Agarwal, S., Kahl, F., Nistér, D., Kriegman, D.: Autocalibration via rank-constrained estimation of the absolute quadric. In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE (2007) 1–8*
12. Habed, A., Pani Paudel, D., Demonceaux, C., Fofi, D.: Efficient pruning lmi conditions for branch-and-prune rank and chirality-constrained estimation of the dual absolute quadric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 493–500*
13. Fusiello, A., Benedetti, A., Farenzena, M., Busti, A.: Globally convergent autocalibration using interval analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(12) (2004) 1633–1638
14. Mendonça, P.R., Cipolla, R.: A simple technique for self-calibration. In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference On. Volume 1., IEEE (1999) 500–505*
15. Heikkilä, J.: Using sparse elimination for solving minimal problems in computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 76–84*
16. Barath, D., Toth, T., Hajder, L.: A minimal solution for two-view focal-length estimation using two affine correspondences. (2017)
17. Sturm, P.: On focal length calibration from two views. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Volume 2., IEEE (2001) II–II*
18. Heyden, A., Åkström, K.: Minimal conditions on intrinsic parameters for euclidean reconstruction. In: *Asian Conference on Computer Vision, Springer (1998) 169–176*

19. Verschelde, J.: Algorithm 795: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)* **25**(2) (1999) 251–276
20. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization* **11**(3) (2001) 796–817
21. Parrilo, P.A.: Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD thesis, California Institute of Technology (2000)
22. Schweighofer, G., Pinz, A.: Globally optimal $O(n)$ solution to the pnp problem for general camera models. In: *BMVC*. (2008) 1–10
23. Gurdjos, P., Bartoli, A., Sturm, P.: Is dual linear self-calibration artificially ambiguous? In: *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE (2009) 88–95
24. Campbell, D., Petersson, L., Kneip, L., Li, H., Huang, C.T., Kim, K., Gu, J., Tyree, S., Molchanov, P., Niessner, M., et al.: Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In: *Proceedings of the IEEE International Conference on Computer Vision*. (2017)
25. Pani Paudel, D., Habed, A., Demonceaux, C., Vasseur, P.: Robust and optimal sum-of-squares-based point-to-plane registration of image sets and structured scenes. In: *Proceedings of the IEEE International Conference on Computer Vision*. (2015) 2048–2056
26. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming* **96**(2) (2003) 293–320
27. Cifuentes, D., Parrilo, P.A.: Sampling algebraic varieties for sum of squares programs. arXiv preprint arXiv:1511.06751 (2015)
28. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu with permanent doi: [dx.doi.org/10.7274/R0H41PB5](https://doi.org/10.7274/R0H41PB5)
29. Verschelde, J.: Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. Preprint (1997)
30. Hartley, R., Zisserman, A.: *Multiple view geometry in computer vision*. Cambridge university press (2003)
31. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. *Computer VisionECCV'96* (1996) 709–720
32. Magerand, L., Del Bue, A.: Practical projective structure from motion (p2sfm). In: *Computer Vision, 2017. The Proceedings of IEEE International Conference on*. (2017) 39–47
33. Pollefeys, M., Van Gool, L., Proesmans, M.: Euclidean 3d reconstruction from image sequences with variable focal lengths. *Computer VisionECCV'96* (1996) 31–42
34. Quan, L.: Self-calibration of an affine camera from multiple views. *International Journal of Computer Vision* **19**(1) (1996) 93–105
35. Alefeld, G., Mayer, G.: Interval analysis: theory and applications. *Journal of computational and applied mathematics* **121**(1) (2000) 421–464
36. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (2008) 1–8
37. : Multi-view and oxford colleges building reconstruction. <http://www.robots.ox.ac.uk/vgg/data/data-mview.html> (2017)
38. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. *Image Analysis* (2011) 524–535
39. : 3dflow srl. 3df zephyr reconstruction showcase. <http://www.3dflow.net/> (2017)
40. Oliensis, J., Hartley, R.: Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(12) (2007) 2217–2233

41. MOSEK, A.: The mosek optimization toolbox for matlab manual, version 8.0. MOSEK ApS, Denmark (2015)
42. : Agisoft photoscan. <http://www.agisoft.com/> (2017)