Minimal Perspective Autocalibration

Andrea Porfiri Dal Cin[1] Timothy Duff[2] Luca Magri[1] Tomas Pajdla[3]
[1] Politecnico di Milano [2] University of Washington [3] CIIRC CTU Prague

Abstract

We introduce a new family of minimal problems for reconstruction from multiple views. Our primary focus is a novel approach to autocalibration, a long-standing problem in computer vision. Traditional approaches to this problem, such as those based on Kruppa's equations or the modulus constraint, rely explicitly on the knowledge of multiple fundamental matrices or a projective reconstruction. In contrast, we consider a novel formulation involving constraints on image points, the unknown depths of 3D points, and a partially specified calibration matrix K. For 2 and 3 views, we present a comprehensive taxonomy of minimal autocalibration problems obtained by relaxing some of these constraints. These problems are organized into classes according to the number of views and any assumed prior knowledge of K. Within each class, we determine problems with the fewest—or a relatively small number of—solutions. From this zoo of problems, we devise three practical solvers. Experiments with synthetic and real data and interfacing our solvers with COLMAP demonstrate that we achieve superior accuracy compared to state-of-the-art calibration methods. The code is available at github.com/andreadalcin/MinimalPerspectiveAutocalibration.

1. Introduction

Autocalibration is the fundamental process of determining intrinsic camera parameters using only point correspondences, without external calibration objects or known scene geometry [11–14, 23, 28, 33, 34, 36, 38, 47, 51].

1.1. Contribution

This paper presents a comprehensive characterization of two- and three-view minimal autocalibration problems in the case of a perspective camera with constant intrinsics. We introduce practical and efficient solvers for minimal autocalibration by introducing a novel formulation that extends the minimal Euclidean reconstruction problem of four points in three calibrated views [24, 40] to the uncalibrated case. Our approach jointly estimates camera intrinsics, encoded in the calibration matrix K, and unknown 3D point depths, and seamlessly integrates any partial knowledge of the camera intrinsics. This gives rise to a variety of two- and three-view minimal autocalibration problems, for which we provide a complete taxonomy in Tab. 1. We develop a general theory of minimal relaxations to address cases where our formulation leads to an over-constrained problem. These minimal relaxations of our depth formulation can be completely enumerated, and each instance of a specific autocalibration problem can be solved offline by applying numerical homotopy continuation (HC) methods to one such relaxation. Crucially, the offline analysis with HC methods also enables us to identify the most efficiently solvable minimal relaxations.

Our practical contributions include implementing a numerical solver for *full* camera calibration, *i.e.*, calibration of all 5 unknown parameters of a perspective camera. We also consider common assumptions—namely, zero-skew and square pixels—and design fast solvers for specialized problems with a partially calibrated camera. These solvers can be fast enough for many online calibration applications, and can also bootstrap solutions using RANSAC-based frameworks with high accuracy in offline calibration settings. Among the strengths of our approach, we avoid well-known degeneracies of Kruppa's equations [48] and recover K directly instead of relying on estimates of the dual image of the absolute conic (DIAC), which may not be positive-semidefinite. Experiments show that our solvers outperform existing autocalibration methods in terms of accuracy in both synthetic and real image sequences despite increased runtime. Interfacing our solvers with COLMAP [44] further highlights the applicability of our approach.

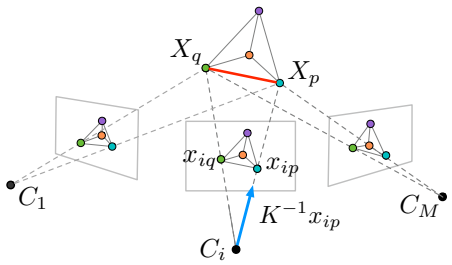

Figure 1. Illustrating the setup of equations (1) and (4).

Thus, our contribution is two-fold: *i*) theoretically, we provide a complete taxonomy of minimal autocalibration problems in 2 or 3 views; *ii*) practically, our novel solvers outperform classical autocalibration approaches in accuracy and are robust against degenerate configurations arising in very practical calibration scenarios when a camera revolves around an object, which is a substantial problem for all methods based on solving Kruppa's equations [49].

1.2. Problem formulation

We recall here some standard constraints on the calibration matrix $K \in \mathbb{R}^{3\times3}$ that involve the images of N 3D points from M different positions, as depicted in Figure 1. We want to estimate the entries of K: focal lengths f and g, principal point (u, v) and camera skew s. Image points are expressed in homogeneous coordinates $x_{ip} \in \mathbb{R}^2 \times \{1\}$, *i.e.*, 3×1 vectors whose third entries equal 1.

In Eq. (1) below and throughout the paper, the letter $i \in [M] := \{1, \dots M\}$ indexes a single image, while $p \in [N] := \{1, \dots, N\}$ indexes a point, $R_i \in \mathbb{R}^{3\times3}$ denotes a rotation matrix, $C_i \in \mathbb{R}^3$ is a camera center, $X_p \in \mathbb{R}^3$ is a 3D point, and $\lambda_{ip} \in \mathbb{R}$ is the depth of the p-th point in the i-th camera [20].

Different flavors of the autocalibration problem exist in practice, depending on the available partial knowledge about the intrinsics in K. For instance, common assumptions are that the camera has square pixels ($f = g$) or zero skew ($s = 0$). In general, we assume that there are L linear equations $f_1(K) = \ldots = f_L(K) = 0$ which encode any partial knowledge of intrinsics in K. For instance, if our camera has square pixels and no skew, then we may take $f_1(K) = s$, $f_2(K) = f - g$.

Thus, assuming no noise in image points, a solution K must satisfy the following conditions:

$$\lambda_{ip}\, x_{ip} = K\, R_i\, (\, I \mid -C_i\,) \begin{pmatrix} X_p \\ 1 \end{pmatrix},\ i \in [M],\ p \in [N],$$
$$K = \begin{pmatrix} f & s & u \\ 0 & g & v \\ 0 & 0 & 1 \end{pmatrix},\quad \begin{matrix} R_i^\top R_i = I, & \det R_i = 1, \\ \zeta_{ip}\lambda_{ip} = 1, & \mu\, f\, g = 1, \\ f_l(K) = 0, & l \in [L]. \end{matrix} \qquad (1)$$

The additional unknowns ζ_{ip} and corresponding equations $\zeta_{ip}\lambda_{ip} - 1$ prohibit spurious solutions with zero depths. Similarly, $\mu\, f\, g - 1 = 0$ ensures that $\det K = f\, g \neq 0$.

1.3. Previous work

We focus on the classical scenario where K is constant across views. For work exploring the non-constant case, *e.g.*, [23, 38] derive minimal conditions on the camera intrinsics for autocalibration. We also note that many works have addressed special cases of autocalibration, such as focal length estimation [1, 41, 46].

General methods fall roughly into two classes.

Direct methods use the so-called *rigidity constraint* encoded in fundamental matrices. In theory, K can be recovered from the knowledge of three fundamental matrices resulting from three different camera motions [11, 34]. Direct methods [19, 30, 53] exploit this observation and recover the intrinsic parameters by solving Kruppa's equations [14, 28]. Methods used to solve these equations vary considerably. In [30], instead of considering a complete, over-constrained system of 6 equations in 5 unknowns, a consensus solution is obtained by solving all 6 of the square subsystems using a HC method. This work has several parallels to ours—namely, its use of HC solvers and the fact that these square subsystems are *minimal relaxations* in the sense of Section 3. The main difference is that their unknowns are the entries of the DIAC. In [53], the over-constrained system of Kruppa's equations is solved with a nonlinear least squares technique; here, good initialization is needed to obtain an accurate estimate. We note that simplified polynomial systems have been derived by exploiting additional assumptions on K [53]. Not all direct methods use Kruppa's equations—in [33], a method analogous to the F4 method for computing Gröbner bases is devised for computing the DIAC. As our experiments illustrate, a common weakness of such direct approaches is that they do not enforce positive-semidefiniteness of the DIAC and hence fail with larger noise that makes the estimated DIAC indefinite.

Certain camera motions give rise to degenerate autocalibration problems [20, Ch. 19], [32], and additional degeneracies may exist for particular methods. For example, the method of [30] also falls short when the optical centers of all cameras lie on a sphere and the optical axes pass through the center of the sphere [49]. Although our approach employs a relaxation procedure analogous to this work, it does not suffer from the same degeneracy. Another limitation of direct methods is that they neglect non-trivial polynomial identities that tuples of compatible fundamental matrices must satisfy [3, 15, 20, 26].

Stratified methods assume that a projective reconstruction is known and stratify the problem into Affine and Euclidean stages. An affine reconstruction can be obtained by estimating the *plane-at-infinity* (PaI); from this, the assumption of constant K allows its entries to be easily retrieved. This idea was pioneered in [21], where chirality constraints are used to estimate the location of the PaI. The PaI can also be located via the so-called *modulus constraints*. Specifically, in [39], this resulted in a system of three quartic polynomials on the coefficient of the PaI.

Rather than using the PaI, the work [51] directly encodes all metric information in terms of the *absolute quadric*, which, once retrieved, allows the intrinsic parameters to be retrieved by Cholesky factorization.

In general, stratified approaches are more robust to noise than direct ones but require good initialization of the PaI.

Thus, some works [4, 5, 13] focus on optimality guarantees exploiting a branch-and-bound framework. Similarly, [17] samples the bounded space of intrinsic parameters to estimate the PaI robustly. Interestingly, [36] presents a branch-and-bound paradigm to solve direct and stratified autocalibration based on sampling algebraic varieties.

2. Our approach

We now outline our approach to autocalibration.

2.1. Depth equations and removing symmetries

In this work, we propose to eliminate camera extrinsics from (1) and use constraints involving the calibration matrix K and depths λ_{ip}. By working with these constraints, we are able to avoid potential issues arising from fundamental matrix compatibility. This approach is also well-suited for constructing new minimal problems.

The main geometric constraint we use is that the Euclidean distance $\|X_p - X_q\|$ between any two 3D points X_p and X_q is the same whether these points are reconstructed from the i-th or the j-th camera, for any $i, j \in [M]$ as depicted in Fig. 1. Expressing each 3D point as $X_p = \lambda_{ip} K^{-1} x_{ip}$, this amounts to the vanishing of the function

$$\begin{aligned} d_{i,j,pq}(\lambda, K; x) :=& (\lambda_{ip}x_{ip} - \lambda_{iq}x_{iq})^T \omega (\lambda_{ip}x_{ip} - \lambda_{iq}x_{iq}) \\ & - (\lambda_{jp}x_{jp} - \lambda_{jq}x_{jq})^T \omega (\lambda_{jp}x_{jp} - \lambda_{jq}x_{jq}), \end{aligned}$$

where $\omega = K^{-T}K^{-1}$ is the image of the absolute conic [20]. Note that $d_{i,j,pq}$ is a polynomial in $\lambda_\bullet$ and $x_\bullet$, and a rational function of K. We parametrize ω as

$$\omega = \begin{pmatrix} \frac{1}{f^\star} & -\frac{s^\star}{f^\star} & \frac{vs^\star - u}{f^\star} \\ -\frac{s^\star}{f^\star} & \frac{1}{g^\star} + \frac{s^{\star 2}}{f^\star} & \frac{us^\star - vs^{\star 2}}{f^\star} - \frac{v}{g^\star} \\ \frac{vs^\star - u}{f^\star} & \frac{us^\star - vs^{\star 2}}{f^\star} - \frac{v}{g^\star} & \frac{v^2}{g^\star} + \frac{(u^2 - v^2 s^\star)^2}{f^\star} + 1 \end{pmatrix}, \tag{2}$$

where $f^\star := f^2$, $g^\star := g^2$, $s^\star := \frac{s}{g}$. This parametrization is motivated by the invariance of ω under substitutions

$$f \to -f, \quad (g, s) \to (-g, -s). \tag{3}$$

Thus, when f, g and s are unknown, solutions to the depth equations typically come in symmetric quadruples, and in pairs if only f or (g, s) are unknown. Substituting (2) into $d_{i,j,pq}$, we may rewrite our main constraint as

$$d_{i,j,pq}(\lambda, \omega; x) = 0. \tag{4}$$

Depending on the minimal problem, (3) may not be the only symmetries present. For example, in the fully calibrated case $L = 5$, our minimal relaxation of four points in three views has 640 solutions that can be grouped into pairs which differ only in the signs of depths in some view.

Remark: Our depth formulation can be viewed as a relaxation of the formulation (1). Thus, if we get a finite number of solutions, they surely give valid solutions to (1)[1].

[1] See SM 6 for more discussion of different formulations.

2.2. Specifying a minimal autocalibration problem

Instead of requiring that (4) holds for all $i, j \in [M]$, $p, q \in [N]$, we consider minimal problems which only require that a subset of these constraints hold. Hence, we will be in a situation similar to *partial visibility* as in [9, 27].

To specify a minimal problem, we consider:

(1) Priors on K: In practical situations, we often either possess or lack knowledge of intrinsics. When we know some intrinsics, we can transform images to normalize their known values to standard ones: $f = 1, g = 1, u = 0, v = 0, s = 0$. Subsequently, we solve for the unknown transformed intrinsics and then recover their original values[2]. We represent the knowledge of intrinsics as a 5-tuple of unknowns `fguvs`. If any intrinsic is known, we replace its unknown with its normalized value. For instance, `f1uv0` indicates that f, u, v are unknown, while $g = 1$, $s = 0$ are known. In the interesting scenario of a camera with square pixels, `ffuvs` encodes that f and g are unknown but equal.

(2) Number of cameras M: For a given 5-tuple `fguvs`, we will see that the minimum number of cameras needed to obtain a minimal problem is either 2 or 3. Hence, we will investigate problems for only 2 and 3 cameras.

(3) Number of points N: For each five-tuple of intrinsics and M cameras, we will consider the least number N of points such that there is a set $\mathbf{g}$ of constraints in (4) providing a minimal problem.

(4) Constraints $\mathbf{g}$: For each triplet $(\texttt{fguvs}, M, N)$, we enumerate all possible subsets of constraints (4) which lead to different minimal problems.

Each four-tuple $(\texttt{fguvs}, M, N, \mathbf{g})$ specifies a candidate minimal problem [8]. Table 1 lists the 80 groups according to $(\texttt{fguvs}, M, N)$. For each group, we list the number $\#\mathbf{g}$ of equivalence classes of constraints leading to a minimal problem and a range of[3] numbers of solutions in ω.

3. Relaxation, Enumeration, and Solving

We now give a more precise description of the taxonomy of minimal autocalibration problems presented in Table 1 and the tools needed to obtain it.

For each pair $(\texttt{fguvs}, M)$, we will determine whether camera calibration is possible and, if so, the minimum number N of points such that there is a subset $\mathbf{g}$ of depth equations (4) providing a minimal problem. First, we determine the number of parameters among `fguvs` that can be estimated from N 3D points seen in M images captured by the same camera with constant K. Then, we determine the minimum number N of 3D points required to solve the perspective autocalibration problem given a pair $(\texttt{fguvs}, M)$.

[2] See SM 7 for more details on the normalization and recovering the corresponding non-normalized values.

[3] For `fguvs`, we checked roughly 20% of the 3313 cases. It is conceivable, but unlikely, that problems with fewer solutions remain unchecked.

Prior on K	M	N	L	Min # sol. in $\mathbb{C}$	Max # sol. in $\mathbb{C}$	# subsys.	$\#\mathbf{g}$	Prior on K	M	N	L	Min # sol. in $\mathbb{C}$	Max # sol. in $\mathbb{C}$	# subsys.	$\#\mathbf{g}$
fguvs	2	-	0	∞	∞	0	0	1g0vs	2	-	2	∞	∞	0	0
fguvs	3	6	0	2985*	1136202*	5852925	3313	1g0vs	3	5	2	29012	315653	1140	8
fguv0	2	-	1	∞	∞	0	0	1g0v0	2	7	3	18	18	1	1
fguv0	3	5	1	2313	2313	190	3	1g0v0	3	5	3	4400	102784	4845	37
fgu0s	2	-	1	∞	∞	0	0	1g00s	2	7	3	24	24	1	1
fgu0s	3	5	1	2058	2058	190	3	1g00s	3	5	3	4480	238544	4845	37
fgu00	2	-	2	∞	∞	0	0	1g000	2	6	4	30	30	1	1
fgu00	3	5	2	9686	33606	1140	8	1g000	3	4	4	668	668	1	1
fg0vs	2	-	1	∞	∞	0	0	11uvs	2	-	2	∞	∞	0	0
fg0vs	3	5	1	2058	2058	190	3	11uvs	3	5	2	57912	201265	1140	8
fg0v0	2	-	2	∞	∞	0	0	11uv0	2	7	3	48	48	1	1
fg0v0	3	5	2	9686	112520	1140	8	11uv0	3	5	3	8940	477080	4845	37
fg00s	2	-	2	∞	∞	0	0	11u0s	2	7	3	36	36	1	1
fg00s	3	5	2	9686	33606	1140	8	11u0s	3	5	3	8786	46192	4845	37
fg000	2	7	3	18	18	1	1	11u00	2	6	4	60	60	1	1
fg000	3	5	3	3884	207664	4845	37	11u00	3	4	4	1336	1336	1	1
f1uvs	2	-	2	∞	∞	0	0	110vs	2	7	3	72	72	1	1
f1uvs	3	5	2	4111	4111	190	3	110vs	3	5	3	16390	85480	4845	37
f1uv0	2	-	2	∞	∞	0	0	110v0	2	6	4	60	60	1	1
f1uv0	3	5	2	29044	100816	1140	8	110v0	3	4	4	1336	1336	1	1
f1u0s	2	-	2	∞	∞	1	1	1100s	2	6	4	60	60	1	1
f1u0s	3	5	2	14760	160190	1140	8	1100s	3	4	4	1336	1336	1	1
f1u00	2	7	3	18	18	1	1	11000	2	5	5	20	20	1	1
f1u00	3	5	3	4400	244544	4845	37	11000	3	4	5	640	640	1	1
f10vs	2	-	2	∞	∞	0	0	ffuvs	2	-	1	∞	∞	0	0
f10vs	3	5	2	24332	86539	1140	8	ffuvs	3	5	1	4617	4617	190	3
f10v0	2	7	3	36	36	1	1	ffuv0	2	-	2	∞	∞	0	0
f10v0	3	5	3	7764	57220	4845	37	ffuv0	3	5	2	16188	119119	1140	8
f100s	2	7	3	18	18	1	1	ffu0s	2	-	2	∞	∞	0	0
f100s	3	5	3	4392	102778	4845	37	ffu0s	3	5	2	29028	100758	1140	8
f1000	2	6	4	30	30	1	1	ffu00	2	7	3	24	24	1	1
f1000	3	4	4	668	668	1	1	ffu00	3	5	3	4484	176992	4845	37
1guvs	2	-	1	∞	∞	0	0	ff0vs	2	-	2	∞	∞	0	0
1guvs	3	5	1	4360	4360	190	3	ff0vs	3	5	2	38700	134352	1140	8
1guv0	2	-	2	∞	∞	0	0	ff0v0	2	7	3	24	24	1	1
1guv0	3	5	2	29046	100808	1140	8	ff0v0	3	5	3	4484	92336	4845	37
1gu0s	2	-	2	∞	∞	0	0	ff00s	2	7	3	36	36	1	1
1gu0s	3	5	2	29024	100718	1140	8	ff00s	3	5	3	7756	396042	4845	37
1gu00	2	7	3	36	36	1	1	ff000	2	6	4	30	30	1	1
1gu00	3	5	3	7760	43315	4845	37	ff000	3	4	4	668	668	1	1

Table 1. **80 groups of minimal autocalibration problems** for $M \in \{2, 3\}$ views. For each triplet $(\texttt{fguvs}, M, N)$ we list: *i*) L, the number of linear constraints on K, *ii*) the minimum and maximum solution count in $\mathbb{C}$, *iii*) #subsys. $= \binom{(M-1)\binom{N}{2}}{M+N-L+4}$, the number of square subsystems of (4), *iv*) $\#\mathbf{g}$, the number of inequivalent minimal relaxations. The numbers $\#\mathbf{g}$ and Min # sol. are most important, as they measure the number of minimal relaxations and the complexity of solving them. Solution counts refer to unknown depths and the parameters of ω in (2)–(4). The two counts $\bullet^*$ are only conjectural extrema, due to the prohibitive time needed to check all cases.

Infeasible cases. In general, for K unknown and non-constant, the reconstruction of N 3D points from M views can be obtained only up to a projective transformation H, which has 15 degrees of freedom. Additional constraints on H may allow us to assume H is a similarity transformation with 7 degrees of freedom. For $M = 2$ views, the assumption that K is constant puts 5 constraints on H. Thus, we need $L \geq 15 - 7 - 5 = 3$ linear constraints on K to obtain a Euclidean reconstruction and hence recover the full K.

To determine the minimum number N of 3D points required to solve the perspective camera autocalibration problem as a function of a pair $(\texttt{fguvs}, M)$, we must ensure that the number of degrees of freedom in image measurements is at least the number of degrees of freedom in the unknown scene and cameras. For this purpose, the full formulation (1) is preferable to the equations we actually use for solving, namely (4). This is because we can rigorously employ a count similar to that given in [8, §5]: we should assume there are at least

$$L \geq 3N + 6M - 2 - 2MN \tag{5}$$

independent linear constraints on K in order to solve the autocalibration problem up to a finite number of candidate solutions. Noting also the trivial upper bound $L \leq 5$, this explains the values of L appearing in Table 1. The infeasible cases where $M = 2$ and $L \leq 2$ have already been addressed above. The remaining cases are accounted for by (5) and the rows of Tab. 1. This table indicates that at least one minimal relaxation for the potentially feasible choices of (M, N, L) actually exists. To properly interpret the table, we must now formalize what we mean when we say a subsystem $\mathbf{g}$ of equations determines a minimal relaxation of the autocalibration problem (1).

3.1. Minimal problems and minimal relaxations

Many estimation problems in vision can be expressed using the language of algebraic geometry. In general, we may consider an irreducible algebraic variety X, whose points consist of problem-solution pairs $(\mathbf{p}, \mathbf{x}) \in \mathbb{C}^m \times \mathbb{C}^n$ satisfying some set of equations depending polynomially on $\mathbf{p}$ and $\mathbf{x}$. Our task is to estimate the solution $\mathbf{x} \in \mathbb{C}^n$ given some problem instance $\mathbf{p} \in \mathbb{C}^m$, meaning $(\mathbf{p}, \mathbf{x}) \in X$.

More specifically, image points $\mathbf{p} = (x_{ip})_{i\in[M],p\in[N]}$ specify an instance of an autocalibration problem. We want to estimate the unknowns $\mathbf{x}$ defining ω in (2) and the (suitably normalized) depths (λ_{ip}). Thus $m = 2MN$ and $n = (5 - L) + MN - 1$. If we define the variety X to be the image of a rational map (much like the *joint camera map* of [8, §4]), the condition that X is irreducible holds.

Let π denote the map which projects X into the space of problem instances $\mathbb{C}^m$, *i.e.*,

$$\pi : X \to \mathbb{C}^m, \quad (\mathbf{p}, \mathbf{x}) \mapsto \mathbf{p}. \tag{6}$$

The set of solutions of some problem instance $\mathbf{p} \in \mathbb{C}^m$ may

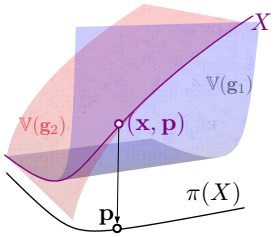

Figure 2. Minimal relaxations $\mathbb{V}(\mathbf{g}_1)$, $\mathbb{V}(\mathbf{g}_2)$, w/ $X = \mathbb{V}(\mathbf{g}_1, \mathbf{g}_2)$, $\mathbf{g}_1(p_1, p_2, x) = x^2 - p_1$, and $\mathbf{g}_2(p_1, p_2, x) = p_2 x^2 - 1$.

be identified with the fiber $\pi^{-1}(\mathbf{p})$. Following [8], we say that π defines a *minimal problem* if the following hold:

1. The problem is *balanced*—that is, $\dim X = m$.
2. Almost every problem instance in $\mathbb{C}^m$ has a solution—equivalently, the image of the map π is dense in $\mathbb{C}^m$.

In practice, we check that a problem is minimal using some system of equations $\mathbf{g}(\mathbf{p}, \mathbf{x}) = 0$ defining X locally, via the following *rank conditions* at a point $(\mathbf{p}_0, \mathbf{x}_0) \in X$:

$$\operatorname{rank} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{p}} \mid \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right) \Big|_{(\mathbf{p}_0, \mathbf{x}_0)} = \operatorname{rank} \left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right) \Big|_{(\mathbf{p}_0, \mathbf{x}_0)} = n. \quad (7)$$

Some of the cases appearing in Table 1 are already minimal problems. These are precisely the rows where both sides of (5) are equal. In general, $\dim X = 3N + 6M - 2 - L$.

When the inequality (5) is strict, we expect the autocalibration problem (1) to be *overconstrained* in the sense that a generic problem in $\mathbb{C}^m$ does not have an exact solution.

To deal with overconstrained problems, consider a system $\mathbf{g}$ consisting of n polynomial or rational functions vanishing on X—that is, $X \subset \mathbb{V}(\mathbf{g})$ where

$$\mathbb{V}(\mathbf{g}) = \overline{\{(\mathbf{p}, \mathbf{x}) \in \mathbb{C}^{m+n} \mid \mathbf{g}(\mathbf{p}, \mathbf{x}) \text{ is defined and equals } 0\}}$$

($\overline{\cdot}$ denotes the Zariski closure [6, §4.4].) If the rank conditions (7) hold at a generic point $(\mathbf{p}_0, \mathbf{x}_0) \in X$, we say that $\mathbf{g}$ determines a *minimal relaxation* of π. Figure 2 illustrates this definition on a simple example (see SM 9 for details).

In general, an overconstrained problem can have different minimal relaxations. In the next section, we obtain a combinatorial classification of all minimal relaxations obtained from subsets of the depth constraints (4), grouping minimal relaxations into natural equivalence classes.

3.2. Enumerating Minimal Relaxations

We now explain how to obtain minimal relaxations of autocalibration problems using the depth equations (4). The combinatorial structure of minimal relaxations obtained by removing a subset of equations (4) is neatly encoded by a *4-coloring*: that is, a function $c\colon \binom{[N]}{2} \to \{B, R, G, W\}$, which assigns one of four colors to all pairs of 3D points. In standard graph-theoretic terminology, these are exactly the improper edge 4-colorings of the complete graph K_N.

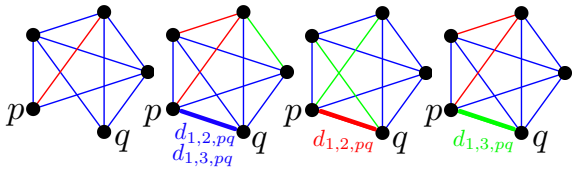

Figure 3. **Non-isomorphic 4-colorings** when $L = 2$ in 3 views. 3D point pairs (p, q) are colored according to the removal of depth equations $d_{i,j,pq}$ in the relaxed subsystem. W (white) indicates removal from both image pairs, B no removal. R and G indicate removal in image pairs (1,3) and (1,2), respectively.

Every 4-coloring determines a subsystem of equations (4)—for each edge $pq \in \binom{[N]}{2}$, we take equations $\mathbf{g}$ in the set

$$\{d_{1,2,pq} \mid c(pq) \in \{B, R\}\} \cup \{d_{1,3,pq} \mid c(pq) \in \{B, G\}\}. \quad (8)$$

We say two 4-colorings $c_1, c_2 : \binom{[N]}{2} \to \{B, R, G, W\}$ are *isomorphic* if there exist permutations $\sigma : [N] \to [N]$ and $\tau : \{B, R, G, W\} \to \{B, R, G, W\}$ such that $\tau(W) = W$, $\tau(B) = B$, and $c_2 = \tau \circ c_1 \circ \sigma$. The minimal relaxations determined by isomorphic 4-colorings are equivalent, since τ corresponds to swapping views 2 and 3, and σ corresponds to relabeling world points. Fig. 3 shows an example.

Determining isomorphism classes of 4-colorings, *i.e.*, and thus equivalence classes of minimal relaxations $\mathbf{g}$, offers us a key practical advantage. Given the large number of 4-colorings, exceeding 5 million for `fguvs`, computing the solution count for all associated problems is computationally prohibitive. Consequently, we opt to consider only one representative per isomorphism class when computing solutions offline with HC. This approach facilitates the creation of the comprehensive taxonomy outlined in Tab. 1. We determine a unique representative c in each isomorphism class using the *line graph* $\mathcal{L}(c)$, as detailed in 10 of the SM.

3.3. Solving with homotopy continuation

For any system $\mathbf{g}(\mathbf{p}, \mathbf{x}) = 0$ encoding a minimal relaxation of an autocalibration problem, we construct minimal solvers using a standard online/offline approach based on numerical HC methods. In the offline stage, we construct a synthetic solution $(\mathbf{p}_0, \mathbf{x}_0) \in X \supset X_{\mathbf{g}}$ by fabricating a 3D scene. If $\mathbf{g}$ arises from a balanced problem, we check that it is minimal via the rank conditions (7), and use monodromy heuristics [7] to recover (with high probability) all remaining solutions in $\pi^{-1}(\mathbf{p}_0, \mathbf{x}_0)$ for the synthetic parameters $\mathbf{p}_0 \in \mathbb{C}^m$. As postprocessing, we use parameter homotopy [45, Ch. 8] with equations $\mathbf{g}$ to track all solutions to new parameter values $\mathbf{p}_1 \in \mathbb{C}^m$ whose coordinates are random complex numbers. Finally, in the online stage, the solver receives a new problem instance $\mathbf{p}_2 \in \mathbb{R}^m$ as input, and uses parameter homotopy to track all solutions for $\mathbf{p}_1$ to those for $\mathbf{p}_2$.

4. Experiments

We evaluate the performance of our proposed minimal solvers on simulated and real image sequences, with a focus on three of the most practical cases: *i*) `ffuv0`, an uncalibrated camera with square pixel aspect ratio and zero-skew, *ii*) `fguv0`, an uncalibrated camera with zero-skew, *iii*) `fguvs`, a fully uncalibrated camera. First, we assess the theoretical correctness of our proposed solvers and their resilience to noise in simulated image sequences (Sec. 4.1). Then, in Sec. 4.2, we perform experiments on real image sequences and compare the results attained by our solvers with several competing autocalibration methods. Finally, we demonstrate that integrating our solvers into the reconstruction pipeline COLMAP [43, 44] improves autocalibration and reconstruction on real image sequences (Sec. 4.3).

Competitors. We compare our solvers to the HC-based method for solving Kruppa's equations in [30]. As described in Sec. 1.3, we remark that, in this method, each subsystem of 5 / 6 Kruppa's equations may also be considered minimal relaxations in the sense of Section 3. Moreover, whether we consider these equations as *rational* or *polynomial* functions matters. In the latter case, considered in [30], it was correctly observed that these equations had the expected number of $2^5 = 32$ solutions over $\mathbb{C}$. However, for 14 of these solutions, denominators appearing in the rational form of Kruppa's equations become undefined. Thus, only 18 HC paths must be tracked to find a valid solution.

To address the imbalance between our method (based on triples of image points) and Kruppa (based on triples of fundamental matrices), we consider three variants of Kruppa that estimate these fundamental matrices differently. The first variant, *Kruppa-8*, estimates fundamental matrices using the non-minimal 8-point algorithm. The second, *Kruppa-7*, estimates fundamental matrices using the minimal 7-point algorithm. The third, *Kruppa-6*, implements a minimal solver for projective reconstruction from 6 points in 3 views [42], from which a set of compatible fundamental matrices can be determined. Kruppa-6 is the closest to our `fguvs` solver, which also requires six points. For all three variants, we normalize the input as in [22].

In real-world experiments, we also compare our solvers with the state-of-the-art camera autocalibration approach presented in [36]. This method uses semidefinite programming and a Branch-and-Bound (BnB) scheme to maximize consensus among polynomials and solve the calibration problem with either the Kruppa equations [31] or the modulus constraint [37]. We refer to these variants as *Kruppa BnB* and *Modulus BnB*, respectively.

Implementation. We implement our solvers in Julia using the package `HomotopyContinuation` [2], with C++ and Python bindings. SM 11.1 reports the minimal relaxations used by these solvers. All experiments were conducted on an Intel Core i9 13900k with 16GB RAM.

4.1. Synthetic Experiments

We evaluate the performance of our `ffuv0`, `fguv0`, and `fguvs` solvers in synthetic images under varying noise levels applied to the generated pixel coordinates. Our evaluation involves comparing the Kruppa-8 [30] and Kruppa-6 methods. Results for Kruppa-7 are inferior in accuracy and are presented in SM 11.4.

Simulations. In each simulated scene, we generate 100 randomly distributed 3D points within the unit sphere. We simulate three camera displacements, with the first located 2 world units from the sphere's center along the y-axis. The other two cameras are translated by ± 0.5 units along all axes relative to the first camera, enforcing a minimum L2-norm of 0.1 for translation vectors. Camera motion is constrained to ensure all views capture the scene, with random rotations obtained by uniformly sampling angles in the ± 45 degrees range along all axes. Simulated points are projected onto 640×480 images, discarding any points not observed in all views. Noise is introduced by adding zero-mean Gaussian displacements to pixel coordinates with standard deviation σ in the $[0, 1]$ range in increments of 0.2. For each noise level σ, we conduct 1000 independent tests for all methods. Our solvers, Kruppa-6 and Kruppa-8 [30] are evaluated with intrinsics: $f = 330$, $g = 310$, $u = 300$, $v = 250$, $s = 10$.

Metrics. We report the relative error Δfg in focal lengths and errors in the principal point Δuv and skew Δs,

$$\Delta uv = \frac{1}{2}\left(\frac{|\hat{u} - u_{\text{gt}}|}{u_{\text{gt}}} + \frac{|\hat{v} - v_{\text{gt}}|}{v_{\text{gt}}}\right), \quad \Delta s = 2\frac{|\hat{s} - s_{\text{gt}}|}{f_{\text{gt}} + g_{\text{gt}}}.$$

We report the reprojection error Re computed using the estimated intrinsics K and camera poses[4] $\{R_i, C_i\}_{i=1}^{M}$. Note that Re is not reported for Kruppa-8 due to the inconsistent reconstruction across the three views obtained from fundamental matrices computed using the 8-point algorithm. This inconsistency leads to significantly higher errors, making any comparison unfair. Instead, we report Re for Kruppa-6, where a metric reconstruction consistent across the three views is obtained by upgrading the projective cameras using the estimated intrinsic parameters.

Results. In Fig. 4, boxes represent the interquartile range of errors in estimated camera parameters and mean reprojection error. Errors generally increase with higher noise levels σ. Across most experiments, `ffuv0` and `fguv0`, despite assuming prior camera knowledge not aligned to the synthetic camera parameters, match or surpass Kruppa methods, particularly in focal length, with principal point results generally within a $\pm 5\%$ deviation from Kruppa methods. `fguvs`,

[4] SM 11.2 explains how camera poses are derived from projective depths and SM 11.3 gives the formula for the reprojection error.

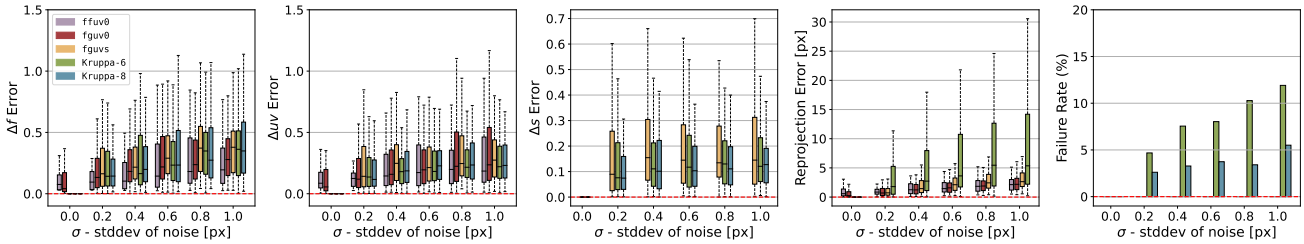

Figure 4. **Autocalibration Evaluation on Synthetic Images.** Solver accuracy is assessed under varying levels of zero-mean Gaussian noise (denoted by σ on the x-axis) applied to pixel coordinates. Mean reprojection error and relative errors in focal lengths Δfg, principal point Δuv, and skew Δs are reported. For error measures, boxes represent the interquartile range of error distribution. The right-most plot illustrates the failure rate as a percentage, with `ffuv0`, `fguv0`, and `fguvs` excluded due to no failures.

similar to Kruppa's in not assuming prior camera knowledge, attains comparable performance to Kruppa methods in focal length and principal point estimation but underperforms in skew estimation, especially for $\sigma > 0.4$. Across all noise levels, all our solvers outperform Kruppa-6 in reprojection error, which jointly assesses the accuracy in intrinsic camera parameters and camera pose estimation.

Kruppa methods recover K via the Cholesky decomposition of the DIAC ω^*. In the presence of noise, ω^* may not be positive-semidefinite. This leads to autocalibration failure, making the estimation of K unfeasible. Failure rates range from 4% to 13% for Kruppa-6 and from 3% to 6% for Kruppa-8, as reported in Fig. 4-right. In principle, our solvers could also fail at higher noise levels. However, we did not encounter these issues in our synthetic experiments.

Remark 1. In SM 11.4, we confirm the theoretical correctness of `ffuv0` and `fguv0` by showing that zero error is attained in the noiseless case when each solver's prior camera knowledge matches the synthetic camera parameters.

Remark 2. All Kruppa-based methods present a degeneracy arising from a singularity in the Kruppa equations when the optical centers of cameras lie on a sphere, and their optical axes intersect at the sphere's center [49]. As discussed in SM 11.4, we reproduce such conditions and confirm that our method is unaffected by the Kruppa degeneracy.

4.2. Evaluation on Real Datasets

We assess autocalibration accuracy on the calibrated Fountain-P11 and Herz-Jesu-P8 [47] datasets. The `ffuv0`, `fguv0`, and `fguvs` solvers are embedded in a conventional MSAC-framework [50]. At each iteration of the MSAC, we evaluate the recovered camera intrinsics and extrinsics in terms of their induced reprojection error weighted by the Huber loss. We set a limit of 200 iterations. Image points are obtained by extracting and matching SIFT [29] keypoints across image triplets.

We compare our solvers with Kruppa-8 [30], Kruppa-7, and Kruppa-6 embedded in MSAC, mirroring our solver setup. For Kruppa-8 and Kruppa-7, we compute camera poses by decomposing the pairwise essential matrices $E = K^\top F K = [t]_\times R$, where F is the fundamental matrix. Then, we compute the reprojection error pairwise, averaging it across all image pairs. Kruppa-6 yields a consistent metric reconstruction across the three views, allowing direct computation of the reprojection error by projecting the 3D points using the recovered camera matrices. Finally, our evaluation includes Kruppa BnB and Modulus BnB [36], representing state-of-the-art autocalibration methods.

Metrics. We assess calibration accuracy using Δfg, Δuv, Δs in Eq. (9). We also report reprojection errors Re, computed using estimated camera intrinsics and extrinsics, and Re_{gt}, computed using the estimated intrinsics, but ground truth camera poses. ϵ_R and ϵ_C represent the angular errors[5] in degrees for estimated camera rotations and centers, respectively. Errors are averaged across all image sequences.

Results. Tab. 2 reports the results of our evaluation. Concerning *full* camera calibration, our `fguvs` solver sets the benchmark for most calibration metrics, except for Δuv in Fountain-P11, where it is the second-best method after Kruppa BnB. `fguvs` also outperforms Kruppa-6 at camera pose estimation. Remarkably, `fguvs` excels in focal length estimation, achieving 3.6 times lower Δfg in Fountain-P11 compared to the second-best Kruppa BnB.

The solvers `ffuv0` and `fguv0` outperform `fguvs` across various metrics, with `ffuv0` emerging as the top-performing method overall. This demonstrates the advantages of integrating partial knowledge of K into our solvers, especially given that the zero-skew assumption and square pixel aspect ratio very often hold in practice.

Our solvers' runtimes depend on the number of paths tracked by HC, *i.e.*, by the solution counts in $\mathbb{C}$. We refer to Tab. 1 to optimize speed and select $\mathbf{g}$ with the lowest solution count. We report the median runtime per iteration: `fguv0` 1.78 s/iter (2313 paths), `fguvs` 2.15 s/iter (2985 paths), `ffuv0` 9.21 s/iter (16188 paths). Our solvers are

[5] See SM 11.3 for complete error definitions.

Method	Fountain-P11							Herz-Jesu-P8						
	Δfg	Δuv	Δs	Re_{gt}	Re	ϵ_R	ϵ_C	Δfg	Δuv	Δs	Re_{gt}	Re	ϵ_R	ϵ_C
Kruppa-6	0.137	0.184	0.022	19.563	2.891	7.061	5.579	0.098	0.112	0.014	14.565	1.112	2.125	1.902
Kruppa-7	0.249	0.204	0.040	28.197	-	-	-	0.122	0.114	0.040	15.252	-	-	-
Kruppa-8 [30]	0.260	0.173	0.029	28.466	-	-	-	0.140	0.115	0.022	13.606	-	-	-
Kruppa BnB [36]	0.127	0.058	0.014	9.231	-	-	-	0.078	0.096	0.018	21.023	-	-	-
Modulus BnB [36]	0.162	0.071	0.016	10.540	-	-	-	0.097	0.102	0.019	22.641	-	-	-
`ffuv0`	**0.017**	**0.029**	-	**4.435**	**0.449**	**0.623**	**0.664**	**0.017**	**0.044**	-	**8.082**	**0.672**	**0.664**	**0.656**
`fguv0`	0.028	0.050	-	8.580	0.554	0.970	1.183	0.029	0.063	-	11.128	0.680	1.295	1.540
`fguvs`	0.035	0.064	**0.008**	9.769	1.075	1.274	1.428	0.041	0.058	**0.013**	11.348	0.989	1.085	1.139

Table 2. **Autocalibration Evaluation on Real Datasets.** Mean relative errors in the focal lengths Δfg, principal point Δuv, and skew Δs are reported. Reprojection error is computed in two variations: *i*) Re_{gt}, using estimated K and ground truth camera poses, *ii*) Re, using estimated K and estimated camera poses (when applicable). ϵ_R and ϵ_C are the angular errors in estimated camera rotations and translations, respectively. Lower values indicate better performance for all metrics.

Variant	Fountain-P11				Rathaus				KITTI-Depth			
	$\Delta fg \downarrow$	$\Delta uv \downarrow$	Re	Points3D	$\Delta fg \downarrow$	$\Delta uv \downarrow$	Re	Points3D	$\Delta fg \downarrow$	$\Delta uv \downarrow$	Re	Points3D
$\text{COLMAP}_{\text{guess}}$	0.3350	0.0140	0.444	4848	0.0671	0.0812	0.624	847	0.6510	0.1360	0.810	210
$\text{COLMAP}_{\texttt{fguv0}}$	0.0058	0.0297	0.241	5356	0.0237	**0.0111**	0.450	823	**0.0720**	**0.0185**	0.409	231
$\text{COLMAP}_{\text{guess}}$ + K-BA	0.0012	0.0013	0.212	5296	0.0185	0.0607	0.435	868	0.3480	0.5072	0.547	232
$\text{COLMAP}_{\texttt{fguv0}}$ + K-BA	**0.0011**	**0.0012**	0.212	5367	**0.0165**	0.0307	0.432	823	0.0626	0.1773	0.404	236
$\text{COLMAP}_{\text{gt}}$ + K-BA	0.0013	0.0011	0.210	5368	0.0069	0.0291	0.430	794	0.0401	0.0553	0.398	237

Table 3. Comparing errors and numbers of registered points for autocalibration strategies in COLMAP (Sec. 4.3.)

multithreaded, with quasi-linear scaling in the number of CPU cores. Comparatively, the median runtime for Kruppa-6, Kruppa-7, and Kruppa-8 is 0.71 s/iter, with $6 \cdot 18 = 108$ solutions paths overall. For Kruppa, we observe that performance scaling is not linear, but we attribute this to the small number of solutions and overhead when running the Julia HC solver. Despite their faster runtimes, these methods exhibit inferior accuracy and higher failure rates, as illustrated in Fig. 4. Setting a strict threshold of 0.02 on Δfg, the `fguvs` solver takes an average of 4.21 minutes on Fountain-P11 and Herz-Jesu-P8, whereas Kruppa methods are, on average, only 27% faster. The BnB methods [36] are the fastest overall, by 62% compared to ours, yet they still provide inferior accuracy.

4.3. Autocalibration in COLMAP

We integrate our autocalibration solvers into COLMAP [43, 44] to initialize the camera intrinsics before 3D reconstruction. The evaluation is conducted on five triplets of images from the Fountain-P11 (2 sequences), Rathaus [47] (1 sequence), and KITTI-Depth [16] (2 sequences) datasets. We report results for `fguv0`. Additional details about other solvers and datasets may be found in SM 11.

We consider two strategies for initializing K: $\text{COLMAP}_{\text{guess}}$ uses the default COLMAP guess based on image size, and $\text{COLMAP}_{\texttt{fguv0}}$ employs the `fguv0` solver. These variants exclude the K from Bundle Adjustment (BA). We also evaluate results obtained using BA on K (+ K-BA). $\text{COLMAP}_{\text{gt}}$ + BA involves starting from ground truth camera parameters and applying BA and is provided as an oracle for performance.

Tab. 3 reports results for each strategy. $\text{COLMAP}_{\texttt{fguv0}}$ estimates K better than $\text{COLMAP}_{\text{guess}}$ in most cases and yields accurate reconstructions, even without refining K. When applying BA, the gap between $\text{COLMAP}_{\texttt{fguv0}}$ and $\text{COLMAP}_{\text{guess}}$ narrows, particularly in Fountain-P11, where many keypoints are available. In Rathaus, the principal point is displaced from the image center. The final calibration accuracy is improved by using the estimate of K from `fguv0`. In KITTI-Depth, BA often struggles due to fewer matches. In this scenario, using BA results in a 9.58x degradation in Δuv, but only a 1.15x improvement in Δfg compared to the calibration by `fguv0`. This indicates that in challenging scenes, our estimates of K are more reliable than those obtained solely through refinement with BA.

5. Conclusion

Motivated by the quest for a complete understanding of the autocalibration of a camera with constant K, we presented a new complete analysis of minimal autocalibration problems and their implementations, improving the state-of-the-art.

Acknowledgements: TD was supported by NSF DMS-2103310. APDC and LM were supported by FAIR (Future Artificial Intelligence Research) project funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence) and by GEOPRIDE ID: 2022245ZYB, CUP: D53D23008370001, (PRIN 2022 M4.C2.1.1 Investment). EU H2020 No. 871245 SPRING project supported TP.

References

[1] Sylvain Bougnoux. From projective to Euclidean space under any practical situation, a criticism of self-calibration. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV-98), Bombay, India, January 4-7, 1998*, pages 790–798. IEEE Computer Society, 1998. 2

[2] Paul Breiding and Sascha Timme. HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In *International Congress on Mathematical Software*, pages 458–465. Springer, 2018. 6

[3] Martin Bråtelund and Felix Rydell. Compatibility of fundamental matrices for complete viewing graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3328–3336, October 2023. 2

[4] Manmohan Krishna Chandraker, Sameer Agarwal, David J. Kriegman, and Serge J. Belongie. Globally optimal affine and metric upgrades in stratified autocalibration. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, pages 1–8. IEEE Computer Society, 2007. 3

[5] Manmohan Krishna Chandraker, Sameer Agarwal, David J. Kriegman, and Serge J. Belongie. Globally optimal algorithms for stratified autocalibration. *Int. J. Comput. Vis.*, 90(2):236–254, 2010. 3

[6] David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer, 4 ed. edition, 2015. 5

[7] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 2018. 5

[8] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomas Pajdla. PLMP - Point-line Minimal Problems in Complete Multi-view Visibility. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1675–1684. IEEE, 2019. 3, 4, 5

[9] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomas Pajdla. PL_1P - point-line minimal problems under partial visibility in three views. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVI*, volume 12371 of *Lecture Notes in Computer Science*, pages 175–192. Springer, 2020. 3

[10] Timothy Duff, Viktor Korotynskiy, Tomas Pajdla, and Margaret H. Regan. Galois/monodromy groups for decomposing minimal problems in 3D reconstruction. *SIAM Journal on Applied Algebra and Geometry*, 2022. 4

[11] Olivier D. Faugeras, Quang-Tuan Luong, and Stephen J. Maybank. Camera self-calibration: Theory and experiments. In Giulio Sandini, editor, *Computer Vision - ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19-22, 1992, Proceedings*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334. Springer, 1992. 1, 2

[12] Andrea Fusiello. Uncalibrated Euclidean reconstruction: a review. *Image and Vision Computing*, 18(6-7):555–563, 2000.

[13] Andrea Fusiello, Arrigo Benedetti, Michela Farenzena, and Alessandro Busti. Globally convergent autocalibration using interval analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(12):1633–1638, 2004. 3

[14] Guillermo Gallego, Elias Mueggler, and Peter F. Sturm. Translation of "Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung" by Erwin Kruppa (1913). *CoRR*, abs/1801.01454, 2018. 1, 2

[15] Amnon Geifman, Yoni Kasten, Meirav Galun, and Ronen Basri. Averaging essential and fundamental matrices in collinear camera settings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6020–6029. Computer Vision Foundation / IEEE, 2020. 2

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 8

[17] Riccardo Gherardi and Andrea Fusiello. Practical autocalibration. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 790–801. Springer, 2010. 3

[18] Frank Harary. *Graph theory*. Addison-Wesley Publishing Co., Reading, Mass.-Menlo Park, Calif.-London, 1969. 4

[19] R.I. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):133–135, 1997. 2

[20] R. Hartley and A. Zisserman. *Multiple view geometry in Computer Vision*. Cambridge University Press, Cambridge, second edition, 2003. With a foreword by Olivier Faugeras. 2, 3, 4

[21] Richard I Hartley. Euclidean reconstruction from uncalibrated views. In *Joint European-US workshop on applications of invariance in computer vision*, pages 235–256. Springer, 1993. 2

[22] Richard I. Hartley. In Defence of the 8-Point Algorithm. In *Procedings of the Fifth International Conference on Computer Vision (ICCV 95), Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, June 20-23, 1995*, pages 1064–1070. IEEE Computer Society, 1995. 6

[23] Anders Heyden and Kalle Åström. Flexible calibration: Minimal cases for auto-calibration. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*, pages 350–355. IEEE Computer Society, 1999. 1, 2

[24] Petr Hruby, Timothy Duff, Anton Leykin, and Tomás Pajdla. Learning to solve hard minimal problems. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5522–5532. IEEE, 2022. 1, 4

[25] Alpár Jüttner and Péter Madarasi. Vf2++—an improved subgraph isomorphism algorithm. *Discrete Applied Mathematics*, 242:69–81, 2018. 4

[26] Yoni Kasten, Amnon Geifman, Meirav Galun, and Ronen Basri. Algebraic characterization of essential matrices and their averaging in multiview settings. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019,*

Seoul, Korea (South), October 27 - November 2, 2019, pages 5894–5902. IEEE, 2019. 2

[27] Joe Kileel. Minimal problems for the calibrated trifocal variety. *SIAM Journal on Applied Algebra and Geometry*, 1(1):575–598, 2017. 3

[28] Erwin Kruppa. *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung*. Hölder, 1913. 1, 2

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 7

[30] Quang-Tuan Luong and Olivier D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *Int. J. Comput. Vis.*, 22(3):261–289, 1997. 2, 6, 7, 8

[31] Quang-Tuan Luong, Rachid Deriche, Olivier Faugeras, and Theodore Papadopoulo. *On determining the fundamental matrix: Analysis of different methods and experimental results*. PhD thesis, Inria, 1993. 6

[32] Yi Ma, René Vidal, Jana Kosecka, and Shankar Sastry. Kruppa equation revisited: Its renormalization and degeneracy. In David Vernon, editor, *Computer Vision - ECCV 2000, 6th European Conference on Computer Vision, Dublin, Ireland, June 26 - July 1, 2000, Proceedings, Part II*, volume 1843 of *Lecture Notes in Computer Science*, pages 561–577. Springer, 2000. 2

[33] Evgeniy V. Martyushev. A minimal six-point auto-calibration algorithm. *http://arxiv.org/abs/1307.3759*, 2013. 1, 2

[34] Stephen J. Maybank and Olivier D. Faugeras. A theory of self-calibration of a moving camera. *Int. J. Comput. Vis.*, 8(2):123–151, 1992. 1, 2

[35] NetworkX Developers. NetworkX, 2023. Accessed: November 24, 2023. 4

[36] Danda Pani Paudel and Luc Van Gool. Sampling algebraic varieties for robust camera autocalibration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 265–281, 2018. 1, 3, 6, 7, 8

[37] Marc Pollefeys, Luc Van Gool, and André Oosterlinck. The modulus constraint: a new constraint self-calibration. In *13th International Conference on Pattern Recognition, ICPR 1996, Vienna, Austria, 25-19 August, 1996*, pages 349–353. IEEE Computer Society, 1996. 6

[38] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *Int. J. Comput. Vis.*, 32(1):7–25, 1999. 1, 2

[39] Marc Pollefeys and Luc Van Gool. A stratified approach to metric self-calibration. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 407–412. IEEE, 1997. 2

[40] Long Quan, Bill Triggs, and Bernard Mourrain. Some results on minimal Euclidean reconstruction from four points. *Journal of Mathematical Imaging and Vision*, 24:341–348, 2006. 1

[41] Torsten Sattler, Chris Sweeney, and Marc Pollefeys. On sampling focal length values to solve the absolute pose problem. In David J. Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*, volume 8692 of *Lecture Notes in Computer Science*, pages 828–843. Springer, 2014. 2

[42] Frederik Schaffalitzky, Andrew Zisserman, Richard I Hartley, and Philip HS Torr. A six point solution for structure and motion. In *Computer Vision-ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part I 6*, pages 632–648. Springer, 2000. 6

[43] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6, 8

[44] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 6, 8

[45] Andrew J. Sommese and Charles W. Wampler, II. *The numerical solution of systems of polynomials arising in engineering and science*. World Scientific, Hackensack, NJ, 2005. 5

[46] Henrik Stewénius, David Nistér, Fredrik Kahl, and Frederik Schaffalitzky. A minimal solution for relative pose with unknown focal length. *Image and Vision Computing*, 26(7):871–877, 2008. 2

[47] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008. 1, 7, 8

[48] Peter Sturm. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1100–1105. IEEE, 1997. 1

[49] P. Sturm. A case against Kruppa's equations for camera self-calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1199–1204, 2000. 2, 7

[50] Philip HS Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156, 2000. 7

[51] Bill Triggs. Autocalibration and the absolute quadric. In *1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), June 17-19, 1997, San Juan, Puerto Rico*, pages 609–614. IEEE Computer Society, 1997. 1, 2

[52] Hassler Whitney. Congruent graphs and the connectivity of graphs. *Hassler Whitney Collected Papers*, pages 61–79, 1992. 4

[53] Cyril Zeller and Olivier Faugeras. *Camera self-calibration from video sequences: the Kruppa equations revisited*. PhD thesis, INRIA, 1996. 2