Minimal Constraint Relaxation for Multiview Autocalibration

Supplementary Material

This is the supplementary material for the paper *N. Kosaka, T. Duff, T. Pajdla. Minimal Constraint Relaxation for Multiview Autocalibration. CVPR 2026.*

Our supplementary materials provide an algebraic derivation of the classical Kruppa constraints in Sec. A, experimental protocols in Sec. B, numerical validation of relaxation properties in Sec. C, solver ablations in synthetic images in Sec. D, and extended real-data analyses in Sec. E referenced in the main paper.

A. Classical Kruppa Formulation

For completeness, we recall the classical *Kruppa equations* introduced by [17, 24] and presented in [15, Result 19.7]. This formulation establishes the algebraic relationship between the fundamental matrix $\mathbf{F}$, the epipole $\mathbf{e}'$, and the dual image of the absolute conic (DIAC) $\omega^* = \mathbf{K}\mathbf{K}^\top$. It provides a compact algebraic link between image correspondences and intrinsic parameters.

Given a view pair generating $(\mathbf{F}, \mathbf{e}')$, the Kruppa constraint can be expressed in matrix form as

$$[\mathbf{e}']_\times \, \omega^* \, [\mathbf{e}']_\times \sim \mathbf{F} \, \omega^* \, \mathbf{F}^\top. \tag{5}$$

Let $\mathbf{F} = \mathbf{U} \operatorname{diag}(\sigma_1, \sigma_2, 0) \, \mathbf{V}^\top$ be a singular value decomposition of the fundamental matrix. Substituting the left and right singular vectors $(\mathbf{u}_i, \mathbf{v}_i)$ and singular values σ_i into (5) yields the following vector cross-product constraint:

$$\begin{pmatrix} \mathbf{u}_2^\mathsf{T} \, \omega^* \, \mathbf{u}_2 \\ -\mathbf{u}_1^\mathsf{T} \, \omega^* \, \mathbf{u}_2 \\ \mathbf{u}_1^\mathsf{T} \, \omega^* \, \mathbf{u}_1 \end{pmatrix} \times \begin{pmatrix} \sigma_1^2 \, \mathbf{v}_1^\mathsf{T} \, \omega^* \, \mathbf{v}_1 \\ \sigma_1 \sigma_2 \, \mathbf{v}_1^\mathsf{T} \, \omega^* \, \mathbf{v}_2 \\ \sigma_2^2 \, \mathbf{v}_2^\mathsf{T} \, \omega^* \, \mathbf{v}_2 \end{pmatrix} = 0. \tag{6}$$

Equation (6) produces three quadratic equations in the six independent entries ω^*_{ij} of ω^*, of which two are linearly independent. Each image pair of three views thus contributes two constraints on the intrinsic parameters.

Notably, a single view pair contributes the same number of algebraically independent constraints in both the conventional and our formulations (Sec. 3)—each provides two constraints on ω^* (Table 1). A conventional formulation is just one particular "randomly" chosen relaxation. Overall, our experiments (Sec. 6) show that using our minimal-relaxation strategy, we choose a relaxation that achieves lower variance and improved empirical stability compared to the conventional formulation.

B. Experiment Details

In this section, we explain the details of the experiments in synthetic and real-world images.

B.1. Canonical Scene Sampling

To evaluate our method under controlled geometric conditions, we generate *canonical camera configurations* by randomly sampling relative poses between views. For each view pair (i, j), we independently sample rotations $\mathbf{R}_i, \mathbf{R}_j \in \mathrm{SO}(3)$ and translations $\mathbf{T}_i, \mathbf{T}_j \in \mathbb{R}^3$. These extrinsic parameters define the relative camera motion between the two views.

Assuming a fixed intrinsic calibration matrix $\mathbf{K}$, we compute the corresponding *fundamental matrix* for a view-pair (i, j) as:

$$\mathbf{F}_{ij} = \mathbf{K}^{-\top} \mathbf{R}_j [\mathbf{T}_j - \mathbf{T}_i]_\times \mathbf{R}_i^\top \mathbf{K}^{-1},$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix operator. The matrix $\mathbf{F}_{ij}$ is defined up to scale.

From this, the *epipole* $\mathbf{e}'$ in the second view of the pair is obtained as the left null vector of $\mathbf{F}_{ij}$, satisfying $F_{ij}^\top \mathbf{e}' = 0$. In the experiments, we used $\mathbf{R}, \mathbf{T}, \mathbf{F}$ with a sampled $\mathbf{K}$ as computed here.

B.2. Synthetic Image Setup and Image Coordinates

We describe here the details of our experimental setup on synthetic images under controlled conditions.

3D Scene. Following [6], we uniformly sample 100 3D points within a unit sphere. Three cameras are used. The first camera is positioned at $(0, 2, 0)$. The second and third cameras are obtained by translating the first by ± 0.5 units on each axis, ensuring a minimum baseline of 0.1 in ℓ_2-norm. Camera rotations are independently sampled from a uniform distribution over $[-\frac{\pi}{4}, \frac{\pi}{4}]$ radians per axis. All cameras are oriented to fully observe the 3D scene. Points are projected onto 640×480 image planes. Only points visible in all three views are retained.

Camera intrinsics variation. To test robustness, we varied the ground-truth camera intrinsics $\mathbf{K}$ by randomly sampling the horizontal field of view (FoV) in the range $[70°, 90°]$. The focal length was computed as $f_x = f_y = \frac{w}{2 \tan(\mathrm{FoV}_h/2)}$, assuming square pixels. The principal point (c_x, c_y) was perturbed around the image center by up to $\pm 5\%$ of the image width and height, respectively. The intrinsic matrix was then constructed as

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

and used to generate synthetic fundamental matrices and corresponding Kruppa constraints under varying calibration conditions.

Normalized Device Coordinates (NDC). To ensure scale-invariant image coordinates, all pixel positions were mapped to the Normalized Device Coordinate space $[-1, 1] \times [-1, 1]$. Given image width W and height H, this is achieved by a linear scaling matrix N that converts pixel coordinates (u, v) to normalized coordinates $(x_{\text{ndc}}, y_{\text{ndc}})$:

$$[x_{\text{ndc}},\, y_{\text{ndc}},\, 1]^\top = \mathbf{N}[u,\, v,\, 1]^\top, \quad \mathbf{N} = \begin{bmatrix} \frac{2}{W} & 0 & \frac{1}{W} - 1 \\ 0 & \frac{2}{H} & \frac{1}{H} - 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

This ensures that the image center lies near the origin, and coordinates remain consistent across different resolutions.

Noise Model. Zero-mean Gaussian noise $\mathcal{N}(0, \sigma^2)$ is added to the projected pixel coordinates, with $\sigma \in [0.0, 1.0]$.

Epipolar Solvers. To generate the triplet of fundamental matrices required for our autocalibration pipeline, we employ standard 6-, 7-, and 8-point epipolar solvers. The 8-point solver uses the classical non-minimal formulation, while the 7-point solver applies the minimal algorithm yielding up to three solutions. For improved numerical stability on real data, we additionally use the three-view minimal 6-point solver of [29], which recovers a projective reconstruction from six correspondences across three views; compatible fundamental matrices are then extracted from the recovered cameras. For all solvers, the inputs are normalized as in [6, 16].

B.3. Implementation Details

All solvers are implemented in `Julia` using `HomotopyContinuation.jl` [3], which enables efficient polynomial system solving with numerical path tracking. Experiments were run on an Intel(R) Core(TM) Ultra 7 165U CPU (1.70 GHz) with 16 GB RAM, without GPU acceleration.

B.4. Dataset Details: Image Sizes

We provide additional information on the real-world multi-view datasets used in our experiments. Table 3 summarizes the image resolutions and the number of views for each evaluated scene. These datasets cover diverse camera setups and scene complexities, serving as standard benchmarks for structure-from-motion and multi-view calibration.

C. Analysis of Constraint Relaxations

We numerically validate the algebraic properties of the relaxation families, including underconstrained patterns and permutation equivalences.

Table 3. Image resolutions and number of images for all evaluated real-world multi-view datasets.

Dataset	Scene	Resolution (pixels)	#Images
Strecha SfM	Herz-Jesús-P8	3072×2048	8
	Castle-P19	3072×2048	19
	Entry-P10	3072×2048	10
	Fountain-P11	3072×2048	11
Multi-view stereo	City Hall Leuven	3072×2048	7
	Semper Statue Dresden	3072×2048	3
	City Hall Brussels	2048×1360	3
COLMAP Public	South Building	3072×2304	100
	Gerrard Hall	5616×3744	128

C.1. Numerical Validation of Underconstrained Relaxations

To corroborate the above symbolic findings, we conduct a numerical evaluation of the same underconstrained relaxations under controlled image perturbations (SM: B.1). While the symbolic analysis certifies algebraic dependence exactly over $\mathbb{Q}$, the numerical experiment empirically confirms that these dependencies persist in practice, even in the presence of image noise and floating-point arithmetic.

Experiment. We used `Julia` to evaluate the entire $(1, 2, 2)$-family of 165k relaxations across 70 random scenes, verifying whether the relaxations containing the 9 pairs detected by the symbolic experiment (Sec. 5.1) remained underconstrained (UC). We observed that 100% of such relaxations were UC, exhibiting Jacobian rank 4 for 25,920 cases and rank 3 for 1,215 cases, in full agreement with Table 1. This consistency across random scenes reflects the expected *genericity* [7] of the rank behavior.

C.2. Numerical validation of permutation equivalences

Following the main experiments (Sec. 6), we used the `Polyhedral` Homotopy Continuation solver [33] to evaluate permutation and swapping invariance within the $(1, 2, 2)$ relaxation family.

Setup. We evaluated the Top-10 and Bottom-10 relaxations, where "Top-10" refers to the ten relaxation patterns with the smallest errors and "Bottom-10" refers to the ten with the largest errors among $(1, 2, 2)$ relaxation family, as determined by the offline scoring in Sec. 5.3. Experiments were conducted under noiseless image conditions with NDC scaling of image coordinates (SM: B.3) and varying the intrinsic calibration matrix K across trials. Each relaxation was tested on *100 random scenes* with varying K. Performance was measured using the mean and variance of the summed relative error in the intrinsic parameters $(\Delta f + \Delta g + \Delta u + \Delta v + \Delta s)$ across all permutation instances.

Results. Under this protocol, the selected $(1, 2, 2)$ patterns and their swapped counterparts exhibited numerically indistinguishable performance: for the Top-10 relaxations, the mean variance of summed errors was $6.0 \times 10^{-11} \pm 2.9 \times 10^{-21}$, and for the Bottom-10, 0.0205 ± 0.0013. These results confirm the theoretical prediction that permutations and intra-pattern swaps of the $(1, 2, 2)$ configuration are algebraically equivalent, producing identical numerical behavior within floating-point precision.

C.3. Relaxations on Different Calibration Tasks

In many real-world scenarios, some camera intrinsics may be known in advance. Following [6], we encode the status of intrinsic parameters using a 5-character code `fguvs`, where each character represents whether a specific parameter is known or unknown. Known values are replaced with their normalized counterparts. For example, the pattern `f1uv0` means that f, u, and v are unknown, while $g = 1$ and $s = 0$ are fixed. A commonly encountered case is when the pixels are assumed to be square; this is represented by `ffuvs`, indicating that f and g are unknown but constrained to be equal. Scene sampling method is explained in SM: B.1 and relaxation patterns follow Sec.4.1.

Calibration tasks We follow [6] in categorizing variations of autocalibration tasks based on different prior knowledge of the camera intrinsics.

For each task, we compute the form of the dual DIAC $\omega^* = \mathbf{K}\mathbf{K}^\top$ where $\mathbf{K}$ reflects the assumed intrinsic constraints. For instance:

- If the task is **`fguvs`** (all intrinsics unknown), then

$$\mathbf{K} = \begin{bmatrix} f & s & u \\ 0 & g & v \\ 0 & 0 & 1 \end{bmatrix},$$

$$\omega^* = \begin{bmatrix} f^2 + s^2 + u^2 & gs + uv & u \\ gs + uv & g^2 + v^2 & v \\ u & v & 1 \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & w_4 \\ w_2 & w_3 & w_5 \\ w_4 & w_5 & 1 \end{bmatrix}.$$

- If the task is **`ff000`** (square pixels with known principal point and zero skew), then

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

$$\omega^* = \begin{bmatrix} f^2 & 0 & 0 \\ 0 & f^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We classify all calibration cases according to the structural form of ω^* in Table 4. Whenever a particular ω^* class contains fewer than five target variables, we select exactly **Vars** equations so that the resulting subsystem remains square, consistent with the full five-variable calibration setting of (**`fguvs`**).

Metrics Table 5 reports symbolic and numerical properties of all dimension-zero calibration tasks. Each row corresponds to a specific calibration pattern.

- $\boldsymbol{\omega}_{\mathbf{class}}$: Structural class of the ω^* form (see Table 4).
- **Views**: Minimum number of camera views required for a subsystem *in our formulation of Kruppa systems* (Sec. 3) to become *zero-dimensional*.
- **[6] + Min/Max Deg ([6])**: Degree bound from depth-based formulation of [6], giving a crude estimate of algebraic complexity of the prior work.
- **Min/Max Deg (Ours)**: Range of isolated complex solutions obtained numerically via monodromy variant of HC [8].

Results. Table 5 summarizes the algebraic complexity of various autocalibration tasks.

Depth-based formulation of [6] encode the calibration intrinsics directly in the form of the image of absolute conic (IAC) $\omega = \mathbf{K}^{-\top}\mathbf{K}^{-1}$, while our epipolar-based formulation including the conventional Kruppa systems employ a map from $\mathbf{K}$ to $\omega^* = \mathbf{K}\mathbf{K}^\top$ reducing the degree of the final system by roughly half ($4 \to 2$). Also, the formulation of [6] involves depth variables enlarging the solution space while ours can purely keep the target variables to the intrinsics, achieving such low degree eventually as in Tab.5.

Several classical approaches have addressed focal length calibration under minimal-view settings, especially two-view scenarios. However, they typically make strong assumptions on the intrinsic parameters. For instance, methods such as those by [1, 18, 36] rely on simplifying constraints (e.g., known or equal focal lengths) to reduce the algebraic burden.

Beyond these, certain works have characterized which sets of intrinsics admit Euclidean reconstruction from minimal views. In particular, [19] showed that Euclidean reconstruction is feasible under configurations such as $(\alpha f, g, u, v, s)$, (f, f, u, v, s), or $(f, f, u, v, 0)$, where f denotes focal length(s), g is aspect ratio, u, v are principal point coordinates, and s is skew. Our analysis in Table 5 generalizes this line of inquiry by exhaustively classifying various minimal (dimension-zero) calibration problems up to five intrinsics.

Remark. We note that underlined entries in Table 5 correspond to numerically sensitive cases in which monodromy HC may have under-explored the solution space; increasing the number of explored paths could plausibly reveal additional solutions.

Table 4. Symbolic forms of ω^* matrices grouped by number of intrinsic variables (**Vars**) and configuration label ($\boldsymbol{\omega}_{\mathbf{class}}$).

Vars	$\boldsymbol{\omega}_{\mathbf{class}}$	ω^*	**Vars**	$\boldsymbol{\omega}_{\mathbf{class}}$	ω^*
5	0	$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_2 & w_4 & w_5 \\ w_3 & w_5 & 1 \end{bmatrix}$	2	0	$\begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
4	0	$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_2 & w_4 & 0 \\ w_3 & 0 & 1 \end{bmatrix}$	2	1	$\begin{bmatrix} w_1 & 0 & w_2 \\ 0 & 1 & 0 \\ w_2 & 0 & 1 \end{bmatrix}$
4	1	$\begin{bmatrix} w_1 & w_2 & 0 \\ w_2 & w_3 & w_4 \\ 0 & w_4 & 1 \end{bmatrix}$	2	2	$\begin{bmatrix} w_1 & w_2 & 0 \\ w_2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
3	0	$\begin{bmatrix} w_1 & 0 & w_2 \\ 0 & w_3 & 0 \\ w_2 & 0 & 1 \end{bmatrix}$	2	3	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & w_1 & w_2 \\ 0 & w_2 & 1 \end{bmatrix}$
3	1	$\begin{bmatrix} w_1 & w_2 & 0 \\ w_2 & w_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	1	0	$\begin{bmatrix} w_1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
3	2	$\begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & w_3 \\ 0 & w_3 & 1 \end{bmatrix}$	1	0	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & w_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
3	3	$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_2 & 1 & 0 \\ w_3 & 0 & 1 \end{bmatrix}$	1	0	$\begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

D. Solver Ablations in Synthetic Images

We present ablations of solver variants, analyzing numerical stability, runtime, and the effect of NDC and parameter homotopies.

D.1. Kruppa-BnB on Synthetic Images

To further assess convergence behavior, Figure 4 reports the performance of `Kruppa-BnB` solver *without local refinement* under the same synthetic noise conditions, across the 6-, 7-, and 8-point epipolar configurations as the main experiments (Sec.6.1). These synthetic evaluations serve as a controlled baseline for the subsequent real-world experiments.

Analysis. The branch-and-bound (BnB) strategy operates purely in the numerical parameter space of candidate solutions, optimizing the algebraic objective without explicitly enforcing the structural constraints of the dual image of the absolute conic (DIAC). As a result, the numerically optimal solution frequently violates the positive-definiteness of the DIAC, rendering the recovered intrinsics invalid. Moreover, the BnB sampling is inherently coarse, exploring the search domain sparsely; without a subsequent local refinement stage (e.g., Levenberg–Marquardt), the solver often fails to converge to a physically meaningful or geometrically consistent solution. This explains the degraded performance observed across all configurations in Figure 4.

D.2. Scale interpretation of Intrinsic Errors

For the 640×480 synthetic setup, the typical focal lengths of the calibrated cameras fall in the range $f \in [500, 700]$ px, corresponding to a horizontal field of view of approximately $45°$–$60°$. Under this setting, a unit normalized intrinsic error ($\Delta f = \frac{|\hat{f}-f|}{f}$) ($\Delta_* = 1$) would imply an absolute deviation of about 500–700 px in the corresponding parameter. Hence, a deviation of 50–80 px in focal length or principal point translates to $\Delta_* \approx 0.1$, which we use as a practical reference scale in the main results.

Task	Vars	ω_{class}	Views	[6]	Min Deg ([6])	Max Deg ([6])	Min Deg (Ours)	Max Deg (Ours)
1g000	1	0	2	2	30	30	1	2
f1000	1	0	2	2	30	30	1	2
ff000	1	0	2	2	30	30	1	2
fg000	2	0	2	2	18	18	2	17
11u00	2	1	2	2	60	60	1	4
f1u00	2	1	2	2	18	18	1	4
1100s	2	2	2	2	60	60	3	4
f100s	2	2	2	2	18	18	1	4
110v0	2	3	2	2	60	60	2	2
1g0v0	2	3	2	2	18	18	1	3
1gu00	3	0	3	2	36	36	1	7
ffu00	3	0	3	2	24	24	1	8
fgu00	3	0	3	3	9686	33606	1	8
1g00s	3	1	3	2	24	24	1	8
ff00s	3	1	3	2	36	36	2	8
fg00s	3	1	3	3	9686	33606	2	8
f10v0	3	2	3	2	36	36	1	8
ff0v0	3	2	3	3	4484	92336	6	27
fg0v0	3	2	3	3	9686	112520	6	27
f1u0s	3	3	2	3	14760	160190	1	1
11u0s	3	3	3	2	36	36	1	8
ffu0s	4	0	3	3	29028	100758	3	16
fgu0s	4	0	3	3	2058	2058	3	16
110vs	4	1	3	2	72	72	1	15
1g0vs	4	1	3	3	29012	315653	2	17
f10vs	4	1	3	3	24332	86539	1	16
ff0vs	4	1	3	3	38700	134352	5	16
fg0vs	4	1	3	3	2058	2058	5	16
11uv0	5	0	3	2	48	48	18	18
11uvs	5	0	3	3	57912	201265	1	10
ffuv0	5	0	3	3	16188	119119	4	32
ffuvs	5	0	3	3	4617	4617	2	32
fguv0	5	0	3	3	2313	2313	4	32
fguvs	5	0	3	3	2985	1136202	18	32

Table 5. Min–max root counts for all calibration tasks, grouped by the number of intrinsic variables (**Vars**) and the configuration label ($\boldsymbol{\omega}_{\text{class}}$). Values are aggregated over 10 randomly generated scenes, with relaxation patterns computed as described in Sec. 4.1. *Underlined values indicate numerically sensitive cases where the estimated count may vary across scenes.*

D.3. Runtime Experiments

We compare the runtimes of the three solver classes integrated into our autocalibration pipeline: `Kruppa-BnB`, `Kruppa` solver, and our `Global-Best` relaxed solver. Both `Kruppa` and `Global-Best` are solved using the same `PolyhedralHC` backend; since they induce identical mixed volumes and therefore the same number of homotopy paths, their runtimes are nearly equal. In contrast, `Kruppa-BnB` performs only a lightweight divide-and-conquer search with no homotopy continuation, making it substantially faster but also less accurate (see Table 14).

Across 100 independent runs per method (Synthetic images experiment in Sec. 6.1), we obtain average runtimes of 1.96×10^{-3} s for `Kruppa-BnB`, 5.38×10^{-2} s for `Kruppa`, and 4.76×10^{-2} s for `Global-Best`. As expected, the two homotopy-based solvers exhibit almost identical runtimes, while BnB offers speed at the cost of poorer intrinsic accuracy.

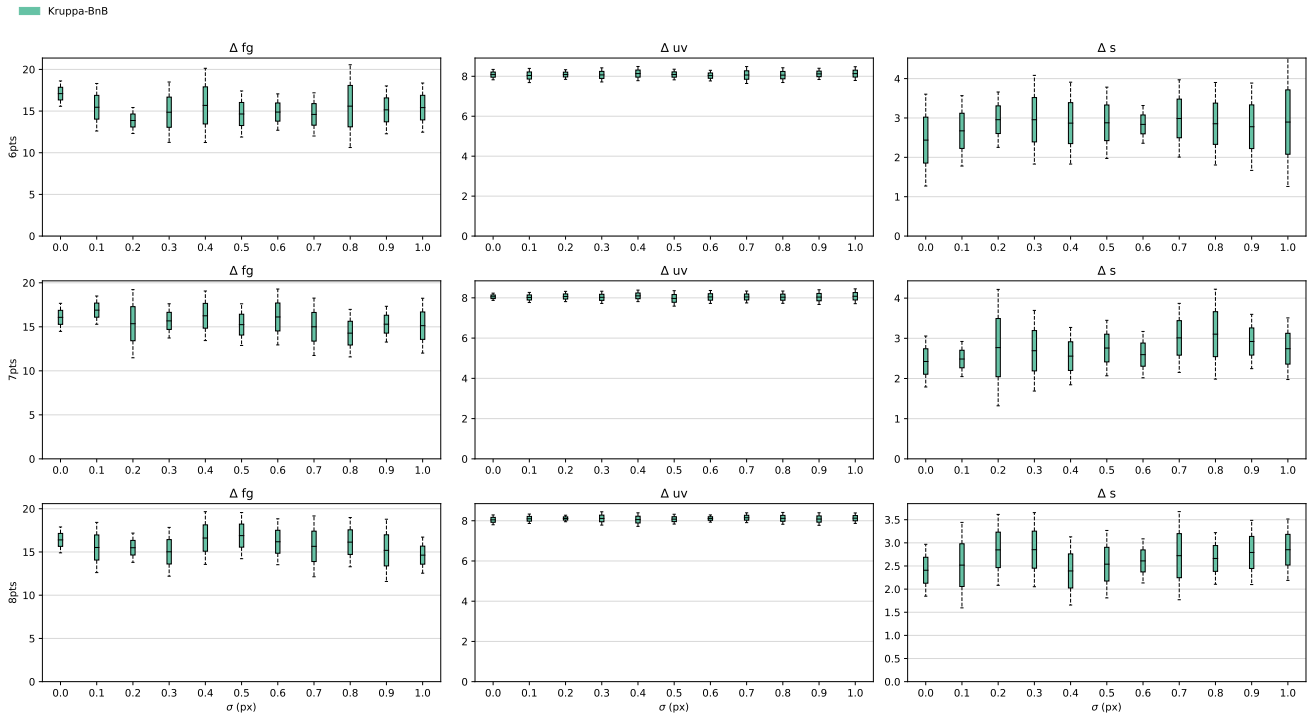

Figure 4. Synthetic performance of *Kruppa-BnB* **without local refinement** across noise levels for the 6-, 7-, and 8-point epipolar solver configurations.

D.4. Ablation: Enhancements on Our Solver

We evaluate several incremental enhancements to our solver—`PolyhedralHC`, `PolyhedralHC+NDC`, `ParamHC`, and `NDC+ParamHC`—to analyze their contribution to calibration accuracy. In the main paper (Sec. 6), we employ the `Global-Best + NDC` configuration as our default solver, as it consistently provides the most stable and accurate results across noise levels.

Variations

- **PolyhedralHC:** The base `Global-Best` solver without any additional enhancements.
- **PolyhedralHC+NDC:** Applies *Normalized Device Coordinate (NDC) scaling* (SM: B), mapping image coordinates to the $(-1, 1)$ range to improve numerical conditioning.
- **ParamHC:** Utilizes the `parameter homotopy continuation` (ParamHC) technique from the `HomotopyContinuation.jl` package, which connects the start system (parameters and solutions) obtained by Monodromy HC [8] on a canonical, zero-noise scene to the target noisy system (SM: B.1). *Here, we connect the start system with* 24 *solutions computed by the monodromy HC on an offline scene (SM: B.1).*
- **NDC+ParamHC:** Combines both NDC scaling and parameter homotopy continuation for improved stability and accuracy.

Metrics. We measure calibration accuracy using the summed mean parameter error $E_{\text{sum-mpe}} = \Delta f + \Delta g + \Delta u + \Delta v + \Delta s$ which aggregates the relative errors of all intrinsic parameters. This metric provides a single scalar quantity that reflects overall intrinsic recovery quality across different solver variants.

Results. Across all minimal configurations (6pts, 7pts, and 8pts), `PolyhedralHC+NDC` provides the most consistent improvement, lowering $E_{\text{sum-mpe}}$ and improving numerical stability at every noise level. In contrast, `ParamHC` alone exhibits mixed behavior—occasionally competitive at low noise but generally less stable than the NDC-enhanced variants. Combining both techniques in `NDC+ParamHC` yields moderate additional gains in several regimes, though it does not surpass `PolyhedralHC+NDC` on average. Overall, `PolyhedralHC+NDC` (our `Global-Best+NDC` configuration) achieves the lowest mean error across noise levels for all epipolar solver settings, supporting its use as the default variant in the main paper. We note that `ParamHC` is theoretically more efficient than `PolyehdralHC` for this problem, as the for-

Table 6. **Calibration accuracy on *6pts***. Each cell reports mean ± standard deviation of summed mean projection error ($E_{\text{sum-mpe}}$) across noise levels σ (px). Lower is better. Best per row is **bold**. Bottom row averages over the listed σ values.

Noise σ	PolyhedralHC	PolyhedralHC+NDC	ParamHC	NDC+ParamHC
0.0	1.52e-03 ± 4.80e-03	**1.01e-10 ± 1.18e-10**	4.34e-01 ± 1.51e-01	1.40e-01 ± 9.86e-02
0.1	3.0065 ± 0.3699	**2.8887 ± 0.4446**	3.0725 ± 0.4296	2.9206 ± 0.3836
0.2	3.6087 ± 1.0340	**3.2262 ± 0.4632**	3.9275 ± 0.7811	3.3960 ± 0.6451
0.3	3.5135 ± 0.2227	**3.4025 ± 0.5359**	3.8725 ± 0.5414	3.6624 ± 0.4698
0.4	3.8935 ± 0.7452	**3.4442 ± 0.1603**	4.1772 ± 0.4239	3.6786 ± 0.5184
0.5	3.9387 ± 0.4367	**3.5806 ± 0.3915**	4.1187 ± 0.6504	3.9541 ± 0.6418
0.6	4.4001 ± 0.7023	**3.7270 ± 0.3938**	4.0262 ± 0.4794	4.0070 ± 0.8365
0.7	4.3560 ± 0.5774	**3.8299 ± 0.3715**	4.1587 ± 0.2131	3.8537 ± 0.5579
0.8	4.2185 ± 0.3884	**3.8881 ± 0.6714**	4.5957 ± 0.6172	3.9017 ± 0.5546
0.9	4.0678 ± 0.3588	**4.0163 ± 0.8889**	4.4208 ± 0.6656	4.0467 ± 0.4286
1.0	4.3096 ± 0.5276	**3.8464 ± 0.3645**	4.4917 ± 0.6572	4.0623 ± 0.6520
Avg over σ	3.5740 ± 0.4880	**3.2591 ± 0.4260**	3.7541 ± 0.5099	3.4203 ± 0.5261

Table 7. **Calibration accuracy on *7pts***. Each cell reports mean ± standard deviation of summed mean projection error ($E_{\text{sum-mpe}}$) across noise levels σ (px). Lower is better. Best per row is **bold**. Bottom row averages over the listed σ values.

Noise σ	PolyhedralHC	PolyhedralHC+NDC	ParamHC	NDC+ParamHC
0.0	**1.81e-10 ± 1.09e-10**	4.46e-03 ± 1.37e-02	5.80e-01 ± 2.68e-01	1.23e-01 ± 6.24e-02
0.1	2.8110 ± 0.4053	**2.1378 ± 0.1816**	2.8150 ± 0.2966	2.2748 ± 0.3858
0.2	3.2651 ± 0.3492	2.9666 ± 0.5219	3.4491 ± 0.5105	**2.8552 ± 0.4938**
0.3	3.5160 ± 0.3840	**3.0236 ± 0.3246**	3.8801 ± 0.5779	3.1081 ± 0.3466
0.4	3.7800 ± 0.3100	3.3697 ± 0.3517	4.2693 ± 0.6923	**3.3091 ± 0.4291**
0.5	4.0567 ± 0.2913	**3.4730 ± 0.3753**	4.1828 ± 0.7326	3.5560 ± 0.3920
0.6	3.9279 ± 0.3901	**3.4080 ± 0.3694**	4.1097 ± 0.7537	3.6847 ± 0.5146
0.7	3.9226 ± 0.3108	**3.5063 ± 0.2645**	4.9137 ± 1.2500	4.0041 ± 0.5884
0.8	4.0961 ± 0.4202	**3.3945 ± 0.3237**	4.3011 ± 0.5825	3.8480 ± 0.5229
0.9	4.3010 ± 0.3144	**3.6987 ± 0.6622**	4.4881 ± 0.7567	3.7686 ± 0.3917
1.0	4.3862 ± 0.3229	**3.6997 ± 0.3370**	4.5907 ± 0.6716	3.9224 ± 0.4622
Avg over σ	3.4602 ± 0.3180	**2.9711 ± 0.3387**	3.7799 ± 0.6447	3.1322 ± 0.4172

mer requires tracking only 24 homotopy paths and the latter must track 32. Surprisingly, though, `PolyhedralHC` exhibits higher accuracy in the zero-noise regime. Since the accuracy of both methods can be improved by tighter Newton refinement at the target solutions, this is likely due to implementation differences rather than an inherent strength of the polyhedral method.

D.5. Local-Refinement with initialization

We evaluate all solvers when followed by local nonlinear refinement. Specifically, we apply Levenberg–Marquardt (LM) optimization to refine the intrinsic matrix $\mathbf{K}$ by minimizing the *algebraic error*, defined as the norm of the constraint residuals. We adopt this objective to match the refinement strategy of [28], ensuring a fair and consistent comparison across methods. This evaluation allows us to assess (i) the quality of each solver's initial estimate and (ii) the additional improvement gained through local optimization.

Methods

- **Kruppa-BnB (LM)**: Full Kruppa branch-and-bound pipeline followed by LM refinement [28].
- **PolyhedralHC+NDC**: Our proposed solver (Sec. 6.1).
- **Kruppa**: Conventional Kruppa solver (Sec. 6.1).
- **SOLVER (LM-Cand-O)**: LM refinement applied to *all* candidate solutions (those returned by `PolyhedralHC` and those satisfying the positive-definiteness constraint).
- **SOLVER (LM-Best-K)**: LM refinement applied only to the single candidate minimizing the sum of relative errors (i.e., the candidate used for error evaluation in the main experiments; Sec. 6.1).

Note that `Kruppa-BnB` and `Kruppa` minimize the 6 classical Kruppa constraints [24], whereas our method (`PolyhedralHC+NDC`) minimizes the full set of 45 con-

Table 8. **Calibration accuracy on *8pts***. Each cell reports mean ± standard deviation of summed mean projection error ($E_{\text{sum-mpe}}$) across noise levels σ (px). Lower is better. Best per row is **bold**. Bottom row averages over the listed σ values.

Noise σ	PolyhedralHC	PolyhedralHC+NDC	ParamHC	NDC+ParamHC
0.0	5.80e-03 ± 1.83e-02	**1.06e-11 ± 5.57e-12**	4.70e-01 ± 1.73e-01	1.16e-01 ± 1.23e-01
0.1	2.4947 ± 0.3566	**2.1450 ± 0.3774**	2.7136 ± 0.3095	2.2150 ± 0.3095
0.2	3.0843 ± 0.3568	2.7204 ± 0.4696	3.3677 ± 0.6789	**2.6817 ± 0.3519**
0.3	3.3008 ± 0.4114	2.9761 ± 0.3983	3.6343 ± 0.3198	**2.9150 ± 0.3881**
0.4	3.7514 ± 0.3811	3.1044 ± 0.3142	3.5832 ± 0.3908	**3.0621 ± 0.3193**
0.5	3.7977 ± 0.3667	3.4180 ± 0.7901	3.9756 ± 0.4526	**3.3368 ± 0.2272**
0.6	3.7316 ± 0.3075	**3.3245 ± 0.3178**	3.7979 ± 0.5373	3.4608 ± 0.3653
0.7	3.7189 ± 0.4770	**3.6570 ± 0.3725**	3.9334 ± 0.5272	3.7314 ± 0.3732
0.8	3.9356 ± 0.3191	**3.7081 ± 0.4688**	3.8177 ± 0.4159	3.9359 ± 0.5438
0.9	4.0872 ± 0.4888	**3.7702 ± 0.3601**	4.0279 ± 0.4557	3.8016 ± 0.3598
1.0	4.1103 ± 0.4565	3.8649 ± 0.4667	4.2289 ± 0.4703	**3.7798 ± 0.2702**
Avg over σ	3.2744 ± 0.3582	**2.9717 ± 0.3941**	3.4136 ± 0.4301	3.0032 ± 0.3301

straints in our formulation (Sec. 3).

Results Tables 9, 10, and 11 report

$$E_{\text{sum-mpe}} = \Delta f + \Delta g + \Delta u + \Delta v + \Delta s,$$

the summed mean projection error, which aggregates the relative error in all intrinsic parameters across 100 random scenes.

Overall, LM refinement improves most solvers across noise levels. However, even after LM refinement, *our method consistently outperforms both Kruppa and Kruppa-BnB* across all epipolar configurations (6 pts, 7 pts, and 8 pts).

Finally, the comparison between `LM-Best-K` and `LM-Cand-O` reveals that refining *all* candidates (`LM-Cand-O`) is often beneficial, especially at higher noise levels where the best algebraic candidate may not correspond to the best geometric solution.

E. Real-World Experiments

We report extended real-data results, including relaxation variability and the impact of solver-based initialization within COLMAP.

E.1. Results with `Local-Best` and `Local-Worst`

We report updated results for `Kruppa-BnB` after tuning its search domain for real-world data. Table 12 summarizes the sum of relative intrinsic errors across all solvers, including the `Local-Best` and `Local-Worst` relaxations (defined in Sec. 6 as the relaxations achieving the smallest and largest errors for each scene, selected from the full set of 84k relaxations described in Sec. 5.3). These results allow us to characterize the full variability of the relaxation family in practical SfM conditions, while also showing that the `Global-Best` relaxation identified in the synthetic experiments remains consistently competitive on real data.

As an additional breakdown, Table 13 reports grouped factor-level errors ($\Delta fg, \Delta uv, \Delta s$), highlighting how each solver behaves with respect to individual intrinsic components. Together, these two tables provide a more complete assessment of solver performance and illustrate how the spread between `Local-Best` and `Local-Worst` reflects the inherent sensitivity of different Kruppa relaxations on real imagery.

E.2. Evaluation of initialization Quality in 3D Reconstruction (COLMAP)

We integrate our autocalibration solvers into a modern 3D reconstruction pipeline (COLMAP [30]) to initialize camera intrinsics prior to sparse reconstruction. The objective is to assess whether more accurate focal lengths and principal points—estimated via minimal or relaxed solvers—can improve upon COLMAP's standard heuristic initialization and thereby enhance downstream reconstruction accuracy. We evaluate on the same real-world scenes from `SfM-eval` (`Castle-P19`, `Entry-P10`, `Fountain-P11`, and `Herz-Jesus-P8`) used in Sec. 6.2.

initialization strategies and evaluation protocol. Following [6], we compare several strategies for initializing and refining the intrinsic matrix $\mathbf{K}$ within COLMAP's incremental SfM pipeline:

- $\text{COLMAP}_{\text{guess}}$: COLMAP's default focal-length heuristic based on image size (Tab. 3).
- $\text{COLMAP}_{\text{Kruppa}} + \mathbf{K}_{\text{BA}}$: initialization with the `Kruppa` solver, followed by bundle adjustment (BA).
- $\text{COLMAP}_{\text{Kruppa-BnB}} + \mathbf{K}_{\text{BA}}$: initialization with the `Kruppa-BnB` solver, followed by BA.

Table 9. **Calibration accuracy on *6pts***. Each cell reports mean ± standard deviation of summed mean projection error (sum_mpe) across noise levels σ (px). Lower is better. Best per row is **bold**. Bottom row averages over the listed σ values.

Noise σ	Kruppa-BnB (LM)	PolyhedralHC+NDC	Kruppa	Kruppa (LM-Cand-O)	Kruppa (LM-Best-K)	Ours (LM-Cand-O)	Ours (LM-Best-K)
0.0	5.56e+00 ± 1.74e+00	1.01e-10 ± 1.18e-10	3.23e-11 ± 4.62e-11	**3.06e-11 ± 4.49e-11**	3.76e+00 ± 7.46e-02	5.54e-11 ± 7.25e-11	3.87e+00 ± 3.73e-02
0.1	5.3699 ± 1.4577	2.8887 ± 0.4446	3.3727 ± 1.8979	2.6750 ± 1.8826	3.8512 ± 0.1741	**2.0136 ± 0.5479**	3.9302 ± 0.0890
0.2	4.2706 ± 1.7204	3.2262 ± 0.4632	4.3261 ± 3.8432	3.7016 ± 4.0238	3.9336 ± 0.1083	**2.0281 ± 0.5581**	3.9755 ± 0.0758
0.3	3.6638 ± 0.9816	3.4025 ± 0.5359	4.1436 ± 2.1860	3.6310 ± 2.6825	3.9882 ± 0.1937	**2.2344 ± 0.6196**	3.9786 ± 0.0749
0.4	4.4808 ± 2.3328	3.4442 ± 0.1603	4.4832 ± 2.2374	3.6349 ± 2.7050	3.9840 ± 0.1933	**2.7244 ± 1.3689**	3.9603 ± 0.1175
0.5	3.5895 ± 0.7079	3.5806 ± 0.3915	4.4247 ± 1.7736	4.7737 ± 2.4174	4.0473 ± 0.1748	**2.7968 ± 1.1543**	4.0236 ± 0.0833
0.6	4.5584 ± 1.1655	3.7270 ± 0.3938	3.7333 ± 1.2687	3.8457 ± 1.1983	4.0250 ± 0.1751	**2.9266 ± 1.0902**	4.0016 ± 0.1028
0.7	4.3521 ± 1.3305	3.8299 ± 0.3715	4.1448 ± 1.5938	5.2367 ± 4.3671	4.0410 ± 0.1720	**3.6822 ± 1.4759**	4.0491 ± 0.0883
0.8	5.3112 ± 2.8882	3.8881 ± 0.6714	4.5432 ± 2.2860	4.9397 ± 2.8969	3.9721 ± 0.1739	**2.6550 ± 0.4884**	4.0086 ± 0.1228
0.9	4.2848 ± 1.9738	4.0163 ± 0.8889	4.3719 ± 2.2621	4.0608 ± 2.6365	3.9288 ± 0.1647	**2.9501 ± 0.7894**	4.0669 ± 0.1129
1.0	4.2873 ± 2.1300	3.8464 ± 0.3645	4.6036 ± 2.2322	4.6640 ± 2.3695	4.0711 ± 0.2310	**3.0287 ± 1.0155**	4.0817 ± 0.1071
Avg over σ	4.5204 ± 1.6755	3.2591 ± 0.4260	3.8316 ± 1.9619	3.7421 ± 2.4709	3.9638 ± 0.1669	**2.4582 ± 0.8280**	3.9950 ± 0.0920

Table 10. **Calibration accuracy on *7pts***. Each cell reports mean ± standard deviation of summed mean projection error (sum_mpe) across noise levels σ (px). Lower is better. Best per row is **bold**. Bottom row averages over the listed σ values.

Noise σ	Kruppa-BnB (LM)	PolyhedralHC+NDC	Kruppa	Kruppa (LM-Cand-O)	Kruppa (LM-Best-K)	Ours (LM-Cand-O)	Ours (LM-Best-K)
0.0	5.39e+00 ± 9.76e-01	4.46e-03 ± 1.37e-02	**2.68e-12 ± 2.34e-12**	**2.67e-12 ± 2.22e-12**	3.75e+00 ± 1.30e-01	1.45e-03 ± 4.58e-03	3.82e+00 ± 5.05e-02
0.1	6.0068 ± 2.0779	2.1378 ± 0.1816	2.1721 ± 0.7082	**1.4646 ± 0.5325**	3.7949 ± 0.0975	1.5295 ± 0.3084	3.9184 ± 0.0605
0.2	6.4435 ± 2.3978	2.9666 ± 0.5219	3.1661 ± 0.8034	2.5584 ± 0.7710	3.8054 ± 0.1251	**2.1768 ± 0.7011**	3.9613 ± 0.0861
0.3	6.3333 ± 1.6488	3.0236 ± 0.3246	3.1079 ± 1.1658	**2.4384 ± 0.9877**	3.8423 ± 0.1241	2.5574 ± 1.0016	3.9617 ± 0.1205
0.4	5.7113 ± 1.0574	3.3697 ± 0.3517	2.7338 ± 1.4230	**2.4454 ± 1.4878**	3.8245 ± 0.0830	2.4465 ± 0.8809	3.9694 ± 0.1366
0.5	6.0917 ± 3.2411	3.4730 ± 0.3753	4.2107 ± 1.4827	3.7900 ± 2.2756	3.9093 ± 0.1211	**2.4851 ± 0.8763**	3.9432 ± 0.0654
0.6	5.4062 ± 0.9770	3.4080 ± 0.3694	**3.3313 ± 1.3775**	3.4716 ± 2.2902	3.8589 ± 0.0736	3.5036 ± 1.4175	3.9713 ± 0.1289
0.7	6.2082 ± 2.6637	3.5063 ± 0.2645	3.8586 ± 1.5651	3.6567 ± 1.2780	3.8248 ± 0.1378	**3.0701 ± 0.8613**	3.9894 ± 0.1204
0.8	5.7653 ± 2.1183	3.3945 ± 0.3237	3.3649 ± 1.0850	**2.9733 ± 0.9826**	3.7966 ± 0.2005	3.1423 ± 0.9917	4.0159 ± 0.1183
0.9	5.1874 ± 1.1243	3.6987 ± 0.6622	3.5656 ± 0.4855	3.1630 ± 0.8838	3.9048 ± 0.1316	**3.0027 ± 0.6604**	3.9886 ± 0.1016
1.0	5.5322 ± 0.8210	3.6997 ± 0.3370	3.5871 ± 0.6284	3.2744 ± 0.7554	3.9061 ± 0.1284	**3.2176 ± 0.8266**	4.0620 ± 0.1382
Avg over σ	5.8253 ± 1.7367	2.9711 ± 0.3387	3.0089 ± 0.9750	2.6578 ± 1.1131	3.8382 ± 0.1230	**2.4666 ± 0.7755**	3.9639 ± 0.1025

- $\text{COLMAP}_{\text{Global-Best}} + \mathbf{K}_{\text{BA}}$: initialization with our `Global-Best` relaxation solver, followed by BA.

We additionally include an oracle baseline, $\text{COLMAP}_{\text{GT}} + \mathbf{K}_{\text{BA}}$, initialized from the ground-truth calibration parameters.

All reconstructions are performed in single-camera mode, ensuring that all views share a common intrinsic matrix $\mathbf{K}$. After reconstruction, we compare the estimated intrinsics $\hat{\mathbf{K}}$ with the ground-truth calibration $\mathbf{K}_{\text{gt}}$, reporting relative focal-length and principal-point errors ($\Delta f g$, Δuv) as in [6]. To ensure comparability, all experiments use identical feature extraction, exhaustive matching, and incremental mapping parameters within COLMAP.

Results. Table 14 reports the results using the seed that yielded the smallest error for each solver. Across all scenes, solver-based methods achieve intrinsic errors that are two–three orders of magnitude lower than COLMAP's heuristic ($\text{COLMAP}_{\text{guess}}$). *Within the solver-based initializers, the Global-Best estimate generally performs favorably, providing a reliable intrinsic prediction prior to SfM.*

Despite their high intrinsic accuracy, solver-based initializers show limited success within COLMAP's incremental mapper: `Kruppa-BnB` rarely succeeds, the minimal `Kruppa` solver works only intermittently (SR = 0.1–0.5), and even our `Global-Best` estimate—while more robust than the other solvers—achieves SR = 0.1–0.6 across datasets. By contrast, COLMAP's heuristic succeeds on every scene (SR = 1.0). Inspection of the COLMAP logs indicates that this behavior arises from a geometric—not numerical—mismatch: although inaccurate, `COLMAP's guess` remains geometrically well conditioned for essential matrix estimation and triangulation, whereas solver-based intrinsics can be geometrically extreme in pixel coordinates, destabilizing early epipolar geometry and causing COLMAP to reject all initial image pairs.

In summary, COLMAP's incremental SfM pipeline is tolerant of *large but geometrically coherent* intrinsic errors, yet brittle to *small but geometrically ill-conditioned* ones—explaining why high-accuracy solver estimates may still fail to bootstrap a 3D reconstruction.

Table 11. **Calibration accuracy on *8pts***. Each cell reports mean ± standard deviation of summed mean projection error (sum_mpe) across noise levels σ (px). Lower is better. Best per row is **bold**. Bottom row averages over the listed σ values.

Noise σ	Kruppa-BnB (LM)	PolyhedralHC+NDC	Kruppa	Kruppa (LM-Cand-O)	Kruppa (LM-Best-$\mathbf{K}$)	Ours (LM-Cand-O)	Ours (LM-Best-$\mathbf{K}$)
0.0	4.81e+00 ± 2.09e+00	1.06e-11 ± 5.57e-12	**3.15e-12 ± 3.66e-12**	**2.92e-12 ± 3.61e-12**	3.80e+00 ± 6.07e-02	**3.27e-12 ± 4.12e-12**	3.85e+00 ± 2.06e-02
0.1	5.1736 ± 1.0199	2.1450 ± 0.3774	2.0480 ± 0.6777	**1.5100 ± 0.6529**	3.8296 ± 0.1018	1.8430 ± 0.5759	3.9292 ± 0.0635
0.2	4.7452 ± 1.5610	2.7204 ± 0.4696	2.0852 ± 0.9160	**1.8561 ± 0.6838**	3.8227 ± 0.1338	2.2368 ± 0.7611	3.9905 ± 0.0682
0.3	5.5354 ± 2.0136	2.9761 ± 0.3983	2.5505 ± 0.7505	**2.3954 ± 0.6578**	3.8131 ± 0.1131	3.0391 ± 1.1436	3.9875 ± 0.0783
0.4	5.3294 ± 0.9577	3.1044 ± 0.3142	2.9872 ± 1.1113	**2.3756 ± 0.4924**	3.8400 ± 0.1307	3.1132 ± 0.6947	3.9987 ± 0.0929
0.5	5.2459 ± 2.3813	3.4180 ± 0.7901	3.2300 ± 0.6382	**2.4854 ± 0.6992**	3.8266 ± 0.1388	3.2250 ± 1.1869	4.0071 ± 0.0750
0.6	4.8255 ± 1.2272	3.3245 ± 0.3178	3.2904 ± 0.6618	**2.5379 ± 0.4120**	3.8810 ± 0.1860	3.6857 ± 1.8163	4.0395 ± 0.1160
0.7	4.7473 ± 0.9303	3.6570 ± 0.3725	3.8645 ± 1.2469	**2.8832 ± 0.7611**	3.8993 ± 0.1735	3.5749 ± 1.7200	4.0699 ± 0.1080
0.8	4.9415 ± 1.0303	3.7081 ± 0.4688	3.1721 ± 0.9695	**2.8147 ± 0.5963**	3.9245 ± 0.1543	3.6217 ± 1.1938	4.0812 ± 0.1168
0.9	5.2865 ± 2.2683	3.7702 ± 0.3601	**3.1507 ± 0.6904**	3.2739 ± 0.7580	3.9278 ± 0.1598	3.4786 ± 1.3672	4.0119 ± 0.0624
1.0	5.1577 ± 2.5388	3.8649 ± 0.4667	3.3031 ± 0.5032	3.1639 ± 0.7150	3.9376 ± 0.1846	**3.1424 ± 0.8280**	4.0310 ± 0.0847
Avg over σ	5.0723 ± 1.6380	2.9717 ± 0.3941	2.6983 ± 0.7423	**2.2997 ± 0.5844**	3.8639 ± 0.1397	2.8146 ± 1.0261	3.9999 ± 0.0806

Table 12. Average over 2000 random scenes using the 6-point epipolar configuration. Lower is better. Metric denotes the sum of relative intrinsic errors $E_{\text{sum-mpe}}$. The smallest entry in each row is highlighted in bold.

Dataset	**Scene**	**Kruppa**	**Kruppa-BnB**	**Global-Best**	**Local Min**	**Local Max**
COLMAP-Public	Gerrard Hall	2.71	**2.10**	2.20	1.51×10^{0}	37675178.94
COLMAP-Public	South Building	**2.43**	5.34	2.96	9.80×10^{-1}	19438204.94
SfM-eval	Castle P19	3.70	5.10	**3.26**	1.63×10^{0}	724875.72
SfM-eval	Entry P10	3.72	5.11	**2.58**	1.82×10^{0}	101349425.08
SfM-eval	Fountain P11	**4.86**	5.49	5.25	3.93×10^{0}	38308763.72
SfM-eval	Herz Jesus P8	9.74	5.38	**5.35**	2.73×10^{0}	50680748.79
Strecha08	Brussel	2.62	8.99	**2.29**	9.24×10^{-1}	1283060.70
Strecha08	Rathaus	4.56	3.34	**2.91**	2.30×10^{0}	33582031.23
Strecha08	Semper	3.47	5.00	**2.25**	1.71×10^{0}	37446217.15
Average		4.20	5.09	**3.23**	—	—

Table 13. Average over 2000 random scenes using the 6-point epipolar configuration. Lower is better. Metrics: Δfg, Δuv, Δs. Per-row minima are bold.

Dataset	**Scene**	**Kruppa**			**Kruppa-BnB**			**Global-Best**		
		Δfg	Δuv	Δs	Δfg	Δuv	Δs	Δfg	Δuv	Δs
COLMAP-Public	Gerrard Hall	0.477	0.720	0.3132	0.714	**0.214**	**0.2412**	**0.414**	0.517	0.3410
COLMAP-Public	South Building	**0.614**	**0.511**	**0.1817**	1.688	0.878	0.2059	0.699	0.654	0.2536
SfM-eval	Castle P19	0.761	0.782	0.6151	1.391	1.027	**0.2630**	**0.677**	**0.759**	0.3922
SfM-eval	Entry P10	0.508	1.075	0.5556	1.410	1.022	**0.2480**	**0.375**	**0.739**	0.3485
SfM-eval	Fountain P11	0.649	1.636	0.2918	1.616	**1.018**	**0.2207**	**0.585**	1.847	0.3896
SfM-eval	Herz Jesus P8	1.669	2.878	0.6424	1.482	**1.059**	**0.2989**	**0.699**	1.783	0.3899
Strecha08	Brussel	0.576	0.601	**0.2697**	2.271	1.984	0.4771	**0.536**	**0.458**	0.3004
Strecha08	Rathaus	0.689	1.623	0.0605	0.845	**0.835**	**0.0158**	**0.503**	0.976	0.0487
Strecha08	Semper	0.537	0.900	0.5978	1.322	1.044	**0.2678**	**0.446**	**0.490**	0.3739
Average		0.720	1.192	0.3920	1.415	1.009	**0.2487**	**0.548**	**0.914**	0.3153

Table 14. Intrinsic calibration errors Δfg, Δuv, and mean reprojection error E_{rep} across scenes and methods. SR denotes success rate over 10 random initializations (best seed kept). Per-scene minima, excluding the oracle method, are shown in bold.

Scene	**Method**	Δfg	Δuv	E_{rep}	**SR**
Castle P19	COLMAP$_{\text{guess}}$	3.901e^{-01}	6.633e^{-01}	1.162e^{+00}	1.00
	COLMAP$_{\text{guess}}$+**K**-BA	2.244e^{-01}	6.678e^{-01}	$\mathbf{4.414}\text{e}^{\mathbf{-01}}$	1.00
	Kruppa-BnB+**K**-BA	—	—	—	0.00
	Kruppa+**K**-BA	4.093e^{-01}	3.393e^{-01}	7.657e^{-01}	0.20
	Global-Best+**K**-BA	$\mathbf{7.961}\text{e}^{\mathbf{-04}}$	$\mathbf{1.887}\text{e}^{\mathbf{-03}}$	4.537e^{-01}	0.10
	COLMAP$_{\text{gt}}$+**K**-BA	2.797e^{-03}	4.477e^{-03}	4.399e^{-01}	1.00
Entry P10	COLMAP$_{\text{guess}}$	3.901e^{-01}	6.633e^{-01}	1.215e^{+00}	1.00
	COLMAP$_{\text{guess}}$+**K**-BA	2.258e^{-01}	6.683e^{-01}	4.827e^{-01}	1.00
	Kruppa-BnB+**K**-BA	$\mathbf{1.330}\text{e}^{\mathbf{-03}}$	$\mathbf{2.990}\text{e}^{\mathbf{-03}}$	$\mathbf{4.826}\text{e}^{\mathbf{-01}}$	0.10
	Kruppa+**K**-BA	6.422e^{-02}	1.544e^{-01}	4.829e^{-01}	0.50
	Global-Best+**K**-BA	1.227e^{-01}	2.659e^{-01}	4.827e^{-01}	0.60
	COLMAP$_{\text{gt}}$+**K**-BA	1.053e^{-03}	2.757e^{-03}	4.827e^{-01}	1.00
Fountain P11	COLMAP$_{\text{guess}}$	3.901e^{-01}	6.633e^{-01}	1.299e^{+00}	1.00
	COLMAP$_{\text{guess}}$+**K**-BA	2.255e^{-01}	6.696e^{-01}	4.335e^{-01}	1.00
	Kruppa-BnB+**K**-BA	—	—	—	0.00
	Kruppa+**K**-BA	$\mathbf{2.441}\text{e}^{\mathbf{-03}}$	$\mathbf{1.747}\text{e}^{\mathbf{-03}}$	4.330e^{-01}	0.10
	Global-Best+**K**-BA	2.530e^{-03}	1.910e^{-03}	$\mathbf{4.328}\text{e}^{\mathbf{-01}}$	0.10
	COLMAP$_{\text{gt}}$+**K**-BA	2.453e^{-03}	1.820e^{-03}	4.334e^{-01}	1.00
Herz Jesus P8	COLMAP$_{\text{guess}}$	3.901e^{-01}	$\mathbf{6.633}\text{e}^{\mathbf{-01}}$	1.132e^{+00}	1.00
	COLMAP$_{\text{guess}}$+**K**-BA	$\mathbf{2.250}\text{e}^{\mathbf{-01}}$	6.697e^{-01}	$\mathbf{4.560}\text{e}^{\mathbf{-01}}$	1.00
	Kruppa-BnB+**K**-BA	—	—	—	0.00
	Kruppa+**K**-BA	—	—	—	0.00
	Global-Best+**K**-BA	—	—	—	0.00
	COLMAP$_{\text{gt}}$+**K**-BA	7.709e^{-04}	1.544e^{-03}	4.569e^{-01}	1.00

References

[1] Daniel Barath, Tekla Toth, and Levente Hajder. A minimal solution for two-view focal-length estimation using two affine correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6003–6011, 2017. 11

[2] Daniel J Bates, Paul Breiding, Tianran Chen, Jonathan D Hauenstein, Anton Leykin, and Frank Sottile. Numerical nonlinear algebra. *arXiv preprint arXiv:2302.08585*, 2023. 2

[3] 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, 10

[4] Martin Byröd, Klas Josephson, and Kalle Åström. A Column-Pivoting Based Strategy for Monomial Ordering in Numerical Gröbner Basis Calculations. In *Computer Vision - ECCV 2008, Marseille, France, October 12-18, 2008, Proceedings, Part IV*, pages 130–143. Springer, 2008. 1

[5] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pages 220–226. IEEE, 2005. 2

[6] Andrea Porfiri Dal Cin, Timothy Duff, Luca Magri, and Tomas Pajdla. Minimal perspective autocalibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5064–5073, 2024. 1, 2, 4, 7, 9, 10, 11, 13, 16, 17

[7] David Cox, John Little, Donal O'shea, and Moss Sweedler. *Ideals, varieties, and algorithms*. Springer, 1997. 10

[8] 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*, 39(3):1421–1446, 2019. 4, 11, 14

[9] Timothy Duff, Kathlen Kohn, Anton Leykin, and Tomas Pajdla. Plmp-point-line minimal problems in complete multi-view visibility. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1675–1684, 2019. 1, 3, 4

[10] Timothy Duff, Viktor Korotynskiy, Anton Leykin, and Tomas Pajdla. Multiprojective geometry of compatible triples of fundamental and essential matrices. *arXiv e-prints*, pages arXiv–2602, 2026. 4

[11] Hongyi Fan, Joe Kileel, and Benjamin Kimia. On the instability of relative pose estimation and ransac's role. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8935–8943, 2022. 6

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[13] Guillermo Gallego, Elias Mueggler, and Peter Sturm. Translation of" zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung" by erwin kruppa (1913). *arXiv preprint arXiv:1801.01454*, 2017. 2

[14] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www2.macaulay2.com. 4

[15] Richard Hartley and Andrew Zisserman. Multipleview geometry in computer vision second edition, 2004. 1, 2, 3, 9

[16] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 10

[17] Richard I. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on pattern analysis and machine intelligence*, 19(2):133–135, 1997. 2, 9

[18] Janne Heikkila. Using sparse elimination for solving minimal problems in computer vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 76–84, 2017. 11

[19] Anders Heyden and Kalle Åkström. Minimal conditions on intrinsic parameters for euclidean reconstruction. In *Asian Conference on Computer Vision*, pages 169–176. Springer, 1998. 11

[20] Joe Kileel and Kathlén Kohn. Snapshot of algebraic vision. *arXiv preprint arXiv:2210.11443*, 2022. 1

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

[22] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla. Beyond Gröbner Bases: Basis Selection for Minimal Solvers. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3945–3954, 2018. 1

[23] Cheng Lei, Fuchao Wu, Zhanyi Hu, and Hung-Tat Tsui. A new approach to solving kruppa equations for camera self-calibration. In *2002 International conference on pattern recognition*, pages 308–311. IEEE, 2002. 2

[24] Q-T Luong and Olivier D Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of computer vision*, 22 (3):261–289, 1997. 2, 6, 9, 15

[25] Evgeniy Martyushev. A minimal six-point auto-calibration algorithm. *arXiv preprint arXiv:1307.3759*, 2013. 2

[26] Pedro Miraldo, Tiago Dias, and Srikumar Ramalingam. A Minimal Closed-Form Solution for Multi-perspective Pose Estimation using Points and Lines. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, pages 490–507, 2018. 1

[27] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. 1

[28] 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. 2, 6, 15

[29] Frederik Schaffalitzky, Andrew Zisserman, Richard I Hartley, and Philip HS Torr. A six point solution for structure and motion. In *European conference on computer vision*, pages 632–648. Springer, 2000. 10

[30] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE con-*

ference on computer vision and pattern recognition, pages 4104–4113, 2016. 7, 16

[31] Johannes L. Schönberger. Colmap dataset collection. https://demuc.de/colmap/datasets/, 2025. Accessed: 8 Nov. 2025. 7

[32] Andrew J Sommese, Jan Verschelde, and Charles W Wampler. Numerical irreducible decomposition using phcpack. In *Algebra, Geometry and Software Systems*, pages 109–129. Springer, 2003. 2

[33] Andrew J Sommese, Charles W Wampler, et al. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005. 6, 10

[34] Christoph Strecha. Multi-view stereo dataset – cvlab, epfl. https://www.epfl.ch/labs/cvlab/data/data-strechamvs/, 2004. Accessed: 8 Nov. 2025. 7

[35] 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. 7

[36] Peter Sturm. On focal length calibration from two views. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, pages II–II. IEEE, 2001. 11

[37] 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. 2, 7

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