Minimal Perspective Autocalibration
Supplementary Material

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

In this document, we provide additional details concerning the main paper.

6. Problem formulation — additional details

Although the depth equations (4) described in Sec. 2.1 are the main constraints used in our approach, we wish to point out that they are by no means the only polynomial equations involving depths λ_{ip}, image points x_{ip} and the calibration matrix K that must be satisfied by an exact solution $((\lambda_{ip}), K)$. In the language of Sec. 11.1: the depth constraints determine the variety of problem-solution pairs X *locally but not globally.*

We may derive additional constraints as follows: using (1), for any view pair $1 \leq i < j \leq M$ and four distinct world points with indices $1 \leq p_1, \ldots, p_4 \leq 4$, we have

$$\det\left(\begin{array}{cccc}\lambda_{ip_1}x_{ip_1}-\lambda_{ip_2}x_{ip_2} & \lambda_{ip_3}x_{ip_3}-\lambda_{ip_2}x_{ip_2} & \lambda_{ip_4}x_{ip_4}-\lambda_{ip_2}x_{ip_2}\end{array}\right) = \det\left(\begin{array}{cccc}\lambda_{jp_1}x_{jp_1}-\lambda_{jp_2}x_{jp_2} & \lambda_{jp_3}x_{jp_3}-\lambda_{jp_2}x_{jp_2} & \lambda_{jp_4}x_{jp_4}-\lambda_{jp_2}x_{jp_2}\end{array}\right). \tag{9}$$

This follows from our assumption that K, and hence also $\det(K)$, is constant: compare with (19) below.

It is important to remember that, when solving with a minimal relaxation, the equations that are not enforced may or may not continue to hold for noisy data. As an example of this, we may consider the unique class of minimal problems in Table 1 for the scenario `11000` with $M = 3$ fully calibrated views. As illustrated in Fig. 5, we may drop exactly one depth equation for the view pair $(i, j) = (1, 2)$ to obtain a representative for the equivalence class of minimal relaxations. This relaxation has the effect that (9) no longer must hold for this view pair. Indeed, we find that this equation is typically violated in the case of noisy data and for all 639 non-synthetic solutions when solving a generic synthetic problem instance. On the other hand, for the view pair $(i, j) = (1, 3)$, the equation (9) holds even for non-synthetic solutions or noisy data.

We may rephrase the observations of the previous paragraph in the geometric language developed in Sec. 3 (see also SM 9 below.) From this point of view, (9) is valid for both view pairs on the incidence variety X associated with the overconstrained problem but only generally valid for the view pair $(1, 3)$ on the incidence variety $\mathbb{V}(\mathbf{g})$ associated to the minimal relaxation. The local nature of parameter homotopy ensures, for this problem, that we do not need to explicitly enforce the constraint (9) for one view pair. However, any attempt to simultaneously enforce these constraints for both view pairs and the chosen depth constraints will invariably lead us back to an overconstrained problem.

7. Normalization of known intrinsics

Referring to Sec. 2.2, we provide details on transforming image coordinates to normalize the value of known intrinsic parameters. Without loss of generality, for $R_i = I_3$ and $C_i = 0$, Eq. (1) writes

$$\lambda_{ip}\, x_{ip} = \begin{pmatrix} K \mid 0 \end{pmatrix} \begin{pmatrix} X_p \\ 1 \end{pmatrix}, \; i \in [M], \; p \in [N]\,. \tag{10}$$

If $X_p = (\alpha \;\; \beta \;\; \gamma)^\top$, then $\lambda_{ip} = \gamma$, and

$$x_{ip} = \begin{pmatrix} x_{ip,1} \\ x_{ip,2} \\ 1 \end{pmatrix} = \begin{pmatrix} fa + sb + u \\ gb + v \\ 1 \end{pmatrix}, \; a = \frac{\alpha}{\gamma}, \; b = \frac{\beta}{\gamma}\,. \tag{11}$$

Additionally, we present the well-known decomposition of K into scaling, shear, and translation transformations ap-

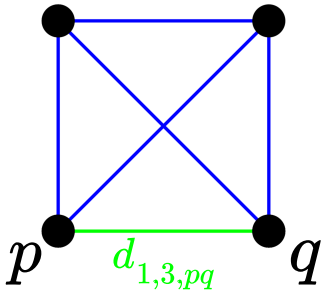


Figure 5. Visualization of a representative of the equivalence class of minimal relaxations for the `11000` problem obtained by dropping a constraint $d_{1,2,pq}$ for view pair $(1, 2)$.

Prior on K	M	N	L	Min # sol. in $\mathbb{C}$	Max # sol. in $\mathbb{C}$	#balanced	#**g**
fguvs	2	-	0	∞	∞	0	0
fguvs	3	6	0	11940^*	4544808^*	5852925	3313
fguv0	2	-	1	∞	∞	0	0
fguv0	3	5	1	9252	9252	190	3
fgu0s	2	-	1	∞	∞	0	0
fgu0s	3	5	1	8232	8232	190	3
fgu00	2	-	2	∞	∞	0	0
fgu00	3	5	2	38744	134424	1140	8
fg0vs	2	-	1	∞	∞	0	0
fg0vs	3	5	1	8232	8232	190	3
fg0v0	2	-	2	∞	∞	0	0
fg0v0	3	5	2	38744	450080	1140	8
fg00s	2	-	2	∞	∞	0	0
fg00s	3	5	2	38744	134424	1140	8
fg000	2	7	3	72	72	1	1
fg000	3	5	3	15536	830656	4845	37
f1uvs	2	-	2	∞	∞	0	0
f1uvs	3	5	2	8222	8222	190	3
f1uv0	2	-	2	∞	∞	0	0
f1uv0	3	5	2	58088	201632	1140	8
f1u0s	2	-	2	∞	∞	1	1
f1u0s	3	5	2	29520	320380	1140	8
f1u00	2	7	3	36	36	1	1
f1u00	3	5	3	8800	489088	4845	37
f10vs	2	-	2	∞	∞	0	0
f10vs	3	5	2	48664	173078	1140	8
f10v0	2	7	3	72	72	1	1
f10v0	3	5	3	15528	114440	4845	37
f100s	2	7	3	36	36	1	1
f100s	3	5	3	8784	205556	4845	37
f1000	2	6	4	60	60	1	1
f1000	3	4	4	1336	1336	1	1
1guvs	2	-	1	∞	∞	0	0
1guvs	3	5	1	8720	8720	190	3
1guv0	2	-	2	∞	∞	0	0
1guv0	3	5	2	58092	201616	1140	8
1gu0s	2	-	2	∞	∞	0	0
1gu0s	3	5	2	58048	201436	1140	8
1gu00	2	7	3	72	72	1	1
1gu00	3	5	3	15520	86630	4845	37
1g0vs	2	-	2	∞	∞	0	0
1g0vs	3	5	2	58024	631306	1140	8
1g0v0	2	7	3	36	36	1	1
1g0v0	3	5	3	8800	205568	4845	37
1g00s	2	7	3	48	48	1	1
1g00s	3	5	3	8960	477088	4845	37
1g000	2	6	4	60	60	1	1
1g000	3	4	4	1336	1336	1	1
11uvs	2	-	2	∞	∞	0	0
11uvs	3	5	2	57912	201265	1140	8
11uv0	2	7	3	48	48	1	1
11uv0	3	5	3	8940	477080	4845	37
11u0s	2	7	3	36	36	1	1
11u0s	3	5	3	8786	46192	4845	37
11u00	2	6	4	60	60	1	1
11u00	3	4	4	1336	1336	1	1
110vs	2	7	3	72	72	1	1
110vs	3	5	3	16390	85480	4845	37
110v0	2	6	4	60	60	1	1
110v0	3	4	4	1336	1336	1	1
1100s	2	6	4	60	60	1	1
1100s	3	4	4	1336	1336	1	1
11000	2	5	5	20	20	1	1
11000	3	4	5	640	640	1	1
ffuvs	2	-	1	∞	∞	0	0
ffuvs	3	5	1	9234	9234	190	3
ffuv0	2	-	2	∞	∞	0	0
ffuv0	3	5	2	32376	238238	1140	8
ffu0s	2	-	2	∞	∞	0	0
ffu0s	3	5	2	58056	201516	1140	8
ffu00	2	7	3	48	48	1	1
ffu00	3	5	3	8968	353984	4845	37
ff0vs	2	-	2	∞	∞	0	0
ff0vs	3	5	2	77400	268704	1140	8
ff0v0	2	7	3	48	48	1	1
ff0v0	3	5	3	8968	184672	4845	37
ff00s	2	7	3	72	72	1	1
ff00s	3	5	3	15512	792084	4845	37
ff000	2	6	4	60	60	1	1
ff000	3	4	4	1336	1336	1	1

Table 4. **80 classes of minimal autocalibration problems** for $M \in \{2, 3\}$ views. Solution counts refer to unknown depths and the unknown enries of K in (18). Other notation and conventoions are the same as in Table 1.

plied to normalized image coordinates, which we will reference in later sections:

$$K = \underbrace{\begin{pmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Translation}} \underbrace{\begin{pmatrix} 1 & s/g & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Shear}} \underbrace{\begin{pmatrix} f & 0 & 0 \\ 0 & g & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Scaling}} . \quad (12)$$

7.1. Known Focal Lengths

Normalization of known focal lengths involves reversing the scaling transformation in (12) along the x and/or y axes. When f is known, x_{ip} may be transformed into normalized coordinates $\tilde{x}_{ip}$ in which $f = 1$,

$$\tilde{x}_{ip} = \begin{pmatrix} a + \tilde{s}b + \tilde{u} \\ gb + v \\ 1 \end{pmatrix}, \; \tilde{s} = \frac{s}{f}, \; \tilde{u} = \frac{u}{f} . \quad (13)$$

Then, we may solve for the unknown transformed intrinsics $(g, \tilde{u}, v, \tilde{s})$ and recover the original values of u, s using the known value of f. Similarly, when g is known, we use

$$\tilde{x}_{ip} = \begin{pmatrix} fa + sb + u \\ b + \tilde{v} \\ 1 \end{pmatrix}, \; \tilde{v} = \frac{v}{g} , \quad (14)$$

we solve for the unknown transformed intrinsics $(f, u, \tilde{v}, s)$, and recover the original value of v.

7.2. Known Principal Point

Normalizing known principal point coordinates involves reversing the translation transformation in (12) to center image coordinates at the origin.

When u is known, normalizing the known value of u to $u = 0$ involves subtracting u from $x_{ip,1}$, the first coordinate of x_{ip}. Notably, no additional substitution of other intrinsic parameters is necessary. Similarly, when v is known, normalizing the known value of v to $v = 0$ may be achieved by subtracting v from the second coordinate $x_{ip,2}$.

7.3. Known Skew

Knowing the camera skew, when it is nonzero, implies knowledge of the shear transformation embedded in K, and that is applied to the normalized image coordinates.

The shear transformation in (12) is determined by $s^\star = \frac{s}{g}$. Thus, when $s^\star$ is known, the skew-induced shear in image coordinates may be removed by transforming $x_{ip,1}$, the first coordinate of x_{ip}, as follows:

$$\tilde{x}_{ip,1} = x_{ip,1} - s^\star x_{ip,2} . \quad (15)$$

Importantly, (15) successfully reverses the shear transformation when either $v = 0$ or when v is known and x_{ip} is first normalized to fix $v = 0$, as detailed in Sec. 7.2. This can be observed by rewriting x_{ip} from (11) as follows:

$$\tilde{x}_{ip} = \begin{pmatrix} \tilde{x}_{ip,1} \\ \tilde{x}_{ip,2} \\ 1 \end{pmatrix} = \begin{pmatrix} fa + s^\star x_{ip,2} + u - s^\star v \\ gb + v \\ 1 \end{pmatrix}, \quad (16)$$

by substituting $b = (x_{ip,2} - v)/g$. Notably, no additional substitution of other intrinsic parameters is necessary.

Our previous autocalibration specification can be extended to minimal problems in the notable case where a *generic* $s^\star$ is known and nonzero, but v is unknown. Referring to 2.1, $s^\star$ appears explicitly in our parametrization of ω, as defined in (2). Thus, referring to Sec. 3.3, given any system $\mathbf{g}(\mathbf{p}, \mathbf{x}) = 0$ encoding a minimal relaxation of an autocalibration problem in which $s^\star$ is known and nonzero and v is unknown, we may treat $s^\star \in \mathbf{p}$, as a parameter of the system. Then, we may construct minimal solvers using a standard online/offline parameter homotopy approach such as described in Section 3.3. Referring to Sec. 2.2, we indicate known nonzero $s^\star$ in the 5-tuple of unknowns `fguvs` by setting $s = s^\star$. This notation is used in Tab. 5 to report the solution count in $\mathbb{C}$ computed during the offline stage for all cases where $s^\star$ is known and nonzero and v is unknown, mirroring the comprehensive approach taken in Tab. 1.

8. Depth equations without symmetry removal

We discuss the effect of substituting (2) into $d_{i,j,pq}$ on the solution count in $\mathbb{C}$ for the 80 interesting minimal autocalibration problems presented in Tab. 1.

Without substituting (2), $d_{i,j,pq}(\lambda, K; x)$ writes:

$$\begin{aligned} d_{i,j,pq}(\lambda, K; x) :=& (K^{-1}\lambda_{ip}x_{ip} - K^{-1}\lambda_{iq}x_{iq})^T \\ & (K^{-1}\lambda_{ip}x_{ip} - K^{-1}\lambda_{iq}x_{iq}) \\ & -(K^{-1}\lambda_{jp}x_{jp} - K^{-1}\lambda_{jq}x_{jq})^T \\ & (K^{-1}\lambda_{jp}x_{jp} - K^{-1}\lambda_{jq}x_{jq}), \end{aligned} \quad (17)$$

and we may write our main constraint as

$$d_{i,j,pq}(\lambda, K; x) = 0\,. \quad (18)$$

In Tab. 4, we list the same 80 interesting groups of problems indexed by $(\texttt{fguvs}, M, N)$, mirroring Tab. 1 in the main paper. Notably, we report the minimum and maximum solution count in $\mathbb{C}$, referring to the unknown depths and the parameters of K in (1)-(17).

9. Details on minimal relaxations

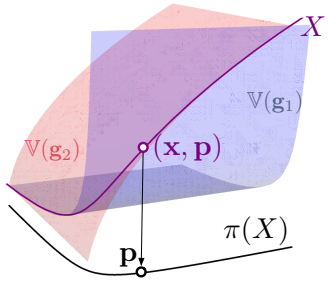
To illustrate some technicalities in our definition of minimal relaxation, we consider again the irreducible variety

$$X = \{(p_1, p_2, x) \in \mathbb{C}^3 \mid x^2 - p_1 = p_2 x^2 - 1 = 0\}$$

Prior on K	M	N	L	Min # sol. in $\mathbb{C}$
`fguvs`*	2	-	1	∞
`fguvs`*	3	5	1	2313
`fg0vs`*	2	-	2	∞
`fg0vs`*	3	5	2	19365
`f1uvs`*	2	-	2	∞
`f1uvs`*	3	5	2	29044
`f10vs`*	2	7	3	36
`f10vs`*	3	5	3	8272
`1guvs`*	2	-	2	∞
`1guvs`*	3	5	2	29046
`1g0vs`*	2	7	3	36
`1g0vs`*	3	5	3	8282
`11uvs`*	2	-	3	48
`11uvs`*	3	5	3	8940
`110vs`*	2	6	4	60
`110vs`*	3	4	4	1336
`ffuvs`*	2	-	2	∞
`ffuvs`*	3	5	2	16188
`ff0vs`*	2	7	3	24
`ff0vs`*	3	5	3	4482

Table 5. **Non-zero Skew Autocalibration Problems.** Specification of 20 notable minimal problems in 2 and 3 views (M) where $s^\star = \frac{s}{g}$ is known and non-zero and v is unknown. For each triplet $(\texttt{fguvs}, M, N)$, we report L, number of linear constraints on K, and the minimum, taken over all minimal relaxations, solution count in $\mathbb{C}$ for generic input (including s^*.) As in Table 1, counts refer to unknown depths and the parameters of ω in (2)–(4).

depicted in Figure 2 of the main paper. The real-valued points of X in $\mathbb{R}^3$ form the purple space curve where the red and blue surfaces intersect below.



In this example, we understand the vector $(p_1, p_2) \in \mathbb{C}^2$ to represent a problem instance and $x \in \mathbb{C}$ one of its solutions. The set of exactly solvable problems is the image of the projection map $\pi : X \to \mathbb{C}^2$, namely the hyperbola $p_1 p_2 = 1$ drawn in black. Then, the problem is overconstrained since a generic problem instance will not lie on the hyperbola and will have no solutions. This manifests in the failure of the rank conditions (7) : for a generic problem-solution pair $(p_1, p_2, x) \in X$, we have

$$\operatorname{rank} \begin{pmatrix} -1 & 0 & 2x \\ 0 & x^2 & -2p_2 x \end{pmatrix} \neq 1 \quad \forall\, (p_1, p_2, x) \in X \backslash \{(0,0,0)\}.$$

Two minimal relaxations can be obtained by dropping one of the two equations defining X. These relaxations correspond to the surfaces $X_1 = X_{x^2-p_1}$ and $X_2 = X_{p_2x^2-1}$, drawn above in red and blue, respectively. The union of these two surfaces is *not* a minimal relaxation, since the Jacobian of $(x^2 - p_1)(p_2x^2 - 1)$ vanishes identically along X. For the rational function $\mathbf{g}(p_1, p_2, x) = 1/x^2 - p_2$, note that we also have $\mathbb{V}(\mathbf{g}) = X_{\mathbf{g}} = X_2$. If we instead consider $\mathbf{g}' = (p_2x^2 - 1)x$, the variety $\mathbb{V}(\mathbf{g}')$ has two irreducible components, given by $\mathbb{V}(x)$ and the minimal relaxation $X_{\mathbf{g}'} = X_{\mathbf{g}} = X_1$. Finally, let us observe that in this example, the degrees of the minimal relaxations $X_i \to \mathbb{C}^2$ are both 2. This need not be the case in general: if we consider instead of X the space curve $\mathbb{V}(x^2 - p_1, x^3 - p_3)$, we see there are minimal relaxations of degree 2 or 3. This example also shows that relaxations can increase the number of solutions, even for an exactly solvable problem instance.

We wish to point out that our notion of a minimal relaxation occurs implicitly in previous works [9, 27] studying constraints involving calibrated trifocal tensors and point-line minimal problems with partial visibility. Both works consider a minimal relaxation of the overconstrained problem of estimating four points in three calibrated views. In this minimal relaxation, one point in one view is replaced by a line. Both of these works formulate the overconstrained problem of estimating four points in three calibrated views and consider the "Scranton" relaxation of this problem in which only a single point-point-line constraint on the trifocal tensor is enforced for one of the point triplets. This problem has 272 solutions. As observed in [24], Scranton can also be formulated in terms of depths and an extra slack variable. The depth-formulated Scranton is *not* a minimal relaxation in the sense defined above. In this work, instead of adding variables, we drop equations. We may simply drop the equation $d_{1,2,12}(\lambda, \omega; x) = 0$ in the fully calibrated case. This gives a minimal relaxation with $640 = 2 \times 320$ solutions and a two-way symmetry that sends $\lambda_{2,p} \to -\lambda_{2,p}$ and fixes all other variables. We remark that a further systematic study of symmetries appearing in our zoo of autocalibration problems, along the lines conducted in [10], would be very interesting. However, this study lies beyond the scope of this investigation.

HC methods for solving Kruppa's equations may also be understood in our framework of minimal relaxations. For Kruppa, solutions $\mathbf{x}$ are the entries of 3×3 matrix representing the DIAC, and triples of fundamental matrices specify parameters $\mathbf{p}$. Moreover, for a synthetic problem-solution pair $(\mathbf{p}_0, \mathbf{x}_0) \in X$ used to initialize monodromy, certain compatability conditions on fundamental matrices encoded in $\mathbf{p}_0$ must be satisfied [20, §15.4].

10. Enumerating minimal problems — additional details

Returning to the enumeration problem described in in Section 3, we now discuss *line graphs*, a standard graph-theoretic construction. The utility of this construction is that the isomorphism class of a 4-coloring is completely determined by an associated line graph $\mathcal{L}(c)$ after equipping it with a suitable vertex labeling. The vertices of the line graph $\mathcal{L}(c)$ are simply the non-white edges, and an edge between two vertices of $\mathcal{L}(c)$ exists whenever the two non-white edges share a vertex between 1 and N. We label each vertex $pq \in \mathcal{L}(c)$ by its color $c(pq)$. Two isomorphic graphs have isomorphic line graphs; conversely, a classical theorem of Whitney [52] implies that two connected graphs whose line graphs are isomorphic are themselves isomorphic, with the sole exception of the complete graph K_3 and the claw graph $K_{1,3}$ (see eg. [18, Theorem 8.3].) Although $\mathcal{L}(K_3) \cong \mathcal{L}(K_{1,3})$, the original graphs K_3 and $K_{1,3}$ have different numbers of edges. From this, it easily follows that we can decide whether or not two 4-colorings c_1, c_2 are isomorphic: form the two graphs $\mathcal{L}(c_1)$ and $\mathcal{L}(c_2)$, decide if there exists an isomorphism that respects their labelings, and repeat this same procedure for the two graphs $\mathcal{L}(c_1), \mathcal{L}(c_2 \circ \tau)$, where τ swaps green and red edges.

We implement software based on the *NetworkX* library [35], and the VF2++ algorithm [25] to compute the isomorphism classes for all minimal problems listed in Tab. 1. The code is implemented in Python and is publicly available at github.com/andreadalcin/MinimalPerspectiveAutocalibration.

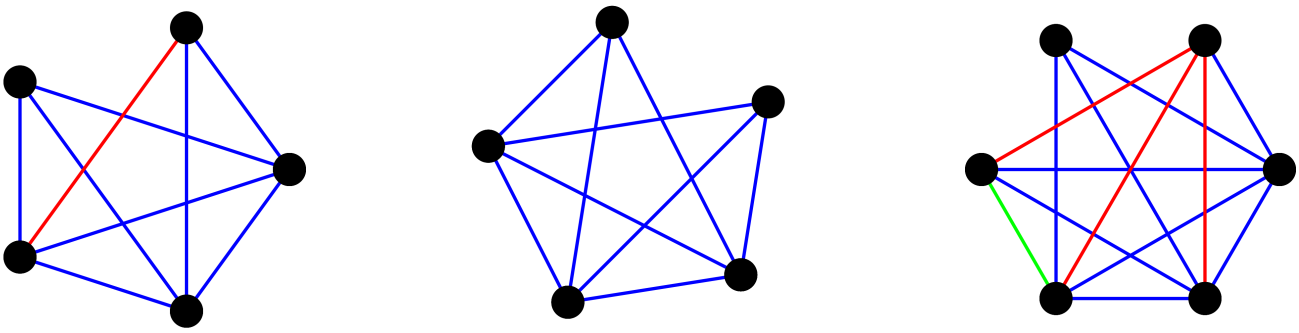
Tab. 6 provides a visualization of one representative 4-coloring for each isomorphism class for the `fguv0`, `fgu00`, `fg000` minimal problems in $M = 3$ views. The isomorphism classes for `fguvs` are too many to be visualized in Tab. 6. Still, as part of our software, we provide a visualization tool and instruction to visualize equivalence classes for `fguvs`.

11. Experiments

We provide additional details on our experimental validation that were omitted from the main paper due to space restrictions.

11.1. Choice of Minimal Relaxations

In case of different minimal relaxations yielding the same minimum solution count, e.g., in `fguv0`, we selected that yielding the lowest mean reprojection error in our synthetic test described in Sec. 4.1. We noted that reprojection error fluctuations among different relaxations with identical solution counts are always below 1%, suggesting comparable numerical performance between different minimal relaxations with the same solution count.



(a) `ffuv0` (b) `fguv0` (c) `fguvs`

Figure 6. 4-colorings of the minimal relaxations used in our implementation of solvers for the `ffuv0`, `fguv0` and `fguvs` problems.

Prior on K	M	N	L	Sol. in $\mathbb{C}$ for isomorphism class	#g
`fguv0`	3	5	1	2313 2313 2313	3
`fgu00`	3	5	2	9686 9686 9686 9686 9686 33606 33606 33606	8
`fg000`	3	5	3	17624 17624 17624 3884 3884 3884 3884 19696 19696 19696 19696 19696 19696 21696 21696 21696 21696 21696 16356 28640 16948 29888 74056 76832 74480 31183 87328 17364 77816 31964 33262 88000 33884 171696 198032 193648 207664	37

Table 6. **Visualization of Isomorphism Classes.** Visualization of one representative 4-coloring for each isomorphism class of minimal relaxations for autocalibration problems `fguv0`, `fgu00`, `fg000` in $M = 3$ views. For each triplet (`fguvs`, M, N), we report one representative 4-coloring for each isomorphism class and its associated solution count in $\mathbb{C}$. Solution counts refer to unknown depths and the parameters of ω in (2)–(4).

For the purposes of this work, we list the specific minimal relaxations used to implement the solvers `ffuv0`, `fguv0`, and `fguvs`, which are used throughout our experiments. Specifically, for each implemented solver, we specify the depth constraints that are omitted from the chosen system of equations $\mathbf{g}$:

- `ffuv0`: $d_{1,2,45}$, $d_{1,3,45}$, $d_{1,2,35}$ (see Fig. 6a).
- `fguv0`: $d_{1,2,45}$, $d_{1,3,45}$ (see Fig. 6b).
- `fguvs`: $d_{1,2,56}$, $d_{1,3,56}$, $d_{1,2,45}$, $d_{1,3,45}$, $d_{1,2,46}$, $d_{1,2,36}$, $d_{1,2,26}$, $d_{1,3,34}$ (see Fig. 6c).

Our public code also includes starting the parameters and solutions for parameter homotopies used by each solver to

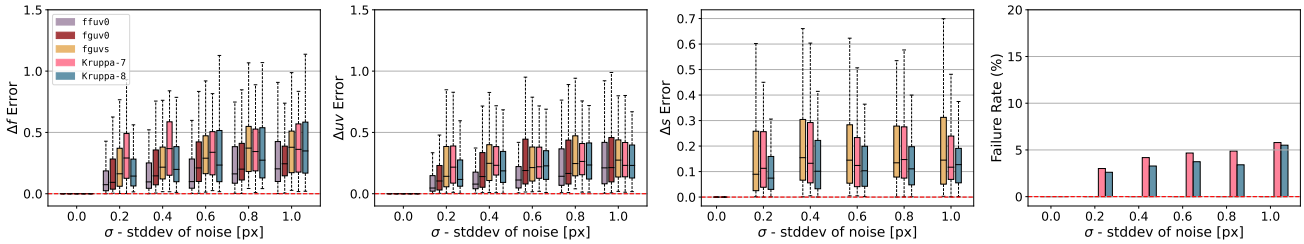


Figure 7. **Kruppa-7 Evaluation on Synthetic Images.** Solver accuracy is assessed under varying levels of zero-mean Gaussian noise (denoted by σ on the x-axis) applied to pixel coordinates. Mean reprojection error and relative errors in focal lengths Δfg, principal point Δuv, and skew Δs are reported. For error measures, boxes represent the interquartile range of error distribution. The right-most plot illustrates the failure rate as a percentage, with `fguv0`, `ffuvs`, and `fguvs` excluded due to no failures. Results are averaged for 1000 synthetic image sequences. Synthetic camera parameters are set to mirror the configuration described in Sec. 4.1. Results are averaged over 1000 synthetic image sequences.

ensure full reproducibility.

11.2. Computing Camera Rotations and Centers from Projective Depths

We discuss the conversion of projective depths into camera rotations and centers. Referring to Sec. 4, our autocalibration formulation allows us to perform an Euclidean reconstruction. However, we obtain the projective depths associated with image points rather than camera roto-translations and 3D point coordinates. However, recovering camera rotations and centers is straightforward, assuming that, without loss of generality, the $i = 1$ camera is at $R_1 = I$ and $C_1 = 0$.

Initially, we compute 3D points, denoted as X_{ip}, for each camera $i \in [M]$, using $X_{ip} = \lambda_{ip} K^{-1} x_{ip}$. Throughout this section, we express X_{ip} in Cartesian coordinates, *i.e.*, $X_{ip} \in \mathbb{R}^3$. Centering all 3D points by subtracting X_{11} (the point $p = 1$ seen by the $i = 1$ camera), we extract translation components $t_i = X_{i1}$ for $i \in [2, \dots, M]$. Subtracting the translation component from the 3D points seen by the i-th view ($\tilde{X}_{ip} = X_{ip} - t_i$), we compute the rotation component

$$R_i = \begin{pmatrix} \tilde{X}_{i2} & \tilde{X}_{i3} & \tilde{X}_{i4} \end{pmatrix} \begin{pmatrix} \tilde{X}_{12} & \tilde{X}_{13} & \tilde{X}_{14} \end{pmatrix}^{-1} . \tag{19}$$

Finally, we compute the camera centers

$$C_i = \lambda_{i2} K^{-1} x_{i2} - R_i \lambda_{12} K^{-1} x_{12} , \quad i \in [2, \dots, M] . \tag{20}$$

As discussed in Sec. 6, a given relaxation might remove constraints that enforce the validity of the recovered rotation matrices, *i.e.*, $R_i \in \mathrm{SO}(3)$. If R_i is not an orthogonal matrix, we compute an SVD $R_i = USV^\top$ and the closest orthogonal matrix $\tilde{R}_i = UV^\top$ (with respect to either the Frobenius or spectral norm.)

This approach is consistently applied in our experimental validation, and we provide Julia code for the conversion from projective depth to camera roto-translation.

11.3. Error Metrics

We provide complete error definitions for the error metrics employed in our experimental evaluation within the main paper. Referring to Sec. 4, we define the relative error Δfg in focal lengths as follows:

$$\Delta fg = \frac{1}{2} \left(\frac{|\hat{f} - f_{\mathrm{gt}}|}{f_{\mathrm{gt}}} + \frac{|\hat{g} - g_{\mathrm{gt}}|}{g_{\mathrm{gt}}} \right) . \tag{21}$$

The reprojection error Re is computed as

$$Re = \frac{1}{N(M-1)} \sum_{i=2}^{M} \sum_{p=1}^{N} \| x_{ip} - \hat{x}_{ip} \|_2 , \tag{22}$$

where $\hat{x}_{ip}$ denotes the projection of the p-th point onto the i-th view using the estimated camera parameters:

$$\hat{x}_{ip} = \frac{1}{\hat{\lambda}_{ip}} \hat{K} \, \hat{R}_i \left(I \mid -\hat{C}_i \right) \begin{pmatrix} \hat{\lambda}_{1p} \hat{K}^{-1} x_{1p} \\ 1 \end{pmatrix} , \tag{23}$$

where $\hat{K}, \hat{R}_i, \hat{C}_i, \hat{\lambda}_{ip}$ denote the estimated camera intrinsics, rotation, center, and projective depth of point p seen by view i, respectively. The definition of the reprojection error Re_{gt} is consistent with (22), but $\hat{x}_{ip}$ is computed as:

$$\hat{x}_{ip} = \frac{1}{\hat{\lambda}_{ip}} \hat{K} \, R_i \left(I \mid -C_i \right) \begin{pmatrix} \hat{\lambda}_{1p} \hat{K}^{-1} x_{1p} \\ 1 \end{pmatrix} , \tag{24}$$

where R_i, C_i are the ground truth camera rotation and center, respectively.

In Sec. 4.2, the angular errors ϵ_R and ϵ_C for estimated camera rotations and centers, respectively, are defined as

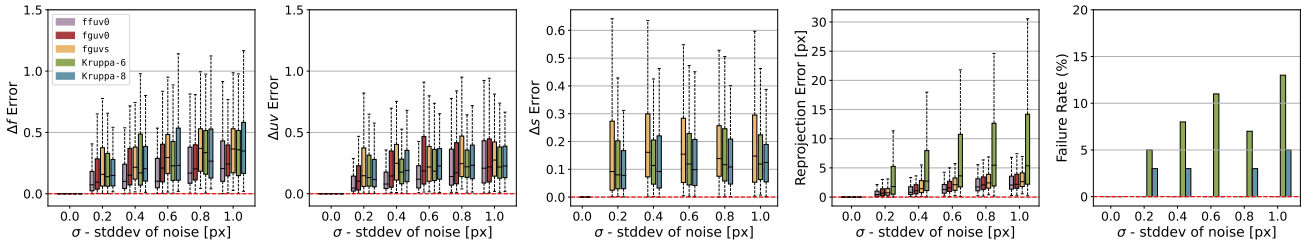


Figure 8. **Autocalibration Evaluation on Synthetic Images with Specialized Intrinsics.** Solver accuracy is assessed under varying levels of zero-mean Gaussian noise (denoted by σ on the x-axis) applied to pixel coordinates. Mean reprojection error and relative errors in focal lengths Δfg, principal 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. Synthetic camera parameters vary across solvers to match their prior camera knowledge. For `fguv0` we set $s = 0$, and for `ffuv0` we set $f = g = 330$. Results are averaged over 1000 synthetic image sequences.

	Fountain-P11				Rathaus				KITTI-Depth			
Variant	$\Delta fg \downarrow$	$\Delta uv \downarrow$	Re	Points3D	$\Delta fg \downarrow$	$\Delta uv \downarrow$	Re	Points3D	$\Delta fg \downarrow$	$\Delta uv \downarrow$	Re	Points3D
COLMAP$_{\text{guess}}$	0.3350	0.0140	0.444	4848	0.0671	0.0812	0.624	847	0.6510	0.1360	0.810	210
COLMAP$_{\texttt{ffuv0}}$	0.0053	0.0281	0.238	5357	0.0215	0.0102	0.441	831	0.0698	0.0161	0.402	231
COLMAP$_{\texttt{fguv0}}$	0.0058	0.0297	0.241	5356	0.0237	0.0111	0.450	823	0.0720	0.0185	0.409	231
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
COLMAP$_{\texttt{ffuv0}}$ + K-BA	0.0011	0.0012	0.212	5367	0.0165	0.0305	0.431	823	0.0691	0.1281	0.401	236
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
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 7. Comparing errors and numbers of registered points for autocalibration strategies in COLMAP.

follows:

$$\epsilon_R = \frac{1}{M-1}\sum_{i=2}^{M}\left|\arccos\left(\frac{\text{tr}(R_i^\top \hat{R}_i)-1}{2}\right)\right|, \quad (25)$$

$$\epsilon_C = \frac{1}{M-1}\sum_{i=2}^{M}\left|\arccos\left(\frac{C_i^\top \hat{C}_i}{\|C_i\|\|\hat{C}_i\|}\right)\right|, \quad (26)$$

where $\hat{R}_i, \hat{C}_i$ denote the estimated rotation and camera center, respectively, of the i-th camera with respect to the $i = 1$ camera, for which $R_1 = I$ and $C_1 = [0, 0, 0]$. R_i, C_i denote the ground truth camera rotation and center, respectively. The values of ϵ_R and ϵ_C are expressed in degrees for all experiments.

11.4. Synthetic Experiments — additional details

We provide additional details regarding our synthetic evaluation, referring to Sec. 4.1 of the main paper.

Kruppa-7. Fig. 7 presents a comparative analysis of the results achieved by the Kruppa-7 solver in relation to the `ffuv0`, `fguv0`, `fguvs`, and Kruppa-8 solvers. Results reveal that Kruppa-7 exhibits inferior accuracy in focal length estimation (Δfg) compared to Kruppa-8, especially at lower noise levels $\sigma \geq 0.6$. Nonetheless, principal point and skew estimation accuracy are comparable to Kruppa-8. Finally, the failure rates of Kruppa-7 match or surpass those of Kruppa-8 across all noise levels σ.

Evaluation with Specialized Intrinsics. Fig. 8 presents the results of our `ffuv0` and `fguv0` solvers evaluated on synthetic scenes generated using varying camera parameters that depend on the prior camera knowledge assumed by each solver. For `fguvs`, Kruppa-6, and Kruppa-8, which do not assume any camera knowledge, synthetic camera parameters are set to $f = 330$, $g = 310$, $u = 300$, $v = 250$, and $s = 10$. For `fguv0`, which assumes zero-skew, we set $s = 0$. Finally, for `ffuv0`, which assumes squared pixel aspect ratio, we set $f = g = 330$.

The results affirm that `ffuv0` and `fguv0` excel in focal length estimation and achieve comparable performance to the other solvers in principal point estimation. As expected, errors Δfg and Δuv are reduced due to the synthetic camera parameters perfectly matching the prior knowledge assumed by each of these solvers.

This evaluation also focuses on evaluating the theoretical correctness of our solvers, *i.e.*, verifying that zero error is achieved in the noiseless case. Both `ffuv0` and `fguv0` at-

tain zero errors when $\delta = 0$ and their assumed prior knowledge matches the ground truth camera parameters, confirming their theoretical correctness. As expected, our solvers do not exhibit any failures in this synthetic evaluation.

Degeneracies. As discussed in Sec. 4.1, our proposed autocalibration solvers are unaffected by the degeneracy of Kruppa-based methods 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]. To verify that our solvers overcome this substantial problem, we synthetically reproduce the aforementioned degeneracy condition and assess calibration on these generated image sequences. We generate 1000 synthetic scenes that exhibit the degeneracy condition of Kruppa and verify our `ffuv0`, `fguv0` and `fguvs` can successfully perform autocalibration in all cases, exhibiting zero-error in the noiseless case and a failure rate of 0%.

Furthermore, referring to Sec. 3.3, we confirm the well-posed nature of our autocalibration problem by verifying that the Jacobian of the given relaxed system $\mathbf{g}(\mathbf{p}, \mathbf{x}) = 0$ is full-rank at the synthetic solution $(\mathbf{p}_0, \mathbf{x}_0)$. Additionally, we observe that the least singular value of the Jacobian of the system never falls below 10^{-4} in our testing. This indicates the robust numerical stability of our solvers.

We implement code to generate degenerate image sequences for Kruppa equations and verify that our solvers are unaffected by this problem. The code is implemented in Julia and is publicly available at github.com/andreadalcin/MinimalPerspectiveAutocalibration.

11.5. Autocalibration in COLMAP — additional details

Finally, we provide further details on evaluating the COLMAP integration of our autocalibration solvers.

Datasets. For Fountain-P11 [47], we consider the 0-2-4 and 0-3-6 image triplets. Rathaus [47] includes a single calibrated image triplet. For KITTI-Depth [16], we extract image triplets from the 2011-09-26-drive-0005 sequence.

Results. Tab. 7 provides an extended view of Tab. 3, including the results achieved by $\text{COLMAP}_{\texttt{ffuv0}}$ — the initialization strategy of K based on the `ffuv0` solver. This variant excludes K from Bundle Adjustment (BA), and we present results achieved by $\text{COLMAP}_{\texttt{ffuv0}}$ when BA is applied to K. The calibration accuracy of $\text{COLMAP}_{\texttt{ffuv0}}$ surpasses that of $\text{COLMAP}_{\texttt{fguv0}}$ marginally, particularly in terms of principal point estimation (Δuv). When Bundle Adjustment is extended to K, the disparity in performance diminishes and becomes negligible, especially in the Fountain-P11 and Rathaus datasets. Notably, in KITTI-Depth, extending Bundle Adjustment to K shows that $\text{COLMAP}_{\texttt{ffuv0}}$ achieves a slightly lower accuracy in focal lengths (Δfg) but exhibits improved accuracy in the principal point (Δuv).

Discussion. In the context of integrating our autocalibration solvers into COLMAP, both `ffuv0` and `fguv0` demonstrate similar accuracy in both Δfg and Δuv, particularly when extending Bundle Adjustment to K. Consequently, opting for `fguv0` as an initialization strategy in COLMAP is preferable for many practical applications. This preference is attributed to the faster processing speed of `fguv0` over `ffuv0` (1.78 s/iter compared to 9.21 s/iter) and its support for cameras without square pixel aspect ratio.

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. 2

[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. 2

[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

[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, 4

[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. 2

[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. 2

[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, 4

[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, 5, 6, 7

[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, 2, 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. 5, 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, 5, 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. 1, 2, 7, 8

[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