A Framework for Reducing the Complexity of Geometric Vision Problems and its Application to Two-View Triangulation with Approximation Bounds

Felix Rydell
Swedish Defence Research Agency

Georg Bökman
University of Amsterdam

Fredrik Kahl
Chalmers University of Technology

Kathlén Kohn
KTH Royal Institute of Technology

Abstract

In this paper, we present a new framework for reducing the computational complexity of geometric vision problems through targeted reweighting of the cost functions used to minimize reprojection errors. Triangulation - the task of estimating a 3D point from noisy 2D projections across multiple images - is a fundamental problem in multiview geometry and Structure-from-Motion (SfM) pipelines. We apply our framework to the two-view case and demonstrate that optimal triangulation, which requires solving a univariate polynomial of degree six, can be simplified through cost function reweighting reducing the polynomial degree to two. This reweighting yields a closed-form solution while preserving strong geometric accuracy. We derive optimal weighting strategies, establish theoretical bounds on the approximation error, and provide experimental results on real data demonstrating the effectiveness of the proposed approach compared to standard methods. Although this work focuses on two-view triangulation, the framework generalizes to other geometric vision problems.

1. Introduction

Multiple view geometry has long been a well-established field within computer vision, with several decades of extensive research, as noted in works such as [9]. The inherent complexity of various subproblems, including relative pose estimation and triangulation, has been rigorously analyzed, often quantified by the number of critical points required to achieve optimal solutions. Typically, these problems are addressed either through slow but guaranteed methods or through faster, local iterative methods that lack assurance of reaching an optimal solution. In contrast, in this paper, a fundamentally different direction is pursued. We introduce a framework that makes targeted modifications to the cost function in order to reduce the underlying complexity. When applied to the triangulation problem, we demonstrate that it can be simplified at its core, reducing the number of critical points and thereby enhancing computational efficiency. An example is given in Figure 1.

Triangulation stands as a cornerstone problem with extensive applications in 3D reconstruction, robotics, and augmented reality. Formally, the task involves recovering the 3D position of a point X from its observed projections $\tilde{\boldsymbol{x}}_i$ in two or more camera images, where each projection is expressed by $\pi_i(X)$. Ideally, if the 3D point X and its corresponding 2D projections $\tilde{\boldsymbol{x}}_i$ are perfectly aligned, that is, $\tilde{\boldsymbol{x}}_i = \pi_i(X)$, then this reconstruction becomes a straightforward calculation. However, in real-world scenarios, various sources of error – such as inaccuracies in the camera's internal parameters, small discrepancies in relative camera positioning, or limitations in point-matching precision – lead to imperfect data. These imperfections result in skewed rays that fail to intersect precisely in 3D space. Consequently, the triangulation problem in practical settings shifts to finding the 3D point that most closely aligns with the observed 2D projections.

Assuming independent Gaussian noise on the image measurements, the maximum likelihood estimate is obtained by minimizing the L_2-error between observed and ideal projections. This geometric error, addressed by Hartley and Sturm already in 1997 [10], involves computing the six critical points intrinsic to the problem [24], meaning that any simpler (non-direct) solution inevitably involves trade-offs. Iterative approaches, such as the fast method proposed by Lindstrom [21], have also been suggested, although these can converge to local optima. An alternative is to change the optimization criterion, for example, using the L_∞-norm [12], which enables an optimal solution through convex optimization. In contrast, we explore a novel approach in multiple view geometry by weighting the L_2-cost function to reduce the set of critical points, allowing direct computation of the solution in closed form.

Our framework can, in principle, be applied to any al-

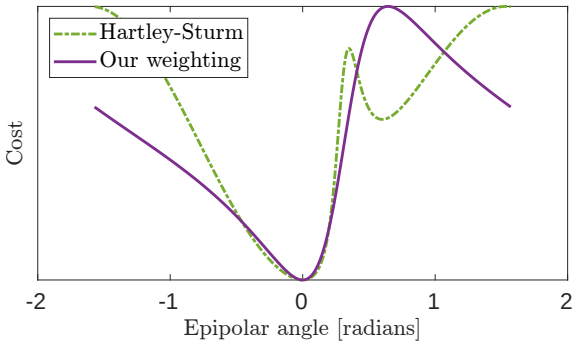

Figure 1. The optimal triangulation problem can have up to three local minima and requires finding the roots of a degree-6 univariate polynomial, following Hartley-Sturm [10]. We propose a weighting of the cost function that lowers the degree to 2, yielding a unique minimum. The plot shows a specific example where Hartley-Sturm's method has to choose between two local minima. Here, we plot the lowest cost value per epipolar plane, parameterised by an angle from the principal axis of one camera (see [10, Section 4.2] for details). The costs are rescaled to have the same minimum and maximum value for ease of viewing.

gebraic optimization problem and is thus widely applicable within multiple view geometry, e.g., to n-view triangulation [11], camera pose estimation [7] and registration [25]. Here, we showcase our strategy in detail for two-view triangulation.

In Sec. 2, we introduce our overall approach formally. In Sec. 3, we recap two-view triangulation and provide an equivalent reformulation via a diagonalizing change of coordinates. In Sec. 4, we study our proposed weighted version of two-view triangulation. We determine the best weights that reduce the number of critical points from 6 to either 4 or 2 (see Theorems 4.1 and 4.3) and find theoretical bounds on the quality of our proposed approximation (see Proposition 4.4). Experiment results and comparisons to baselines on real data are given in Sec. 5.

1.1. Related Work

Triangulation. The most well-known approach to triangulation is to solve for the six critical points as described in [10], but the algorithm tends to be slow. Computing the midpoint between 3D rays does not work well for near parallel rays, hence it should in general be avoided. Using algebraic cost functions, such as the Direct Linear Transform, can be fast but may be inaccurate [9]. Iterative methods for triangulation were pioneered by Kanatani et. al. [14, 15] and the method by Lindstrom [21] leverages this approach. In theory, the method does not guarantee an optimal solution with respect to the cost function, whereas our approach does with respect to the weighted cost function. Other works on triangulation modify the optimize criterion, for instance, the L_1-cost and the L_∞-cost functions are optimized in [12, 13, 19]. In [18], a variant of the midpoint method is proposed. We experimentally compare to both Lindstrom [21] and Hartley-Sturm [10], two natural baselines that solve the optimal triangulation problem — the former being a fast, iterative method, and the latter a slower, exact method.

The Geometric Error. In applied algebraic geometry, fitting noisy data points to a mathematical model defined by polynomials has seen a lot of interest [3]. The smallest distance between the data and the model is the *geometric error*. The corresponding geometric error for homographies was first introduced by Sturm [32]. The *Euclidean distance degree* is the number of smooth complex critical points for the optimization problem, given random data. It expresses the algebraic complexity of fitting data to a model; the higher the Euclidean distance degree, the more computationally expensive the optimization. For 3D reconstruction, several works have studied or computed these degrees, e.g., [6, 10, 23, 24, 26]. The Euclidean distance degree is used to implement efficient solvers in homotopy continuation [2] or to solve the associated polynomial systems via specialized symbolic solvers [17].

The Sampson Error. Sampson approximation was first proposed in [28] and independently by Taubin [33] to approximate the point-conic distance. Luong and Faugeras [22] introduced it to approximate the reprojection error in epipolar geometry. The Sampson error has been considered in other vision settings [4, 5, 20, 34]. This extensive use of Sampson approximation for geometric problems shows its versatility. Recently, the Sampson error was revisited and studied from a mathematical perspective [27].

Weighted Euclidean Problems. The authors of [16] studied weighted Euclidean distance problems for rank-one approximations of tensors, variations thereof, and quadric hypersurfaces. Similarly to this article, they analyze the weights that lead to the smallest number of critical points.

2. Framework

We consider geometric vision problems involving 3D points, their image projections, and the cameras. A full-rank 3×4 matrix C defines a camera that projects a 3D point $X \in \mathbb{R}^3$ as

$$\begin{aligned} \pi : \mathbb{R}^3 &\dashrightarrow \mathbb{R}^2 \\ X &\mapsto \begin{bmatrix} C(X;1)_1 / C(X;1)_3 \\ C(X;1)_2 / C(X;1)_3 \end{bmatrix}. \end{aligned} \tag{1}$$

Here, $\dashrightarrow$ denotes a *rational map*, meaning a map well-defined almost everywhere. We study both uncalibrated cameras, where C is unconstrained, and calibrated cameras of the form $C = [R; t]$, where $R \in \mathrm{SO}(3)$ and $t \in \mathbb{R}^3$.

Now, consider the residual between a projected 3D point X and its measured image point $\tilde{x}$, given by $\epsilon = \tilde{x} - \pi(X)$. If the measured image points are corrupted by independent, normally distributed noise, the maximum likelihood estimate is obtained by minimizing $||\boldsymbol{\epsilon}||^2$ over the unknowns, where $\boldsymbol{\epsilon}$ is the vector of all image residuals. The unknowns, depending on the task, may consist of 3D points and/or cameras. In compact form, we seek to solve problems of the form

$$\begin{aligned} \min_{\boldsymbol{\epsilon},\boldsymbol{z}} \quad & ||\boldsymbol{\epsilon}||^2 \\ \text{s.t.} \quad & p(\boldsymbol{\epsilon},\boldsymbol{z}) = 0, \end{aligned} \tag{2}$$

where $\boldsymbol{z}$ encodes the unknown parameters of interest. The constraint vector $p(\boldsymbol{\epsilon},\boldsymbol{z}) = 0$ can be written as polynomial constraints by clearing denominators. Solving this optimization problem exactly quickly becomes intractable for large problems. One measure of complexity is the number of smooth complex critical points of the optimization given generic measured image points $\tilde{\boldsymbol{x}}$, known as the *Euclidean distance degree* (ED-degree) of the problem.

Example 2.1 (Triangulation)**.** Given cameras C_i for $i = 1,\dots,n$ and corresponding image points $\tilde{x}_i$ in n views, computing the 3D point X is known as *triangulation*. The ED-degree for $n = 2$ is well known to be 6 (for generic cameras), with an algorithm for computing the six stationary points first presented in [10]. For three-view triangulation ($n = 3$), the ED-degree is 47 [23, 31].

The ED-degree of a geometric vision problem is intrinsic, meaning that reducing complexity requires altering the problem itself. We address this by reweighting the objective function and analyzing how different choices of weights affect the ED-degree. Concretely, we investigate

$$\begin{aligned} \min_{\boldsymbol{\varepsilon},\boldsymbol{z}} \quad & \sum_i \lambda_i \varepsilon_i^2 \\ \text{s.t.} \quad & p(\boldsymbol{\varepsilon},\boldsymbol{z}) = 0, \end{aligned} \tag{3}$$

where λ_i are positive weights applied to the residual terms. The optimal choice of weights depends heavily on the constraints p and can be challenging to determine. One strategy to mitigate this is to first make the constraints simpler by applying a coordinate transformation of the form

$$\boldsymbol{\varepsilon} = \boldsymbol{R}\boldsymbol{\epsilon} \tag{4}$$

with an orthogonal matrix $\boldsymbol{R}$. This leaves the problem (2) unchanged, as $||\boldsymbol{\varepsilon}|| = ||\boldsymbol{\epsilon}||$, ensuring that the ED-degree remains the same. We carry out this strategy in detail for two-view triangulation.

3. Two-View Triangulation

A common way of expressing two-view triangulation is via the fundamental matrix $\boldsymbol{F}$ of the camera pair C_1, C_2. More precisely, given a fundamental matrix $\boldsymbol{F}$, i.e., a 3×3 rank-2 matrix, it is the following squared-error minimization:

$$\begin{aligned} \mathcal{E}^2_{\boldsymbol{F}}(\tilde{\boldsymbol{x}}) := \min_{\boldsymbol{\epsilon}} \quad & \|\boldsymbol{\epsilon}\|^2 \\ \text{s.t.} \quad & (\tilde{\boldsymbol{x}}_1 + \boldsymbol{\epsilon}_1; 1)^\intercal \boldsymbol{F} (\tilde{\boldsymbol{x}}_2 + \boldsymbol{\epsilon}_2; 1) = 0. \end{aligned} \tag{5}$$

Our goal is to find an approximate solution to (5) that is simpler and faster to compute via (3) and (4). In this direction, we first simplify the epipolar constraint. Note that

$$(\boldsymbol{x}_1; 1)^\top \boldsymbol{F} (\boldsymbol{x}_2; 1) \tag{6}$$

equals

$$(\boldsymbol{x}; 1)^\top \underbrace{\frac{1}{2}\begin{bmatrix} 0 & F_{2\times 2} & F_h \\ F_{2\times 2}^\top & 0 & F_v^\top \\ F_h^\top & F_v & 2F_{3,3} \end{bmatrix}}_{\boldsymbol{Q}(\boldsymbol{F}) :=} (\boldsymbol{x}; 1), \tag{7}$$

where

$$\boldsymbol{F} = \begin{bmatrix} F_{2\times 2} & F_h \\ F_v & F_{3,3} \end{bmatrix}. \tag{8}$$

Lemma 3.1. *For a fundamental matrix $\boldsymbol{F}$, the matrix $\boldsymbol{Q}(\boldsymbol{F})$ is rank-deficient. Moreover, if $F_{2\times 2}$ is invertible, then $\boldsymbol{Q}(\boldsymbol{F})$ has rank 4 and its kernel is*

$$\begin{bmatrix} k(\boldsymbol{F}) \\ 1 \end{bmatrix}, \quad \text{where} \quad k(\boldsymbol{F}) := \begin{bmatrix} -F_{2\times 2}^{-\top} F_v^\top \\ -F_{2\times 2}^{-1} F_h \end{bmatrix}. \tag{9}$$

Proof. The determinant of $\boldsymbol{Q}(\boldsymbol{F})$ is $\det(F_{2\times 2})\det(\boldsymbol{F})/16$. Therefore, it is always rank-deficient. Moreover, since $F_{2\times 2}$ is invertible, the top left 4×4 matrix of $\boldsymbol{Q}(\boldsymbol{F})$ has rank 4, implying that $\boldsymbol{Q}(\boldsymbol{F})$ has rank 4. □

Now we can rewrite (7) further. Denote by $\boldsymbol{P}(\boldsymbol{F})$ the upper left 4×4 matrix of $\boldsymbol{Q}(\boldsymbol{F})$. By construction,

$$\begin{aligned} &(\boldsymbol{x}; 1)^\top \boldsymbol{Q}(\boldsymbol{F}) (\boldsymbol{x}; 1) = \\ &\qquad (\boldsymbol{x} - k(\boldsymbol{F}))^\top \boldsymbol{P}(\boldsymbol{F}) (\boldsymbol{x} - k(\boldsymbol{F})). \end{aligned} \tag{10}$$

Next, we express this constraint in terms of the eigenvalues of $\boldsymbol{P}(\boldsymbol{F})$. Since that matrix is symmetric, its eigenvalues are real. In fact, they are the signed singular values of $F_{2\times 2}$. Hence, the eigenvalues of $\boldsymbol{P}(\boldsymbol{F})$ are $a_1, -a_1, a_2, -a_2$ for some $a_1, a_2 \geq 0$. Up to translation and orthogonal action, we now see that (10) is equal to

$$\sum_i q_i \boldsymbol{y}_i^2, \quad \text{where } \boldsymbol{q} = (a_1, -a_1, a_2, -a_2). \tag{11}$$

Here, the new variables $\boldsymbol{y}$ are obtained from $\boldsymbol{x}$ via $\boldsymbol{y} = \boldsymbol{R}(\boldsymbol{F})^\top (\boldsymbol{x} - k(\boldsymbol{F}))$ for some orthogonal 4×4 matrix $\boldsymbol{R}(\boldsymbol{F})$. Our updated optimization problem is then

$$\begin{aligned} \mathcal{E}^2_{\boldsymbol{q}}(\tilde{\boldsymbol{y}}) := \min_{\boldsymbol{\varepsilon}} \quad & \|\boldsymbol{\varepsilon}\|^2 \\ \text{s.t.} \quad & \sum_i q_i (\tilde{y}_i + \varepsilon_i)^2 = 0. \end{aligned} \tag{12}$$

Proposition 3.2. *Let $\boldsymbol{F}$ be a fundamental matrix such that $F_{2\times 2}$ is invertible, and let $\operatorname{diag}(\boldsymbol{q}) = \boldsymbol{R}^\top \boldsymbol{P}(\boldsymbol{F})\boldsymbol{R}$ be a diagonalization of $\boldsymbol{P}(\boldsymbol{F})$. Then the critical points of (5) with data $\tilde{\boldsymbol{x}}$ are in bijection with the critical points of (12) with data $\tilde{\boldsymbol{y}} = \boldsymbol{R}^\top(\tilde{\boldsymbol{x}} - k(\boldsymbol{F}))$ via*

$$\boldsymbol{\epsilon} \mapsto \boldsymbol{\varepsilon} = \boldsymbol{R}^\top \boldsymbol{\epsilon}. \tag{13}$$

In particular, $\mathcal{E}_{\boldsymbol{F}}(\tilde{\boldsymbol{x}}) = \mathcal{E}_{\boldsymbol{q}}(\tilde{\boldsymbol{y}})$.

Proof. It is a well-known fact in metric algebraic geometry that translation and orthogonal transformation preserve the ED-degree and that the critical points are in bijection via (13). To see this, one can directly study the critical equations. The details are worked out in [26]. □

Remark 3.3. *All parameters $a_1, a_2 > 0$ are possible, also when we restrict ourselves to calibrated cameras. This is because all non-zero matrices $F_{2\times 2}$ can be obtained from $C_1 = [I\ 0]$ and $C_2 = [R\ t]$ with $R \in \mathrm{SO}(3)$. Indeed, one can choose $t_1 = t_2 = 0$ and add a column to $\left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right] F_{2\times 2}^\top$ such that the resulting 2×3 matrix S has orthogonal rows of the same norm. Then choose t_3 such that S/t_3 has rows of norm 1 and extend that matrix to a 3×3 rotation matrix R. That way, the top left block of $R^\top [t]_\times$ is $F_{2\times 2}$.*

4. The weighted optimization problem

We will now show that one can change the standard squared-error minimization to a *weighted* squared-error loss such that the number of critical points drops and the optimization problem becomes simpler. More concretely, we replace (12) by

$$\begin{aligned} \mathcal{E}^2_{\boldsymbol{q},\boldsymbol{\lambda}}(\tilde{\boldsymbol{y}}) := \min_{\boldsymbol{\varepsilon}} \quad & \sum_i \lambda_i \varepsilon_i^2 \\ \text{s.t.} \quad & \sum_i q_i(\tilde{y}_i + \varepsilon_i)^2 = 0, \end{aligned} \tag{14}$$

where $\boldsymbol{q} = (a_1, -a_1, a_2, -a_2)$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3, \lambda_3) \in \mathbb{R}^4_{>0}$. The restriction that the λ_i are positive ensures that the optimization problem corresponds to minimizing a distance (that may differ from the standard Euclidean distance). The number of complex critical points of (14) for general a_1, a_2 and $\tilde{\boldsymbol{y}}$ is called the $\boldsymbol{\lambda}$-*Weighted ED-degree* ($\boldsymbol{\lambda}$-degree).

Theorem 4.1. *Let $a_i, \lambda_i > 0$. The $\boldsymbol{\lambda}$-degree of (14) is*

I. 2 if $\lambda = (\mu a_1, \nu a_1, \mu a_2, \nu a_2)$ for some $\mu, \nu \in \mathbb{R}_{>0}$,
II. 4 otherwise if $(\lambda_1, \lambda_3) = \mu(a_1, a_2)$ for some $\mu \in \mathbb{R}_{>0}$ or $(\lambda_2, \lambda_4) = \nu(a_1, a_2)$ for some $\nu \in \mathbb{R}_{>0}$,
III. 6 otherwise.

As a consequence, if $a_1 = a_2$, then the $\boldsymbol{\lambda}$-degree is 2 for $\boldsymbol{\lambda} = (1, 1, 1, 1)$, meaning that the ED-degree is also 2. This is in particular the case for calibrated cameras $C_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $C_2 = \begin{bmatrix} R & t \end{bmatrix}$ where the last row of R is $(0, 0, 1)$, as we will see in Prop. 4.5. Note that the last row of R encodes the optical axis of the camera, and hence for a stereo rig with parallel image planes, we obtain the optimal (unweighted) L_2-solution without having to solve a degree-6 polynomial!

Proof. Given noisy measurements $\tilde{\boldsymbol{y}}$, the critical points of the optimization problem (14) are those $\boldsymbol{\varepsilon} \neq -\tilde{\boldsymbol{y}}$ that satisfy the problem's constraint and such that the Jacobian matrix

$$\begin{bmatrix} \lambda_1\varepsilon_1 & \lambda_2\varepsilon_2 & \lambda_3\varepsilon_3 & \lambda_4\varepsilon_4 \\ a_1(\tilde{y}_1 + \varepsilon_1) & -a_1(\tilde{y}_2 + \varepsilon_2) & a_2(\tilde{y}_3 + \varepsilon_3) & -a_2(\tilde{y}_4 + \varepsilon_4) \end{bmatrix}$$

has rank one. Writing $q = (a_1, -a_1, a_2, -a_2)$, the rank constraint means that $\lambda_i\varepsilon_i = sq_i(\tilde{y}_i + \varepsilon_i)$ for some scalar s and all $i = 1, 2, 3, 4$. This allows us to express

$$\varepsilon_i = sq_i\tilde{y}_i/(\lambda_i - sq_i). \tag{15}$$

Plugging the latter into the constraint of (14) yields a rational function

$$\mathcal{R} = \frac{r_6(s)}{\prod_i(\lambda_i - sq_i)^2}, \tag{16}$$

whose numerator $r_6(s)$ depends on $a_i, \tilde{y}_i$ and is of degree 6 in s. It has too many terms to be displayed here, but the `Macaulay2` [8] code in the SM computes it explicitly. The roots of the numerator correspond to the critical points of the optimization (14), showing that the $\boldsymbol{\lambda}$-degree is at most 6 for generic weights $\boldsymbol{\lambda}$. This is in fact an equality since the standard Euclidean distance problem (6) has ED-degree 6.

A priori, there are two ways how the $\boldsymbol{\lambda}$-degree (i.e., the degree of the numerator above) can drop. For special choices of weights $\boldsymbol{\lambda}$, either the leading coefficient of $r_6(s)$ can vanish, or $r_6(s)$ can share a common factor with the denominator of $\mathcal{R}$. The first case cannot happen, as the leading coefficient is

$$a_1^3a_2^3(a_2\lambda_1^2\tilde{y}_1^2 - a_2\lambda_2^3\tilde{y}_2^2 + a_1\lambda_3^2\tilde{y}_3^2 - a_1\lambda_4^2\tilde{y}_4^2), \tag{17}$$

which is non-zero for generic $\tilde{y}$.

Next we analyze under which conditions one of the factors $(\lambda_i - sq_i)$ of the denominator divides the numerator. This is equivalent to that $s = \lambda_i/q_i$ is a root of the numerator. Plugging $s = \lambda_1/q_1$ into the numerator yields

$$a_1^{-3}\,\lambda_1^2\,(\lambda_1 + \lambda_2)^2\,\tilde{y}_1^2\,(a_1\lambda_4 + a_2\lambda_1)^2\,(a_1\lambda_3 - a_2\lambda_1)^2. \tag{18}$$

Due to $a_i, \lambda_j > 0$, only the last factor in this expression can be zero (for generic $\tilde{y}$). That term being zero means that $(\lambda_1, \lambda_3) = \mu(a_1, a_2)$ for some $\mu \in \mathbb{R}_{>0}$. Hence, we have shown that the latter condition is equivalent to $(\lambda_1 - sq_1)$ dividing the numerator.

Analogously, we obtain that $(\lambda_3 - sq_3)$ divides the numerator if and only if $(\lambda_1, \lambda_3) = \mu(a_1, a_2)$ for some $\mu \in$

$\mathbb{R}_{>0}$; and that $(\lambda_2 - sq_2)$ or $(\lambda_4 - sq_4)$ divide the numerator if and only if $(\lambda_2, \lambda_4) = \nu(a_1, a_2)$ for some $\nu \in \mathbb{R}_{>0}$.

Without loss of generality, we now assume that $(\lambda_1, \lambda_3) = \mu(a_1, a_2)$ for some $\mu \in \mathbb{R}_{>0}$. Then

$$\mathcal{R} = \frac{r_4(s)}{(\lambda_2 - sq_2)^2(\lambda_4 - sq_4)^2(\mu - s)^2}, \tag{19}$$

whose numerator $r_4(s)$ depends on $a_i, \tilde{y}_i$ and is of degree 4 in s. The code in the SM produces also this numerator. Therefore, for generic λ_2 and λ_4, the $\boldsymbol{\lambda}$-degree is now 4. As before, the leading coefficient of the numerator of $\mathcal{R}$ does not vanish for generic $\tilde{y}$. Thus, the degree can only drop further if one of the factors in the denominator of $\mathcal{R}$ divides the numerator. The factor $(\mu - s)$ cannot divide the numerator for generic $\tilde{y}$, since plugging $s = \mu$ into the numerator yields

$$\mu^2 (a_1\mu + \lambda_2)^2 (a_2\mu + \lambda_4)^2 (a_1\tilde{y}_1^2 + a_2\tilde{y}_3^2), \tag{20}$$

which is positive for generic $\tilde{y}$. Hence, the degree can only drop further if $(\lambda_2 - sq_2)$ or $(\lambda_4 - sq_4)$ divide the numerator. We have already shown above that this is equivalent to $(\lambda_2, \lambda_4) = \nu(a_1, a_2)$ for some $\nu \in \mathbb{R}_{>0}$. In this case,

$$\mathcal{R} = \frac{r_2(s)}{(\mu - s)^2(\nu + s)^2}, \tag{21}$$

whose numerator $r_2(s)$ depends on $a_i, \tilde{y}_i$ and is of degree 2 in s. (Its coefficients are explicit stated in (22); see proof of Lemma 4.2). This shows that the $\boldsymbol{\lambda}$-degree is now 2. □

The weighted squared-error minimizations of $\boldsymbol{\lambda}$-degree 2 and 4 from Theorem 4.1 can be explicitly solved. Therefore, they are significantly faster to solve than the original problem of ED-degree 6. This raises two natural questions:

- Which choice of scalars μ, ν is 'best' in the sense that the solution to (14) best approximates the original problem?
- How good is this 'best' approximation?

More concretely, we say that the best μ, ν are those such that the minimizer $\boldsymbol{\varepsilon}(\mu, \nu)$ of (14) minimizes the standard squared error $\sum_i \varepsilon_i^2$. Since the minimizer $\varepsilon(\mu, \nu)$ is not affected by multipying λ with a global scalar, we assume from now on without loss of generality that $\mu = 1$. (We could of course also assume that $a_1 = 1$, but choose not do so.)

We solve the two questions above for weights of the form $\lambda = (a_1, \nu a_1, a_2, \nu a_2)$. We find the optimal ν by theoretical means in Theorem 4.3 and use this result to bound the error $\mathcal{E}_{\boldsymbol{F}}$ in Proposition 4.4. We then perform experiments on the quality of our proposed approximation in Sec. 5.

We begin by describing the critical points of (14) in terms of

$$\begin{aligned} A &:= a_1\tilde{y}_1^2 - \nu^2 a_1\tilde{y}_2^2 + a_2\tilde{y}_3^2 - \nu^2 a_2\tilde{y}_4^2, \\ B &:= 2\nu(a_1\tilde{y}_1^2 + \nu a_1\tilde{y}_2^2 + a_2\tilde{y}_3^2 + \nu a_2\tilde{y}_4^2), \\ C &:= \nu^2(a_1\tilde{y}_1^2 - a_1\tilde{y}_2^2 + a_2\tilde{y}_3^2 - a_2\tilde{y}_4^2), \end{aligned} \tag{22}$$

and

$$\begin{aligned} \Delta :=& B^2 - 4AC \\ =& 4\nu^2(\nu+1)^2(a_1\tilde{y}_1^2 + a_2\tilde{y}_3^2)(a_1\tilde{y}_2^2 + a_2\tilde{y}_4^2). \end{aligned} \tag{23}$$

Lemma 4.2. *For $\lambda = (a_1, \nu a_1, a_2, \nu a_2)$, the two critical points of (14) are*

$$\varepsilon^{\pm}(\nu) = (s^{\pm} q_i \tilde{y}_i / (\lambda_i - s^{\pm} q_i))_{i=1}^4, \tag{24}$$

where $s^{\pm} = (-B \pm \sqrt{\Delta})/2A$.

Proof. We consider the rational function $\mathcal{R}$ from the proof of Theorem 4.1 whose roots correspond to the critical points of (14). The numerator of $\mathcal{R}$ is $As^2 + Bs + C$. The discriminant Δ of the numerator is positive for generic $\tilde{y}$ and $\nu > 0$. Thus, there are 2 real roots $s^{\pm} = (-B \pm \sqrt{\Delta})/2A$. These yield the critical points $\varepsilon_i^{\pm}(\nu) = s^{\pm} q_i \tilde{y}_i / (\lambda_i - s^{\pm} q_i)$. □

Now we can describe the best ν in terms of

$$S = (\tilde{y}_1^2 + \tilde{y}_3^2)(a_1\tilde{y}_2^2 + a_2\tilde{y}_4^2), \tag{25}$$

$$T = (\tilde{y}_2^2 + \tilde{y}_4^2)(a_1\tilde{y}_1^2 + a_2\tilde{y}_3^2), \tag{26}$$

which we do in the next result. For the proof, we define $\delta := (a_1\tilde{y}_1^2 + a_2\tilde{y}_3^2)(a_1\tilde{y}_2^2 + a_2\tilde{y}_4^2)$, which satisfies $\Delta = 4\nu^2(\nu+1)^2\delta$.

Theorem 4.3. *Let $\lambda = (a_1, \nu a_1, a_2, \nu a_2)$. For every $\nu > 0$, $\boldsymbol{\varepsilon}^+(\nu)$ minimizes both the weighted and the nonweighted squared error, i.e., $\sum_i \lambda_i(\boldsymbol{\varepsilon}_i^+(\nu))^2 < \sum_i \lambda_i(\boldsymbol{\varepsilon}_i^-(\nu))^2$ and $\sum_i(\boldsymbol{\varepsilon}_i^+(\nu))^2 < \sum_i(\boldsymbol{\varepsilon}_i^-(\nu))^2$. Further, there is a unique $\nu \in \mathbb{R}_{>0}$ for which $\sum_i(\boldsymbol{\varepsilon}_i^+(\nu))^2$ is minimized. It is*

$$\nu = \frac{T}{S}. \tag{27}$$

Proof. Evaluating the weighted squared-error at the critical points in (24) yields

$$\sum_i \lambda_i(\varepsilon_i^{\pm}(\nu))^2 = \frac{\nu}{\nu+1}\Big(\underbrace{\big(\sum |q_i|\tilde{y}_i^2\big)}_{\alpha^{\pm} :=} - \pm 2\sqrt{\delta}\Big). \tag{28}$$

The code that produces this identity is provided in the SM. The minimizer is therefore $\varepsilon^+(\nu)$. Note that

$$\alpha^{\pm} = \Big(\sqrt{a_1\tilde{y}_1^2 + a_2\tilde{y}_3^2} - \pm\sqrt{a_1\tilde{y}_2^2 + a_2\tilde{y}_4^2}\Big)^2, \tag{29}$$

so that for generic data $\tilde{y}_i$, $\alpha^{\pm} > 0$. Evaluating the nonweighted squared-error at the critical points gives

$$\sum_i (\varepsilon_i^{\pm}(\nu))^2 = \frac{\alpha^{\pm}(S\nu^2 + T)}{\delta(\nu+1)^2}. \tag{30}$$

The code that produces this identity is provided in the SM. Since each $\alpha^{\pm}, S, T, \delta$ are > 0 for generic data, it follows that $\varepsilon^+(\nu)$ is the minimizer of the standard squared error.

Next we compute the best ν. The derivative of (30) for $\pm = +$ with respect to ν is

$$\frac{2\alpha^+(S\nu - T)}{\delta(\nu+1)^3}. \tag{31}$$

Setting this expression to 0, the unique solution T/S is non-negative. One can check that the second derivative at T/S is positive, implying that this choice of ν yields the global minimum. □

Now we provide bounds on the error $\mathcal{E}_{\boldsymbol{F}}$. Recall from Proposition 3.2 that $\mathcal{E}_{\boldsymbol{F}}(\tilde{\boldsymbol{x}}) = \mathcal{E}_{\boldsymbol{q}}(\tilde{\boldsymbol{y}})$, and note that from the proof of Theorem 4.3,

$$\sqrt{\alpha^+} = \left|\sqrt{a_1\tilde{y}_1^2 + a_2\tilde{y}_3^2} - \sqrt{a_1\tilde{y}_2^2 + a_2\tilde{y}_4^2}\right|. \tag{32}$$

Observe that $\alpha^+ = 0$ if and only if $\tilde{\boldsymbol{y}}$ lies on the model, and is strictly greater than 0 otherwise.

Proposition 4.4. *The inequality*

$$\mathcal{E}_{\boldsymbol{q}}(\tilde{\boldsymbol{y}}) \le \sqrt{\frac{\alpha^+}{\delta}\frac{ST}{S+T}} \tag{33}$$

holds, along with the bounds

$$\frac{\sqrt{\alpha^+}}{\sqrt{2\max\{a_1,a_2\}}} \le \mathcal{E}_{\boldsymbol{q}}(\tilde{\boldsymbol{y}}) \le \frac{\sqrt{\alpha^+}}{\sqrt{2\min\{a_1,a_2\}}}. \tag{34}$$

The right-hand side of (33) can be simplified somewhat by noting that

$$ST = (\tilde{y}_1^2 + \tilde{y}_3^2)(\tilde{y}_2^2 + \tilde{y}_4^2)\delta. \tag{35}$$

The ratio between the upper and lower bounds of (34) is

$$\sqrt{\frac{\max\{a_1,a_2\}}{\min\{a_1,a_2\}}}. \tag{36}$$

Therefore, the closer the ratio a_2/a_1 is to 1, the better the these bounds are. In comparison, the upper bound (33) is a better approximation, which follows from the proof. However, this comes at the cost of a more complicated expression.

Proof. Let ε^* be the minimizer for the nonweighted problem. Then the minimizer from Theorem 4.3 gives an upper bound for $\mathcal{E}_{\boldsymbol{q}}^2(\tilde{\boldsymbol{y}}) = \sum(\varepsilon_i^*)^2$. As in the proof of Theorem 4.3, plugging $\nu = T/S$ into (30) for $\pm = +$, we get that the minimum value of (30) is

$$\frac{\alpha^+}{\delta}\frac{ST}{S+T}, \tag{37}$$

which proves the upper bound.

For the other part, we prove the lower bound and note that the upper bound is proven the same way. We observe that

$$\max\{\lambda_i\}\sum(\varepsilon_i^*)^2 \ge \sum\lambda_i(\varepsilon_i^*)^2. \tag{38}$$

Thus, Theorem 4.3 implies $\mathcal{E}_{\boldsymbol{q}}^2(\tilde{\boldsymbol{y}}) \ge \frac{1}{\max\{\lambda_i\}}\sum\lambda_i(\varepsilon_i^+(\nu))^2$ for $\lambda = (a_1, \nu a_1, a_2, \nu a_2)$ and arbitrary $\nu > 0$. By (28), the latter expression equals

$$\frac{\nu}{\max\{\lambda_i\}(\nu+1)}\alpha^+. \tag{39}$$

Note that $\max\{\lambda_i\} = \max\{1,\nu\}\max\{a_1,a_2\}$. Therefore, (39) reaches its maximum value when $\nu = 1$. □

Finally, we investigate when the case $a_1 = a_2$ happens.

Proposition 4.5. *We have $a_1 = a_2$ if and only if $F_{2\times2}$ is a scalar times an orthogonal matrix.*

This condition is satisfied for calibrated cameras $C_1 = \begin{bmatrix} I & 0\end{bmatrix}$ and $C_2 = R\begin{bmatrix} I & -c\end{bmatrix}$ if and only if the last row of R is $(0,0,\pm1)$ or c is a scalar times $(r_{31}, r_{32}, r_{33} \pm 1)^\top$.

Recall that the last row of R encodes the optical axis of the camera C_2. So that row being $(0,0,\pm1)$ is equivalent to the optical axes of both cameras being parallel, while the latter condition in Proposition 4.5 means that the center of C_2 is proportional to the sum / difference of the optical axes.

Proof. The first statement is clear since a_1, a_2 are the singular values of $F_{2\times2}$. For calibrated cameras $C_1 = \begin{bmatrix} I & 0\end{bmatrix}$ and $C_2 = R\begin{bmatrix} I & -c\end{bmatrix}$, we have $\boldsymbol{F} = R^\top[t]_\times$, where $t = -Rc$ and

$$[t]_\times := \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}. \tag{40}$$

We note that $\boldsymbol{F} = R^\top[t]_\times = R^\top[-Rc]_\times = -[c]_\times R^\top$ due to rotation equivariance of the cross-product. If $R = \left[\begin{smallmatrix} R' & 0 \\ 0 & \pm1\end{smallmatrix}\right]$, then $F_{2\times2} = t_3(R')^\top\left[\begin{smallmatrix} 0 & -1 \\ 1 & 0\end{smallmatrix}\right] \in t_3\mathrm{O}(2)$.

Next, we analyze the case $r_{33} \neq \pm1$ and $(c_1, c_2) = (r_{31}, r_{32})\frac{c_3}{r_{33}\pm1}$. For $\pm = +$, a direct computation reveals that $\boldsymbol{F}$ becomes

$$\frac{c_3}{r_{33}+1}\begin{bmatrix} r_{12}-r_{21} & r_{11}+r_{22} & r_{32} \\ -r_{11}-r_{22} & r_{12}-r_{21} & -r_{31} \\ -r_{23} & r_{13} & 0 \end{bmatrix}. \tag{41}$$

This can also be verified via `Macaulay2` code provided in the SM. In particular, we see that the top left 2×2 block is a scaled $\mathrm{SO}(2)$ matrix. Analogously, for $\pm = -$, we obtain

$$\boldsymbol{F} = \frac{c_3}{r_{33}-1}\begin{bmatrix} -r_{12}-r_{21} & r_{11}-r_{22} & -r_{32} \\ r_{11}-r_{22} & r_{12}+r_{21} & r_{31} \\ -r_{23} & r_{13} & 0 \end{bmatrix} \tag{42}$$

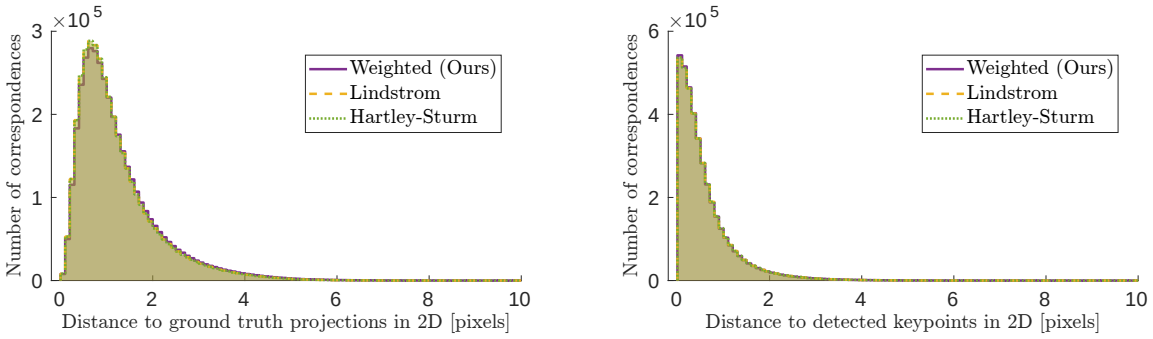

Figure 2. 2D errors over correspondences from randomly sampled image pairs from the Pantheon dataset, when solving the triangulation problem using different methods. The methods perform very similarly, but our weighted method is slightly worse than the others. Left: Distance to ground truth projections; Right: Distance to measured 2D points.

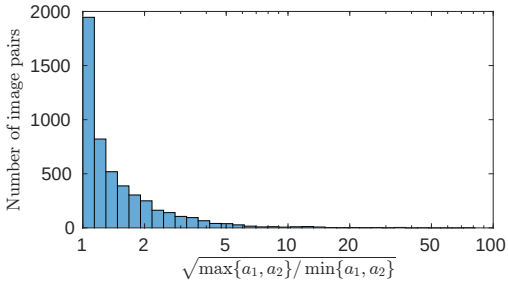

Figure 3. The eigenvalue ratios in our randomly sampled set of 5000 image pairs from the Pantheon dataset.

and so $F_{2\times2}$ is a scalar times an orthogonal matrix (of determinant -1).

For the converse direction, we consider $\boldsymbol{F} = R^\top[t]_\times = -[c]_\times R^\top$ for some rotation matrix R and $t = -Rc$ such that $F_{2\times2}$ is a scaling of an orthogonal matrix. We provide `Macaulay2` code in the SM for solving the resulting equations. Here we show a straightforward calculation of c_1 in the case of $F_{2\times2}$ having positive determinant. (Other cases can be proven similarly.) We have that

$$F_{2\times2} = -\begin{bmatrix} r_{13}c_2 - r_{12}c_3 & r_{23}c_2 - r_{22}c_3 \\ -r_{13}c_1 + r_{11}c_3 & -r_{23}c_1 + r_{21}c_3 \end{bmatrix}, \quad (43)$$

which is a scaled rotation if and only if

$$\begin{aligned} r_{13}c_2 - r_{12}c_3 &= -r_{23}c_1 + r_{21}c_3 \\ r_{23}c_2 - r_{22}c_3 &= r_{13}c_1 - r_{11}c_3. \end{aligned} \quad (44)$$

By multiplying the first equation with r_{23}, the second with r_{13}, subtracting the first from the second equation and collecting terms we obtain

$$(r_{13}^2 + r_{23}^2)c_1 = (r_{23}(r_{21} + r_{12}) + r_{13}(r_{11} - r_{22}))c_3. \quad (45)$$

We can rewrite the left-hand-side as $(1 - r_{33}^2)c_1$ by using that the last column of R has unit norm. Further, since the cross product of the first two rows of R equals the third row we have $r_{23}r_{12} - r_{13}r_{22} = r_{31}$ and since the columns of R are orthogonal we have $r_{23}r_{21} + r_{13}r_{11} = -r_{33}r_{31}$. Thus, we can rewrite (45) as

$$(1 + r_{33})(1 - r_{33})c_1 = r_{31}(1 - r_{33})c_3. \quad (46)$$

This means that when $r_{33} \neq \pm1$, we obtain $c_1 = r_{31}c_3/(1 + r_{33})$, as we wanted to show. □

5. Experiments

We evaluate the proposed triangulation method on 5000 randomly sampled image pairs from the Pantheon collection of the Image Matching Competition training set [1, 35], which has an associated 3D reconstruction from COLMAP [29, 30]. We select pairs which have at least 100 covisible 3D points in the COLMAP reconstruction. Figure 3 shows the distribution of the eigenvalue ratio (38) over the selected image pairs. Clearly, for most image pairs, the ratio is close to 1, meaning that our reweighted cost function is close to the original unweighted cost function.

In Figure 2, we show results for our method compared to Lindstrom's [21] and Hartley-Sturm's [10] methods.[1] We take 2D correspondences from the COLMAP reconstruction, compute new 2D points using the respective methods and measure the 2D distance both from the projected associated 3D point (which has been refined by bundle adjustment in COLMAP) and to the original 2D keypoints. The three methods all perform very similarly, with our method falling slightly behind the others. Lindstrom's method is furthermore very efficient, 1.3 – 1.4 times faster than our method in our optimized implementations. We have not implemented an optimized version of Hartley-Sturm's method, but Lindstrom reports that his method is around 50 times faster than a fast implementation of Hartley-Sturm's. Hence, practically, we recommend using Lindstrom's method, except if the eigenvalue ratio is known to be 1 (see Proposition 4.5 for when this happens) in which case our method is optimal.

[1] When comparing with Lindstrom, we refer to his `niter2`-method which is the fastest variant.

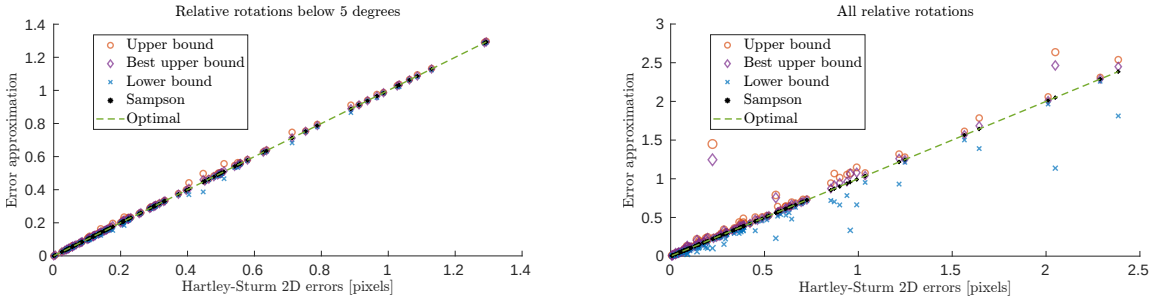

Figure 4. Evaluation of the error bounds (33) and (34) under different amounts of relative rotation between cameras. For cameras with close to the same viewing direction, the bounds are sharp as predicted by the theory. The markers are scaled (logarithmically) by the eigenvalue ratio (38). We subsample 100 random correspondences to plot, for ease of viewing.

In Figure 4, we show the approximating quality of the bounds in equation (33) (denoted "Best upper bound") and equation (34). The markers are scaled according to the eigenvalue ratio (38), illustrating the fact that the bounds become worse with increased ratio as predicted by the theory. Our bounds can be used for outlier rejection, with computation time roughly identical to the Sampson error [27, 28].[2] Under data-dependent assumptions, the Sampson method provides bounds for the true error [27, Section 3], whereas our bounds do not require any such assumptions. However, we find that in practice, it should be recommended to use the Sampson error as it more closely aligns with the optimal value. The noise levels need to be extremely high for Sampson to fail in approximating the true error well, so this is not a practically relevant issue, see Figure 5.

In summary, the experiments show that our triangulation method does not outperform the state-of-the-art but performs competitively. In settings where the optical axes are parallel, our method is preferred as it only involves solving a quadratic equation and it is guaranteed to be optimal. This is promising for future development of reweighted cost functions for other problems in multiple view geometry.

6. Conclusions

We showed that diagonalizing the constraint in optimal 2-view triangulation makes it possible to devise a weighted optimization objective such that the problem reduces from finding the roots of a degree 6 polynomial to finding the roots of a degree 2 polynomial. Further, we showed how to choose the weights to perturb the minimum as little as possible from the unweighted objective. We also derived several bounds on the unweighted objective as direct consequences.

While our experiments showed that prior methods may be preferable in practice for some settings, we also found that the methods developed in this paper are close to the state-of-the-art and they provide approximation guarantees. This illustrates the potential of reweighting the objective to reduce the ED-degree as a method in multiple view geometry, and algebraic optimization more generally, which opens up new avenues for future research. As an example, can the 3-view triangulation problem — which is known to have ED-degree 47 [23, 31] — be simplifed by reweighting? By computational exploration we have found the intriguing non-definite matrix $Q = \begin{bmatrix} \mathbf{0} & F_{12} & F_{13} \\ F_{12}^\top & \mathbf{0} & F_{23} \\ F_{13}^\top & F_{23}^\top & \mathbf{0} \end{bmatrix}$, where F_{ij} denotes the top-left 2×2 block of the fundamental matrix of the matrix pair (C_i, C_j), such that minimizing $\boldsymbol{\epsilon}^\top Q \boldsymbol{\epsilon}$ instead of the standard squared-error $\|\boldsymbol{\epsilon}\|^2$ during 3-view triangulation reduces the ED-degree from 47 to 4. It remains a challenging problem to find all (positive-definite) matrices with that low ED-degree, and determine which best approximates the original squared-error minimization.

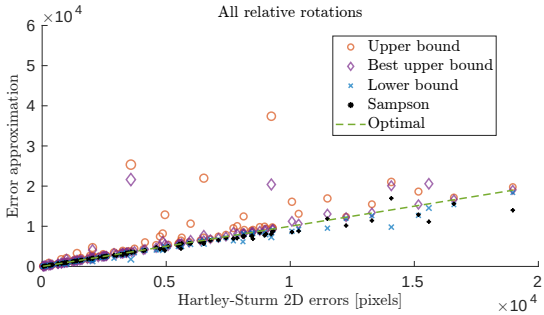

Figure 5. Evaluation of the error bounds (33) and (34) under a large amount of noise on the 2D correspondences. The markers are scaled (logarithmically) by the eigenvalue ratio (38) as in Figure 4. We subsample 100 random correspondences to plot, for ease of viewing. At this point Sampson fails to approximate the optimal errors well, but all of the correspondences would anyway be classified as outliers, so practically this failure is not relevant.

Acknowledgements

All authors have been supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

[2] $\boldsymbol{P}(\boldsymbol{F})$ in Proposition 3.2 is diagonalized using a fast SVD of $F_{2\times 2}$. The diagonalization is computed once per image pair. Further, we can avoid computationally expensive square roots when checking if the upper bound in (34) is smaller than an inlier threshold r, by using that $|\sqrt{\alpha} - \sqrt{\beta}| < r$ is equivalent to $(\alpha + \beta - r^2)^2 < 4\alpha\beta$.

References

[1] Fabio Bellavia, Jiri Matas, Dmytro Mishkin, Luca Morelli, Fabio Remondino, Weiwei Sun, Amy Tabb, Eduard Trulls, Kwang Moo Yi, Sohier Dane, and Ashley Chow. Image matching challenge 2024 - hexathlon. https://kaggle.com/competitions/image-matching-challenge-2024, 2024. 7

[2] Paul Breiding and Sascha Timme. Homotopycontinuation.jl: A package for homotopy continuation in julia. In *Mathematical Software – ICMS 2018*, pages 458–465, Cham, 2018. Springer International Publishing. 2

[3] Paul Breiding, Kathlén Kohn, and Bernd Sturmfels. *Metric algebraic geometry*. Springer Nature, 2024. 2

[4] Wojciech Chojnacki, Michael J. Brooks, Anton Van Den Hengel, and Darren Gawley. On the fitting of surfaces to data with covariances. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 2000. 2

[5] Ondřej Chum, Tomáš Pajdla, and Peter Sturm. The geometric error for homographies. *Computer Vision and Image Understanding (CVIU)*, 2005. 2

[6] Timothy Duff and Felix Rydell. Metric multiview geometry–a catalogue in low dimensions. *arXiv preprint arXiv:2402.00648*, 2024. 2

[7] Olof Enqvist and Fredrik Kahl. Robust optimal pose estimation. In *European Conference on Computer Vision (ECCV)*, 2008. 2

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

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2

[10] Richard I Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding (CVIU)*, 1997. 1, 2, 3, 7

[11] Klas Josephson and Fredrik Kahl. Triangulation of points, lines and conics. *Journal of Mathematical Imaging and Vision (JMIV)*, 2010. 2

[12] Fredrik Kahl and Richard Hartley. Multiple view geometry under the L_∞-norm. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 30(9):1603–1617, 2008. 1, 2

[13] Fredrik Kahl, Sameer Agarwal, Manmohan Krishna Chandraker, David Kriegman, and Serge Belongie. Practical global optimization for multiview geometry. *International Journal of Computer Vision (IJCV)*, 3(79):271–284, 2008. 2

[14] Kenichi Kanatani. *Statistical optimization for geometric computation: theory and practice*. Courier Corporation, 2005. 2

[15] Kenichi Kanatani, Yasuyuki Sugaya, and Hirotaka Niitsuma. Triangulation from two views revisited: Hartley-sturm vs. optimal correction. *practice*, 4(5):99, 2008. 2

[16] Khazhgali Kozhasov, Alan Muniz, Yang Qi, and Luca Sodomaco. On the minimal algebraic complexity of the rank-one approximation problem for general inner products. *arXiv preprint arXiv:2309.15105*, 2023. 2

[17] Viktor Larsson, Kalle Astrom, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[18] Seong Hun Lee and Javier Civera. Triangulation: Why optimize? In *British Machine Vision Conference (BMVC)*, 2019. 2

[19] Seong Hun Lee and Javier Civera. Closed-form optimal two-view triangulation based on angular errors. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[20] Spyridon Leonardos, Roberto Tron, and Kostas Daniilidis. A metric parametrization for trifocal tensors with non-colinear pinholes. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[21] Peter Lindstrom. Triangulation made easy. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 2, 7

[22] Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision (IJCV)*, 1996. 2

[23] Laurentiu G Maxim, Jose I Rodriguez, and Botong Wang. Euclidean distance degree of the multiview variety. *SIAM Journal on Applied Algebra and Geometry*, 4(1):28–48, 2020. 2, 3, 8

[24] David Nister, Richard Hartley, and Henrik Stewenius. Using galois theory to prove structure from motion algorithms are optimal. In *Computer Vision and Pattern Recognition (CVPR)*, 2007. 1, 2

[25] Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. The registration problem revisited: Optimal solutions from points, lines and planes. In *International Conference on Pattern Recognition (ICPR)*, 2006. 2

[26] Felix Rydell, Elima Shehu, and Angélica Torres. Theoretical and numerical analysis of 3d reconstruction using point and line incidences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3748–3757, 2023. 2, 4

[27] Felix Rydell, Angélica Torres, and Viktor Larsson. Revisiting sampson approximations for geometric estimation problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4990–4998, 2024. 2, 8

[28] Paul D Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the bookstein algorithm. *Computer graphics and image processing*, 1982. 2, 8

[29] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[30] 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. 7

[31] H. Stewenius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really? In *International Conference on Computer Vision (ICCV)*, 2005. 3, 8

[32] Peter Sturm. *Vision 3D non calibrée: contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1997. 2

[33] Gabriel Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 1991. 2

[34] Mikhail Terekhov and Viktor Larsson. Tangent sampson error: Fast approximate two-view reprojection error for central camera models. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[35] Eduard Trulls, Yuhe Jin, Kwang Yi, Dmytro Mishkin, Jiri Matas, and Pascal Fua. Image matching challenge 2020. https://www.cs.ubc.ca/research/image-matching-challenge/2020/, 2020. 7