Minimal Solvers for Full DoF Motion Estimation from Asynchronous Tracks

Petr Hruby
ETH Zürich
Rämistrasse 101, 8006 Zürich
petr.hruby@inf.ethz.ch

Marc Pollefeys
ETH Zürich / Microsoft Spatial AI Lab
Rämistrasse 101, 8006 Zürich
marc.pollefeys@inf.ethz.ch

Abstract

We address the problem of estimating both translational and angular velocity of a camera from asynchronous point tracks, a formulation relevant to rolling shutter and event cameras. Since the original problem is non-polynomial, we propose a polynomial approximation, classify the resulting minimal problems, and determine their algebraic degrees. Furthermore, we develop minimal solvers for several problems with low degrees and evaluate them on synthetic and real datasets. The code is publicly available.

1. Introduction

Relative pose estimation is a fundamental task in computer vision, with applications in 3D reconstruction [53, 65], Simultaneous Localization and Mapping (SLAM) [14], camera self-calibration [12], multi-view stereo [22], and visual odometry [51]. Most minimal solvers for relative pose estimation assume a pinhole [50] or radially distorted [30] camera model, where all image rows are captured simultaneously. However, this assumption does not hold for two important sensor types: *rolling shutter cameras* and *event cameras*. Rolling shutter cameras capture images row by row, while event cameras output a continuous stream of intensity changes. When these cameras move, the resulting image differs from that of the pinhole model.

Among these two sensor modalities, rolling shutter cameras have received more attention. Numerous methods have been developed for absolute pose estimation [3–7, 9, 33, 34, 37, 47, 48, 63], image rectification [2, 8, 10, 18–21, 31, 36, 54–57, 59, 60, 67, 69, 70, 73, 77, 79], structure-from-motion [27, 32, 62], and bundle adjustment [28, 43, 44, 49, 64, 66, 72, 75].

Relative pose estimation for rolling shutter cameras, however, remains a particularly challenging problem. Existing methods typically treat the cameras as moving independently, requiring the estimation of 17 parameters in the bifocal case (5 for the relative pose and 6 for the motion of each camera). The full-DoF problem is still unsolved, and many approaches rely on simplifying assumptions: zero angular velocity [13, 78], known angular velocity [41, 42], or known vertical direction [29]. Other methods refine pinhole-based pose estimates with rolling shutter models [40], or target special cases such as planar scenes [35] or camera rigs [71]. Recent works have surveyed minimal solvers for rolling shutter relative pose estimation using points and lines under special order-one models [25], or purely from lines without a camera model [29]. For video-based rolling shutter imagery, [68] proposed a method assuming constant translational velocity over neighboring frames, with known angular velocity.

Event camera pose estimation has been studied less extensively. Some works addressed motion estimation from collinear events, both under known [23, 24] and unknown [76] angular velocity, and from the normal flow [58, 74]. A recent method [68] proposed estimating translational motion from asynchronous point tracks assuming known angular velocity.

In this paper, we extend the problem of [68] to the full-DoF setting, where both translational and angular velocities are unknown. We approximate the rotation with a polynomial model of degree K, classify minimal problems for $K < 6$, and construct minimal solvers for $K = 1$ and $K = 2$. Furthermore, we provide a geometric explanation for why, in the pure translational case, a single essential matrix fits all data, which allows the classical five-point solver [50] to be applied. Throughout the paper, we assume a fully calibrated camera without radial distortion.

2. Problem Formulation

In this section, we formulate the problem of estimating the camera velocity and rotation velocity from asynchronous tracks, introduce polynomial approximations of this problem, and enumerate minimal cases of this problem.

We consider a calibrated perspective camera whose center moves along a straight line with a constant translational velocity $\mathbf{V} \in \mathbb{R}^3$, while simultaneously rotating around a fixed axis $\mathbf{a} \in \mathbb{R}^3$ (with $\|\mathbf{a}\| = 1$) at a constant angular velocity $\omega \in \mathbb{R}$.

Due to gauge freedom, we may, without loss of generality, assume that at the initial time $t = 0$ the camera center is located at the origin,

$$\mathbf{C}(0) = \mathbf{0},$$

and its orientation is given by the identity rotation matrix,

$$\mathbf{R}(0) = \mathbf{I}.$$

For a general time $t \in \mathbb{R}$, the camera center and orientation are given by

$$\mathbf{C}(t) = t \cdot \mathbf{V}, \tag{1}$$

$$\mathbf{R}(t) = \mathbf{I} + \sin(t \cdot \omega)\,[\mathbf{a}]_\times + \big(1 - \cos(t \cdot \omega)\big)[\mathbf{a}]_\times^2, \tag{2}$$

where $[\mathbf{a}]_\times$ denotes the skew-symmetric matrix associated with the vector $\mathbf{a}$.

Let $\{\mathbf{X}_i \in \mathbb{R}^3 \mid i = 1, \dots, n\}$ be a set of $n \in \mathbb{Z}_+$ 3D points. Each point $\mathbf{X}_i$ is observed at $m \in \mathbb{Z}_+$ distinct time instants $\{t_{i,j} \in \mathbb{R} \mid j = 1, \dots, m\}$. The perspective projection $\mathbf{p}_{i,j} \in \mathbb{P}^2$ of point $\mathbf{X}_i$ at time $t_{i,j}$ is given by

$$\mathbf{p}_{i,j} \sim \mathbf{R}(t_{i,j}) \cdot \big(\mathbf{X}_i - \mathbf{C}(t_{i,j})\big), \tag{3}$$

where $\sim$ denotes equality up to a nonzero scale.

In the considered problem formulation, the projections $\mathbf{p}_{i,j}$ and the corresponding time instants $t_{i,j}$ are assumed to be known. The task is to jointly estimate the motion parameters $\mathbf{V}$, $\mathbf{a}$, and ω, as well as the 3D coordinates of the points $\mathbf{X}_i$, $i = 1, \dots, n$, such that the projection model (3) holds for all observations.

2.1. Capture Times for Rolling Shutter Cameras

Event cameras and rolling shutter cameras differ in the way they provide the capture times $t_{i,j}$ corresponding to the projections $\mathbf{p}_{i,j}$. For event cameras, these times are measured explicitly along with the projections, so both $\mathbf{p}_{i,j}$ and $t_{i,j}$ are directly available. In contrast, for rolling shutter cameras, the capture times must be computed from the projection coordinates

$$\mathbf{p}_{i,j} = [u_{i,j}\ v_{i,j}\ 1]^\top$$

and the known camera parameters.

Let $f \in \mathbb{R}$ denote the focal length of the camera in pixels, $h \in \mathbb{R}$ the image height in pixels, t_r the frame readout time, and t_d the inter-frame delay. Since a rolling shutter camera acquires the image row by row, the time required to read a single row is $\frac{t_r}{h}$. The row index of the projection $\mathbf{p}_{i,j}$ can be determined from the y-coordinate of its uncalibrated projection $\mathbf{K}\mathbf{p}_{i,j}$, which is given by

$$f \cdot v_{i,j} + \frac{h}{2}.$$

We choose $t = 0$ to correspond to the capture time of the middle row ($\frac{h}{2}$-th) of the first frame. Under this convention, the capture time $t_{i,1}$ of projection $\mathbf{p}_{i,1}$ in the first frame is

$$t_{i,1} = \frac{f \cdot v_{i,1} \cdot t_r}{h}. \tag{4}$$

Furthermore, the middle row of the second frame is captured at $t = t_r + t_d$, and, in general, the middle row of the j-th frame is captured at

$$t = (j-1) \cdot (t_r + t_d).$$

Thus, the capture time $t_{i,j}$ of projection $\mathbf{p}_{i,j}$ is

$$t_{i,j} = (j-1) \cdot (t_r + t_d) + \frac{f \cdot v_{i,j} \cdot t_r}{h}. \tag{5}$$

The parameters t_r and t_d can be obtained from the camera specifications or through calibration, as in [21]. However, this information may not always be available. Since the delay t_d is often negligible, an alternative approach is to assume $t_d = 0$, redefine the motion parameters as $\mathbf{V} \coloneqq t_r \cdot \mathbf{V}, \omega \coloneqq t_r \cdot \omega.$, and set

$$t_{i,j} = (j-1) + \frac{f \cdot v_{i,j}}{h}. \tag{6}$$

In this case, the recovered velocities are expressed in scaled units (distance per scanline) rather than physical ones (distance per second), therefore, the camera poses at the middle row of the j-th frame, $\mathbf{C}(j-1)$ and $\mathbf{R}(j-1)$, can be determined without knowledge of the readout time.

2.2. Approximating Constant Rotation

Here, we introduce polynomial approximations of the camera motion model (3) and discuss their implications for minimal problem formulations.

Equation (3) imposes constraints on the motion parameters $\mathbf{V}, \mathbf{a}, \omega$ and on the 3D structure of the scene. However, the exact rotation model $\mathbf{R}(t)$ given in (2), which assumes a constant angular velocity $\omega \in \mathbb{R}$, cannot be expressed exactly as a polynomial function. The reason is that, for any fixed $\omega \neq 0$, there exist infinitely many isolated values of t such that $\mathbf{R}(t) = \mathbf{I}$; conversely, for any fixed $t \neq 0$, there exist infinitely many isolated values of ω for which $\mathbf{R}(t) = \mathbf{I}$. This contradicts a fundamental property of polynomial systems, which either have a finite number of solutions or their solution sets have dimension at least one.

Therefore, to solve the problem using standard algebraic methods such as Gröbner basis techniques [39] or homotopy continuation [17], the Rodrigues formula (2) must be approximated by a polynomial expression. For this purpose, we use its Taylor expansion [52]:

$$\mathbf{R}(t) = \sum_{k=0}^{\infty} \frac{t^k \cdot \omega^k}{k!} [\mathbf{a}]_\times^k = \sum_{k=0}^{\infty} \frac{t^k}{k!} [\omega \cdot \mathbf{a}]_\times^k, \tag{7}$$

where $[\mathbf{a}]_\times$ denotes the skew-symmetric matrix of $\mathbf{a}$.

Truncating (7) to terms of degree at most K yields a polynomial approximation $\mathbf{R}_K(t)$ of the rotation matrix. Defining the Euler vector $\mathbf{v} = \omega \cdot \mathbf{a} \in \mathbb{R}^3$, we can write

$$\mathbf{R}_K(t) = \sum_{k=0}^{K} \frac{t}{k!} [\mathbf{v}]_\times^k. \tag{8}$$

In the special case $K = 1$, we obtain the linearized model $\mathbf{R}(t) \approx \mathbf{I} + t\,[\mathbf{v}]_\times$, which has been employed in various applications [5–7].

Since $\mathbf{R}_K(t)$ is generally not an exact rotation matrix, the identity $\mathbf{R}_K(t)^{-T} = \mathbf{R}_K(t)$ does not hold in general. Therefore, there are two natural ways to approximate (3) using $\mathbf{R}_K(t)$:

- **Approximation 1**:

$$\mathbf{p}_{i,j} \sim \mathbf{R}_K(t_{i,j})\big(\mathbf{X}_i - \mathbf{C}(t_{i,j})\big), \tag{9}$$

  which leads to the constraint

$$[\mathbf{p}_{i,j}]_\times\, \mathbf{R}_K(t_{i,j})\big(\mathbf{X}_i - \mathbf{C}(t_{i,j})\big) = 0. \tag{10}$$

- **Approximation 2**:

$$\mathbf{R}_K(t_{i,j})^T\, \mathbf{p}_{i,j} \sim \big(\mathbf{X}_i - \mathbf{C}(t_{i,j})\big), \tag{11}$$

  which leads to the constraint

$$\big[\mathbf{R}_K(t_{i,j})^T\, \mathbf{p}_{i,j}\big]_\times \big(\mathbf{X}_i - \mathbf{C}(t_{i,j})\big) = 0. \tag{12}$$

In both cases, the constraints (10) and (12) are homogeneous in the elements of $\mathbf{V}$ and $\mathbf{X}_i$, implying that velocity and 3D structure can only be recovered up to a common scale. Furthermore, as these constraints are also homogeneous in the elements of $\mathbf{R}_K(t_{i,j})$, we use the adjugate of $\mathbf{R}_K(t_{i,j})$ instead of the inverse to avoid dividing by $\det \mathbf{R}_K(t_{i,j})$, and multiply each term in (8) by $K!$ to avoid division by $k!$. As shown in Sec. 2.4, the two approximations lead to minimal problems with different numbers of solutions.

2.3. Multifocal constraints

Here, we are going to discuss the bi- and trifocal constraints obtained by eliminating 3D points $\mathbf{X}_i$ from constraints (9), (11).

For each observation (i, j), we define the projection matrix

$$\mathbf{P}_{i,j} = \big[\mathbf{I} \,\big|\, -t_{i,j}\mathbf{V}\big]. \tag{13}$$

Furthermore, we set

$$\mathbf{A}_{i,j} = \begin{cases} \mathbf{R}_K(t_{i,j})^{-1}, & \text{for Approximation 1,} \\ \mathbf{R}_K(t_{i,j})^{T}, & \text{for Approximation 2.} \end{cases}$$

With these definitions, the projection equation can be written as

$$\mathbf{A}_{i,j}\, \mathbf{p}_{i,j} \;\sim\; \mathbf{P}_{i,j} \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix}. \tag{14}$$

Fixing a point index $i \in \{1, \dots, n\}$ and two distinct observation indices $j, j' \in \{1, \dots, m\}$, the essential matrix between the corresponding projection matrices $\mathbf{P}_{i,j}$ and $\mathbf{P}_{i,j'}$ is $[\mathbf{V}]_\times$, since the orientation of both cameras is $\mathbf{I}$. Therefore, the transformed image vectors $\mathbf{A}_{i,j}\mathbf{p}_{i,j}$ and $\mathbf{A}_{i,j'}\mathbf{p}_{i,j'}$ satisfy the epipolar constraint

$$\mathbf{p}_{i,j'}^\top\, \mathbf{A}_{i,j'}^\top\, [\mathbf{V}]_\times\, \mathbf{A}_{i,j}\, \mathbf{p}_{i,j} = 0. \tag{15}$$

For Approximation 1, this becomes

$$\mathbf{p}_{i,j'}^\top\, \mathbf{R}_K(t_{i,j'})^{-T}\, [\mathbf{V}]_\times\, \mathbf{R}_K(t_{i,j})^{-1}\, \mathbf{p}_{i,j} = 0, \tag{16}$$

and for Approximation 2,

$$\mathbf{p}_{i,j'}^\top\, \mathbf{R}_K(t_{i,j'})\, [\mathbf{V}]_\times\, \mathbf{R}_K(t_{i,j})^{T}\, \mathbf{p}_{i,j} = 0. \tag{17}$$

In the case of pure translation, the constraint becomes

$$\mathbf{p}_{i,j'}^\top\, [\mathbf{V}]_\times\, \mathbf{p}_{i,j} = 0, \tag{18}$$

which reveals that a single essential matrix $[\mathbf{V}]_\times$ exists for all correspondences, regardless of their capture times.

We can also derive a trifocal constraint. Fix $i \in \{1, \dots, n\}$ and three distinct observation indices $j, j', j'' \in \{1, \dots, m\}$. The trifocal tensor relating $\mathbf{P}_{i,j}$, $\mathbf{P}_{i,j'}$, and $\mathbf{P}_{i,j''}$ takes the form [26]:

$$\begin{aligned} \mathbf{T}_1^{j,j',j''} &= (t_{i,j} - t_{i,j''})\, \mathbf{e}_1 \mathbf{V}^\top - (t_{i,j} - t_{i,j'})\, \mathbf{V}\mathbf{e}_1^\top, \\ \mathbf{T}_2^{j,j',j''} &= (t_{i,j} - t_{i,j''})\, \mathbf{e}_2 \mathbf{V}^\top - (t_{i,j} - t_{i,j'})\, \mathbf{V}\mathbf{e}_2^\top, \\ \mathbf{T}_3^{j,j',j''} &= (t_{i,j} - t_{i,j''})\, \mathbf{e}_3 \mathbf{V}^\top - (t_{i,j} - t_{i,j'})\, \mathbf{V}\mathbf{e}_3^\top, \end{aligned} \tag{19}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^3$ are the standard basis vectors.

From (14), the transformed points $\mathbf{A}_{i,j}\mathbf{p}_{i,j}$, $\mathbf{A}_{i,j'}\mathbf{p}_{i,j'}$, and $\mathbf{A}_{i,j''}\mathbf{p}_{i,j''}$ satisfy the trifocal constraint

$$\begin{aligned} [\mathbf{A}_{i,j'}\mathbf{p}_{i,j'}]_\times \big((\mathbf{A}_{i,j}\mathbf{p}_{i,j})_{(1)}\, \mathbf{T}_1^{j,j',j''} + (\mathbf{A}_{i,j}\mathbf{p}_{i,j})_{(2)}\, \mathbf{T}_2^{j,j',j''} \\ + (\mathbf{A}_{i,j}\mathbf{p}_{i,j})_{(3)}\, \mathbf{T}_3^{j,j',j''}\big) [\mathbf{A}_{i,j''}\mathbf{p}_{i,j''}]_\times = \mathbf{0}. \end{aligned} \tag{20}$$

For Approximation 1, we substitute $\mathbf{A}_{i,j} = \mathbf{R}_K(t_{i,j})^{-1}$, $\mathbf{A}_{i,j'} = \mathbf{R}_K(t_{i,j'})^{-1}$, $\mathbf{A}_{i,j''} = \mathbf{R}_K(t_{i,j''})^{-1}$, and for Approximation 2, $\mathbf{A}_{i,j} = \mathbf{R}_K(t_{i,j})^{T}$, $\mathbf{A}_{i,j'} = \mathbf{R}_K(t_{i,j'})^{T}$, $\mathbf{A}_{i,j''} = \mathbf{R}_K(t_{i,j''})^{T}$.

2.4. Minimal Problems

We now enumerate the minimal problems for the proposed formulations and report their numbers of solutions.

A *minimal problem* is a parametric polynomial system $f(\theta, \mathbf{x}) = 0$ that has a finite, nonzero number of solutions

$\mathbf{x}^*$ for generic parameters θ [15]. Compared to overconstrained systems, minimal problems offer two key advantages: (i) they typically involve the smallest sufficient number of constraints, reducing RANSAC runtime; and (ii) they can be solved via standard algebraic techniques, such as Gröbner bases[39] or homotopy continuation [17].

A necessary condition for minimality is that the problem is *balanced*, *i.e.*, the number of independent constraints equals the number of degrees of freedom (DoF) of the space of unknowns.

For $K = 0$, the model reduces to pure translation, which is a special case of the known angular velocity problem treated in [68]. We now analyze the balanced condition for problems (10) and (12), assuming $K \geq 1$.

The Euler vector $\mathbf{v} \in \mathbb{R}^3$ has 3 DoF. The translational velocity $\mathbf{V} \in \mathbb{R}^3$, being recoverable only up to scale, has 2 DoF. Each 3D point contributes 3 DoF. Hence, the unknown space has $3+2+3 \cdot n$ DoF. Each constraint of the form (10) or (12) fixes 2 DoF. Therefore, the condition for balanced problems is

$$2 \cdot m \cdot n = 5 + 3 \cdot n. \tag{21}$$

Balanced problems correspond to all $m, n \in \mathbb{N}$ satisfying (21), regardless of the approximation type or degree K. For fixed m, the number of points n for a balanced problem is

$$n(m) = \frac{5}{2 \cdot m - 3}. \tag{22}$$

The only pairs (m, n) that are balanced are:
- $m = 2, n = 5$,
- $m = 4, n = 1$.

For $m = 3$, $n(3) = 1.66$, so there is no balanced problem using all constraints. However, we can set
- $m = 3, n = 2$,

and obtain a balanced problem by omitting one equation.

For $m = 1$, the system is underconstrained for all n. For $m \geq 5$, the problems are overconstrained for any n, since $n(4) = 1$ and $n(m)$ increases strictly for $m \geq 2$.

To verify the minimality of the problems and to determine their numbers of solutions, we encoded the case $m = 2, n = 5$ using the bifocal constraints (15), and the cases $m = 3, n = 2$ and $m = 4, n = 1$ using the original constraints (10) or (12). Using Gröbner basis computations, we verified that all balanced problems with $N \leq 6$ (where 6 is an arbitrarily selected bound) are minimal, and we computed their degrees. The degrees are presented in Table 1 for both Approximation1 and Approximation2.

The results in the table indicate that, in certain cases, the chosen type of approximation has a substantial impact on the number of solutions. For example, the problem with $m = 2, n = 5$, and $K = 1$ has 120 solutions under Approximation 1, but only 20 solutions under Approximation 2. Conversely, the problem with $m = 4$, $n = 1$, and $K = 1$ has only 2 solutions for Approximation 1, compared to 8 solutions for Approximation 2. In general, problems with $m = 2$, $n = 5$ and $m = 3$, $n = 2$ tend to have fewer solutions under Approximation 2, whereas problems with $m = 4$, $n = 1$ have identical numbers of solutions for both approximations, except in the case $K = 1$.

	Approximation 1			Approximation 2		
K	m=2	m=3	m=4	m=2	m=3	m=4
1	120	22	2	20	20	8
2	426	120	36	122	102	36
3	732	210	64	276	196	64
4	1038	300	92	430	290	92
5	1344	390	120	584	384	120
6	1650	480	148	738	478	148

Table 1. Numbers of solutions of the minimal problems for estimating translational and angular velocity from asynchronous tracks. Here, m denotes the track length, and K the degree of the rotation approximation. See Section 2.4 for details.

These findings demonstrate that the choice of approximation can significantly influence the runtime of the solver, since the computational cost often scales with the number of solutions. Moreover, the table shows that for $2 \leq K \leq 6$, the number of solutions for each problem grows linearly with K. While it is impossible to verify this numerically for all $K \geq 2$ up to infinity, polynomial dependencies of the number of solutions on problem parameters have been proven for other geometric problems—such as optimal multi-view triangulation [61]—using tools from intersection theory [16].

3. Minimal Solvers

In this section, we propose minimal solvers for selected problems enumerated in Sec. 2.4. Our focus is on problems with a small number of solutions, which are typically more efficient and numerically stable.

We propose solvers based on two approaches: Gröbner basis (GB) and homotopy continuation (HC). All Gröbner basis solvers are generated using the solver generator [39], while the solvers based on homotopy continuation are implemented within the MINUS framework [17]. The solvers are written in C++, and all runtimes were benchmarked on an AMD Ryzen 9 CPU with 3.9 GHz.

For brevity, we denote Approximation 1 by A1 and Approximation 2 by A2 throughout this section.

3.1. Problems with m=2, n=5

We now present minimal solvers for the problems with $n = 5$ points and $m = 2$ observations per point. We begin with the case $K = 1$ and A2, since this problem has a relatively low number of 20 solutions. We model it using five constraints of the form (17), with $i \in \{1, \ldots, 5\}$, $j = 1$, and $j' = 2$. Because the translational velocity $\mathbf{V}$ can only be recovered up to scale, its last element was fixed to 1 be-

fore feeding the system into the automatic generator [39]. The resulting solver has an elimination template with 190 rows and a runtime of about 470 μs.

Since the automatic generator often produces more efficient solvers when the number of variables is reduced, we further eliminated $\mathbf{V}$ using the hidden variable approach: The constraints (17) are linear and homogeneous in $\mathbf{V}$, and can therefore be rewritten as

$$c_{i,1}(\mathbf{v}) \cdot \mathbf{V}_{(1)} + c_{i,2}(\mathbf{v}) \cdot \mathbf{V}_{(2)} + c_{i,3}(\mathbf{v}) \cdot \mathbf{V}_{(3)} = 0, \quad (23)$$

where $i \in \{1, \ldots, 5\}$ is the point index, and $c_{i,1}, c_{i,2}, c_{i,3}$ are polynomials in $\mathbf{v}$. By selecting three points i, i', i'', we construct the matrix

$$\mathbf{M}_{i,i',i''}(\mathbf{v}) = \begin{bmatrix} c_{i,1}(\mathbf{v}) & c_{i,2}(\mathbf{v}) & c_{i,3}(\mathbf{v}) \\ c_{i',1}(\mathbf{v}) & c_{i',2}(\mathbf{v}) & c_{i',3}(\mathbf{v}) \\ c_{i'',1}(\mathbf{v}) & c_{i'',2}(\mathbf{v}) & c_{i'',3}(\mathbf{v}) \end{bmatrix}, \quad (24)$$

which satisfies

$$\mathbf{M}_{i,i',i''}(\mathbf{v}) \cdot \mathbf{V} = \mathbf{0}, \quad (25)$$

$$\det \mathbf{M}_{i,i',i''}(\mathbf{v}) = 0. \quad (26)$$

Collecting all 10 equations of the form (26), we fed them into the automatic generator. The resulting solver has an elimination template with only 36 rows and a runtime of approximately 120 μs. Since the GB solvers invert the elimination template, smaller elimination templates typically lead to faster and more stable solvers.

In addition, we developed solvers for the problems with $K = 1$+A1, and with $K = 2$+A2. Due to the high algebraic degrees of these problems (120 and 122), these solvers are based on homotopy continuation. Both are formulated using five constraints of the form (15). The runtime of the solver for $K = 1$+A1 is about 180 ms, while that of the solver for $K = 2$+A2 is about 210 ms.

For comparison, we also constructed a homotopy continuation solver for the problem $K = 1$, A2, using the same formulation as the previous two solvers, achieving a runtime of about 5.2 ms.

3.2. Problems with m=3, n=2

We now present minimal solvers for the problems with $n = 2$ points and $m = 3$ observations per point. We begin with the case $K = 1$, where we developed Gröbner basis solvers for both A1 and A2. These problems are formulated using the bifocal constraints (15) and the trifocal constraints (20). As before, the last element of $\mathbf{V}$ is fixed due to scale ambiguity.

Since the problems are overconstrained, we omit the bifocal constraints involving the observation $\mathbf{p}_{2,3}$, and retain only the first column from the second trifocal constraint. Geometrically, this enforces the existence of a point $\mathbf{X}_2 \in \mathbb{R}^3$ that projects onto $\mathbf{p}_{2,1}$ at time $t_{2,1}$, onto $\mathbf{p}_{2,2}$ at time $t_{2,2}$, and onto the horizontal line passing through $\mathbf{p}_{2,3}$ at time $t_{2,3}$. With this relaxation, the system becomes minimal and can be solved using an automatic solver generator.

The minimal solver for A1 has an elimination template with 363 rows and achieves a runtime of 2.1 ms. The solver for A2 has an elimination template with 387 rows and a runtime of 2.3 ms. We also experimented with the hidden variable approach described in Sec. 3.1, but the coefficients used for building the elimination template grew prohibitively large, making the resulting solvers too unstable for practical use.

In addition, we developed homotopy continuation minimal solvers for both $K = 1$ and $K = 2$.These solvers use the original formulations (10) and (12) with variables $\mathbf{v}$, $\mathbf{V}$, $\mathbf{X}_1$, and $\mathbf{X}_2$. To enforce minimality, only the first element of the constraint involving $\mathbf{p}_{2,3}$ is retained. The runtimes of these solvers are presented in Table 2.

3.3. Problems with m=4, n=1

We now present minimal solvers for the problems with $n = 1$ point and $m = 4$ observations of this point. We first developed Gröbner basis solvers for the problems $K = 1$+A1, $K = 1$+A2, and $K = 2$+A2. These solvers are based on the formulation using both the bifocal constraints (15) and the trifocal constraints (20), with the last element of $\mathbf{V}$ fixed to resolve scale ambiguity. The solver for $K = 1$+A1 has an elimination template with 73 rows and a runtime of 81 μs, while the solver for $K = 1$+A2 has 103 rows and runs in 100 μs. The solver for $K = 2$+A2 is substantially larger, with an elimination template of 1073 rows and a runtime of 31 ms.

We also attempted to design a Gröbner basis solver for the $K = 2$+A1 problem, which also has a relatively low algebraic degree. However, the resulting solver failed to compile, likely due to the large size of the coefficients. Similarly, we explored the hidden variable approach in order to simplify the solvers, but in this case the generator was unable to find a suitable elimination template within a reasonable time.

In addition, we developed homotopy continuation-based minimal solvers for the problems $K = 1$+A1, $K = 1$+A2, $K = 2$+A1, and $K = 2$+A2. These solvers employ the original formulations (10) and (12), with variables $\mathbf{v}$, $\mathbf{V}$, and $\mathbf{X}_1$. The runtimes of these homotopy continuation solvers are reported in Table 2.

4. Local Optimization

We now introduce the local optimization procedure used to refine the output of the minimal solvers presented in Section 3. This refinement has two main benefits. First, it reduces the effect of noise by incorporating a larger number of correspondences. Second, unlike the minimal solvers,

m,n	K	appx	type	#vars	#sols	time (μs)
2,5	1	A1	HC	5	120	1.8E5
2,5	1	A2	HC	5	20	5.2E3
2,5	1	A2	GB	5	20	4.7E2
2,5	1	A2	GB	3	20	**1.2E2**
2,5	2	A2	HC	5	122	2.1E5
3,2	1	A1	HC	11	22	1.6E4
3,2	1	A1	GB	5	22	**2.1E3**
3,2	1	A2	HC	11	20	1.2E4
3,2	1	A2	GB	5	20	2.3E3
3,2	2	A1	HC	11	120	3.4E5
3,2	2	A2	HC	11	120	2.1E5
4,1	1	A1	HC	5	2	6.7E2
4,1	1	A1	GB	5	8	**8.1E1**
4,1	1	A2	HC	8	8	3.2E3
4,1	1	A2	GB	8	8	1.0E2
4,2	2	A1	HC	8	36	6.6E4
4,2	2	A2	HC	8	36	2.8E4
4,2	2	A2	GB	5	36	3.1E4

Table 2. Overview of the minimal solvers presented in Section 3. Here, m and n denote the track length and number of tracks; K is the maximal degree of the rotation approximation; *appx* indicates the type of approximation (Sec. 2.2); and *type* specifies the solver method, either HC (Homotopy Continuation) or GB (Gröbner Basis). #vars denotes the number of estimated variables (*i.e.*, not eliminated beforehand). Runtimes are reported in microseconds, with the lowest time for each (m, n) highlighted in bold.

which rely on the approximate rotation model (8), the local optimization employs the exact rotation model (2), thereby mitigating the errors caused by the approximation. This local optimization is based on the Ceres framework [1], although efficiency can be further improved by employing the Poselib framework [38]. We apply this local optimization procedure to refine the solutions in the real experiments in Sec. 5.3.

The optimization is initialized with estimates $\mathbf{v}_0$ of the Euler vector and $\mathbf{V}_0$ of the translational velocity obtained with a minimal solver. From these, the initial rotation axis $\mathbf{a}_0$ and angular velocity ω_0 are obtained as $\mathbf{a}_0 = \frac{1}{\|\mathbf{v}_0\|} \cdot \mathbf{v}_0$ and $\omega_0 = \|\mathbf{v}_0\|$. For every pair of correspondences $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i,j'}$, captured at times $t_{i,j}$ and $t_{i,j'}$, respectively, we construct an essential matrix

$$\mathbf{E}_{i,j,j'} = \mathbf{R}(t_{i,j'}) \, [\mathbf{V}]_\times \, \mathbf{R}(t_{i,j})^T. \tag{27}$$

The local optimization minimizes the sum of squared Sampson errors

$$\frac{\mathbf{p}_{i,j'}^T \mathbf{E}_{i,j,j'} \mathbf{p}_{i,j}}{\sqrt{\|(\mathbf{E}_{i,j'} \mathbf{p}_{i,j})_{(1:2)}\|^2 + \|(\mathbf{E}_{i,j'}^T \mathbf{p}_{i,j'})_{(1:2)}\|^2}}, \tag{28}$$

over all correspondences that are inliers to the initial motion estimate.

5. Experiments

In this section, we evaluate the minimal solvers introduced in Section 3 and summarized in Table 2. We first present synthetic experiments to assess numerical stability (Sec. 5.1) and robustness to noise as well as to high angular velocities (Sec. 5.2). Finally, we validate the solvers in real-world experiments (Sec. 5.3) using the rolling shutter datasets Fastec and Carla [46].

5.1. Numerical Stability

To evaluate the numerical stability of the proposed solvers, we synthetically generated instances of the corresponding minimal problems, using the approximate rotation $\mathbf{R}_K(\mathbf{v})$ instead of the exact rotation $\mathbf{R}(\mathbf{v})$. For each problem instance, we sampled an Euler vector $\mathbf{v}_{GT}$ from the normal distribution with $\mu = \mathbf{0}$ and $\sigma = 0.1$, and a translation vector $\mathbf{V}_{GT}$ from a normal distribution with $\mu = \mathbf{0}$ and $\sigma = 1$. We further sampled n points $\mathbf{X}_i$ from a normal distribution with $\mu = [0\ 0\ 2]^T$ and $\sigma = 1$, and, for each point, we generated m capture times $t_{i,j}$ from a normal distribution with $\mu = 0$ and $\sigma = 1$.

The image points $\mathbf{p}_{i,j}$ were obtained by projecting $\mathbf{X}_i$ according to (9) for A1, and according to (11) for A2. Each solver was applied to recover an estimate of the Euler vector $\mathbf{v}_{est}$ and the translation vector $\mathbf{V}_{est}$. To assess accuracy, we constructed the rotation matrices $\mathbf{R}(\mathbf{v}_{GT})$ and $\mathbf{R}(\mathbf{v}_{est})$ at time $t = 1$, and measured the rotation error as the angle of the relative rotation $\mathbf{R}(\mathbf{v}_{est})^T \mathbf{R}(\mathbf{v}_{GT})$. The translation error was computed as the angle between $\mathbf{V}_{GT}$ and $\mathbf{V}_{est}$.

The results, depicted in Figure 1, show that all solvers are sufficiently stable for the use in standard pose estimation pipelines. However, the Gröbner basis solvers for the cases with $m = 3$ and $m = 4$ display somewhat lower stability, which may negatively affect performance in some cases.

5.2. Robustness to Angular Velocity and Noise

We now evaluate the solvers in a more realistic scenario, where the image projections are generated using the exact rotation $\mathbf{R}(\mathbf{v})$. Since the solvers instead rely on the approximation $\mathbf{R}_K(\mathbf{v})$ with $K = 1$ or $K = 2$, the recovered motion parameters have nonzero error, even in the absence of projection noise. This experiment therefore evaluates the accuracy of the approximations used in the solvers.

The experimental setup follows the same procedure as in the previous section, except that the angular velocity ω is treated as a controllable parameter rather than sampled randomly. We consider two different cases: the *event camera* and the *rolling shutter camera*. In the event camera case, the capture times are sampled randomly, as in Section 5.1, and then ordered increasingly. In the rolling shutter case, the capture times are determined by jointly estimating the projection $\mathbf{p}_{i,j}$ and the capture time $t_{i,j}$ such that (5) holds.

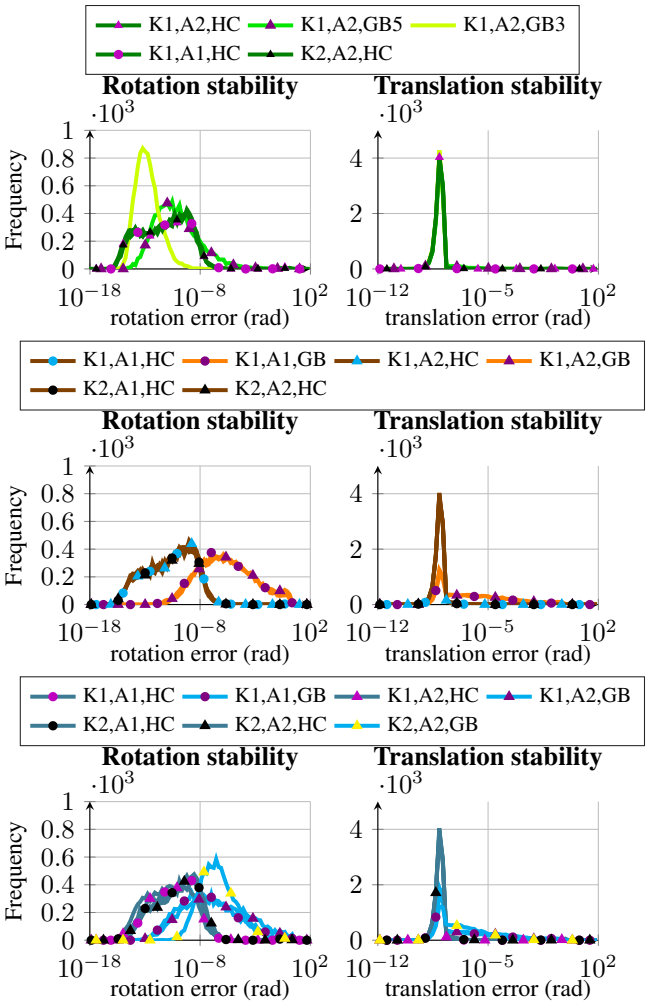

Figure 1. **Stability test.** Histograms of rotation (*left*) and translation (*right*) errors, computed over 10^4 noiseless samples. Results are shown for solvers with *top:* $m = 2, n = 5$, *middle:* $m = 3, n = 2$, and *bottom:* $m = 4, n = 1$. K1 and K2 denote $K = 1$ and $K = 2$, respectively, while A1 and A2 indicate the type of approximation (Sec. 2.2).

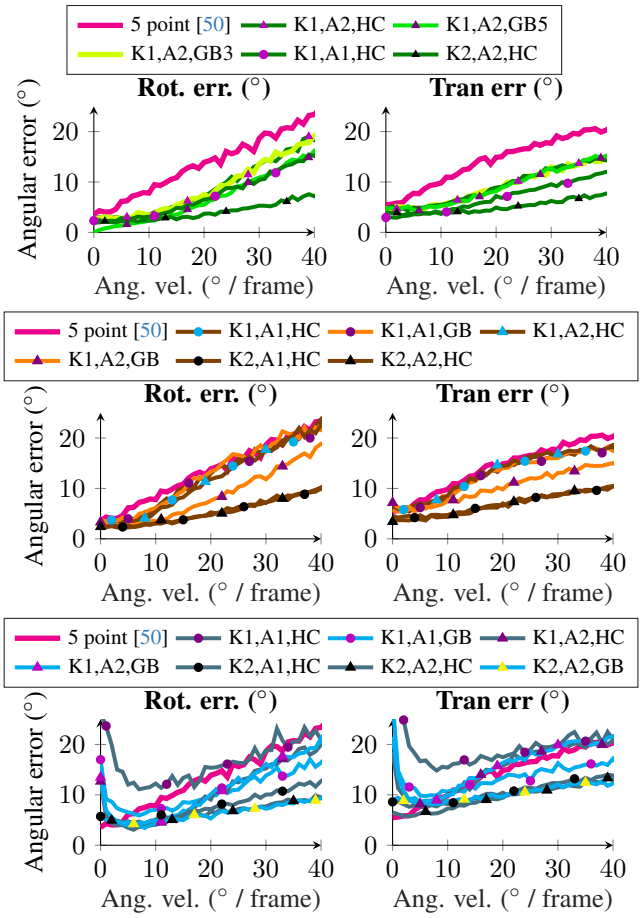

Figure 2. **Noise test, rolling shutter cameras.** Rotation (*left*) and translation (*right*) errors of the solvers with *top:* $m = 2, n = 5$, *middle:* $m = 3, n = 2$, and *bottom:* $m = 4, n = 1$, as the function of the angular velocity ω, Averaged over 1000 synthetic rolling shutter samples with additional noise of magnitude $1px$. The result of the five point solver is shown in all graphs.

Since this is not a polynomial system, we locate the correct $t_{i,j}$ by bisection.

In our experiments, we assume a focal length of $f = 700\,\text{px}$ and add Gaussian noise with $\sigma = (1\,\text{px})/f$ to each projection $\mathbf{p}_{i,j}$. In the rolling shutter case, the capture times $t_{i,j}$ are recovered using the noisy projections. We evaluate the rotation error at the time corresponding to the center of the second frame and the translation error as in the previous section. The resulting errors, plotted as functions of angular velocity, are shown in Figure 2 for the rolling shutter case and in Figure 3 for the event camera case. The noise-free variant of the experiment is included in the Appendix.

We compare our solvers against the five-point solver [50]. When $\omega = 0$, this solver is able to recover the translational velocity $\mathbf{V}$. The results confirm that the errors of all solvers increase with angular velocity, as expected, since the approximation becomes less accurate for higher ω. In the rolling shutter scenario, the five-point solver performs well for low angular velocities, but in the event camera setting its error grows rapidly as ω increases. Overall, most of the proposed solvers demonstrate greater robustness than the five-point solver. In particular, solvers based on quadratic formulations with $m = 2$ or $m = 3$ exhibit the highest robustness, while those assuming $m = 4$ tend to be less stable and appear to be degenerate in the special case of $\omega = 0$.

5.3. Real experiments

Here, we evaluate the proposed solvers on two rolling shutter datasets, Fastec [46] and Carla [46], and compare them

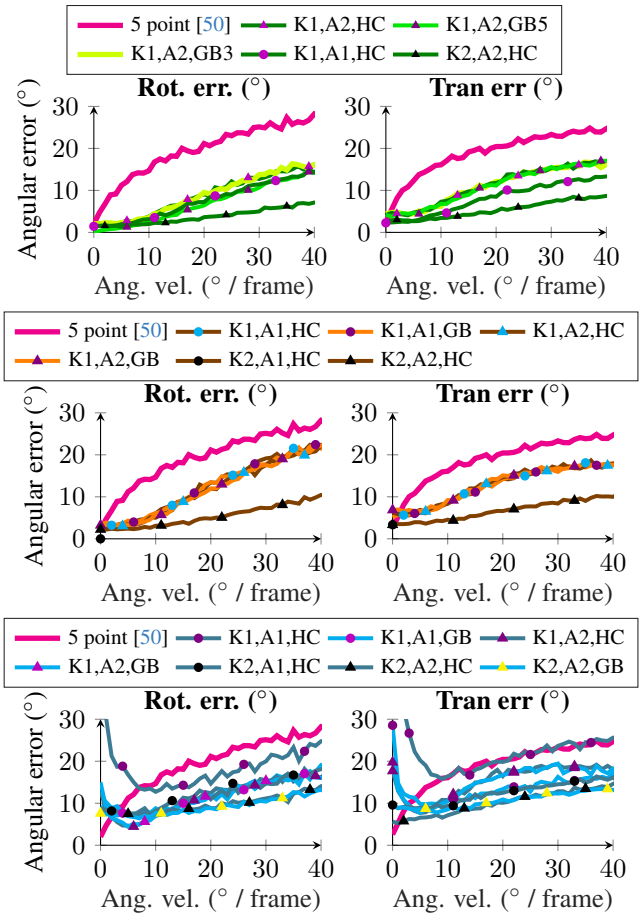

Figure 3. **Noise test, event cameras.** Rotation (*left*) and translation (*right*) errors of the solvers with *top:* $m = 2, n = 5$, *middle:* $m = 3, n = 2$, and *bottom:* $m = 4, n = 1$, as the function of the angular velocity ω, Averaged over 1000 synthetic event samples with additional noise of magnitude $1px$. The result of the five point solver is shown in all graphs.

to the five-point solver [50]. These datasets contain rolling-shutter (RS) and global shutter (GS) images with the same poses. While the rotational velocities in the Fastec sequences are negligible, the Carla dataset typically has a noticeable inter-frame rotation of 1 to 4 degrees. Specifically, we compare the solvers $m = 2, n = 5, A2, K1, HC$, $m = 2, n = 5, A2, K1, GB3$, $m = 2, n = 5, A2, K2, HC$, $m = 3, n = 2, A1, K1, GB$, $m = 3, n = 2, A2, K2, GB$, and $m = 4, n = 1, A2, K1, GB$ against the five-point solver. Since the datasets do not provide ground-truth poses or velocities, we reconstructed pseudo-ground-truth poses using COLMAP [65] from the GS images. For each solver, we then selected sequences of m consecutive RS images, matched them with LightGlue [45], estimated the capture times $t_{i,j}$ following the procedure in Section 2.1, and computed poses using LO-RANSAC [11] with local refinement as described in Section 4.

Solver	AUC1	AUC5	AUC10	AUC20
5 point [50]	0.10	0.41	0.60	0.72
m=2,A2,K1,HC	0.19	0.50	**0.64**	**0.74**
m=2,A2,K1,GB3	0.18	0.49	0.63	**0.74**
m=2,A2,K2,HC	0.18	0.49	0.63	0.73
m=3,A1,K1,GB	0.22	0.50	0.61	0.69
m=3,A2,K2,GB	**0.23**	**0.52**	0.63	0.71
m=4,A2,K1,GB	0.00	0.01	0.08	0.25

Table 3. The AUC score at 1°, 5°, 10°, and 20° on the Carla dataset [46].

Solver	AUC1	AUC5	AUC10	AUC20
5 point [50]	**0.12**	**0.37**	**0.52**	**0.68**
m=2,A2,K1,HC	0.09	0.30	0.46	0.63
m=2,A2,K1,GB3	0.09	0.30	0.46	0.61
m=2,A2,K2,HC	0.09	0.30	0.47	0.63
m=3,A1,K1,GB	0.01	0.06	0.12	0.24
m=3,A2,K2,GB	0.02	0.08	0.14	0.23
m=4,A2,K1,GB	0.01	0.17	0.32	0.47

Table 4. The AUC score at 1°, 5°, 10°, and 20° on the Fastec dataset [46].

The results were evaluated against the pseudo-ground-truth using the same metrics as in Section 5.1, with the pose error defined as the maximum of the rotation and translation errors. We report the area under the recall curve (AUC) of the pose errors, thresholded at $1°, 5°, 10°, 20°$, in Table 3 for the Carla dataset and Table 4 for the Fastec dataset. The results show that on the Carla dataset, which exhibits small but non-negligible angular velocities, the proposed solvers outperform the five-point solver. In contrast, on the Fastec dataset, where inter-frame rotations are negligible, the five-point solver achieves superior performance.

6. Conclusion

In this paper, we addressed the problem of estimating the translational and angular velocity of a camera from asynchronous point tracks, applicable to motion estimation with rolling shutter and event cameras.

Since the original formulation is non-polynomial, we introduced a polynomial approximation, classified the resulting minimal problems, and determined their algebraic degrees. We also provided a geometric explanation for why the five-point solver [50] exactly solves the pure translation case.

We developed minimal solvers for several problems with relatively low degrees (2–122) and evaluated them on synthetic and real data. The experiments show that even with small nonzero rotation, our solvers outperform the five-point solver. The code is available at https://github.com/petrhruby97/asynchronous_full_dof.

References

[1] Sameer Agarwal and Keir Mierle. Ceres solver. http://ceres-solver.org. 6

[2] Omar Ait-Aider and François Berry. Structure and kinematics triangulation with a rolling shutter stereo rig. In *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, 2009. 1

[3] Omar Ait-Aider, Nicolas Andreff, Jean-Marc Lavest, and Philippe Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part II*, 2006. 1

[4] Omar Ait-Aider, Adrien Bartoli, and Nicolas Andreff. Kinematics from lines in a single rolling shutter image. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[5] Cenek Albl, Zuzana Kukelova, and Tomás Pajdla. R6P - rolling shutter absolute pose problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015. 3

[6] Cenek Albl, Zuzana Kukelova, and Tomás Pajdla. Rolling shutter absolute pose problem with known vertical direction. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

[7] Cenek Albl, Zuzana Kukelova, Viktor Larsson, and Tomás Pajdla. Rolling shutter camera absolute pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 1, 3

[8] Cenek Albl, Zuzana Kukelova, Viktor Larsson, Michal Polic, Tomás Pajdla, and Konrad Schindler. From two rolling shutters to one global shutter. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020. 1

[9] Fang Bai, Agniva Sengupta, and Adrien Bartoli. Scanline homographies for rolling-shutter plane absolute pose. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 2022. 1

[10] Mingdeng Cao, Zhihang Zhong, Jiahao Wang, Yinqiang Zheng, and Yujiu Yang. Learning adaptive warping for real-world rolling shutter correction. *CoRR*, 2022. 1

[11] Ondrej Chum, Jiri Matas, and Josef Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*, 2003. 8

[12] Andrea Porfiri Dal Cin, Timothy Duff, Luca Magri, and Tomás Pajdla. Minimal perspective autocalibration. In *CVPR*. IEEE, 2024. 1

[13] Yuchao Dai, Hongdong Li, and Laurent Kneip. Rolling shutter camera relative pose: Generalized epipolar geometry. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016. 1

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep SLAM. *CoRR*, 2017. 1

[15] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomás Pajdla. PLMP - point-line minimal problems in complete multi-view visibility. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024. 4

[16] David Eisenbud and Joe Harris. *3264 and All That*. Cambridge University Press, Cambridge, England, first edition. edition, 2016. 4

[17] Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret H. Regan, David da Costa de Pinho, Elias P. Tsigaridas, Charles W. Wampler, Jonathan D. Hauenstein, Peter J. Giblin, Benjamin B. Kimia, Anton Leykin, and Tomás Pajdla. TRPLP - trifocal relative pose from lines at points. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020. 2, 4

[18] Bin Fan, Yuchao Dai, and Ke Wang. Rolling-shutter-stereo-aware motion estimation and image correction. *Comput. Vis. Image Underst.*, 2021. 1

[19] Bin Fan, Yuchao Dai, Zhiyuan Zhang, and Ke Wang. Differential sfm and image correction for a rolling shutter stereo rig. *Image Vis. Comput.*, 2022.

[20] Bin Fan, Yuchao Dai, and Hongdong Li. Rolling shutter inversion: Bring rolling shutter images to high framerate global shutter video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[21] Per-Erik Forssén and Erik Ringaby. Rectifying rolling shutter video from hand-held devices. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, 2010. 1, 2

[22] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010. 1

[23] Ling Gao, Hang Su, Daniel Gehrig, Marco Cannici, Davide Scaramuzza, and Laurent Kneip. A 5-point minimal solver for event camera relative motion estimation. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 8015–8025. IEEE, 2023. 1

[24] Ling Gao, Daniel Gehrig, Hang Su, Davide Scaramuzza, and Laurent Kneip. An n-point linear solver for line and motion estimation with event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, 2024. 1

[25] Marvin Anas Hahn, Kathlén Kohn, Orlando Marigliano, and Tomás Pajdla. Order-one rolling shutter cameras. *CoRR*, 2024. 1

[26] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 3

[27] Johan Hedborg, Erik Ringaby, Per-Erik Forssén, and Michael Felsberg. Structure and motion estimation from rolling shutter video. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, 2011. 1

[28] Johan Hedborg, Per-Erik Forssén, Michael Felsberg, and Erik Ringaby. Rolling shutter bundle adjustment. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, 2012. 1

[29] Petr Hruby and Marc Pollefeys. Single-scanline relative pose estimation for rolling shutter cameras. In *ICCV*, 2025. 1

[30] Petr Hruby, Viktor Korotynskiy, Timothy Duff, Luke Oeding, Marc Pollefeys, Tomás Pajdla, and Viktor Larsson. Four-view geometry with unknown radial distortion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, 2023. 1

[31] Chao Jia and Brian L. Evans. Probabilistic 3-d motion estimation for rolling shutter video rectification from visual and inertial measurements. In *14th IEEE International Workshop on Multimedia Signal Processing, MMSP 2012, Banff, AB, Canada, September 17-19, 2012*, 2012. 1

[32] Georg Klein and David William Murray. Parallel tracking and mapping on a camera phone. In *Science & Technology Proceedings, 8th IEEE International Symposium on Mixed and Augmented Reality 2009, ISMAR 2009, Orlando, Florida, USA, October 19-22, 2009*, 2009. 1

[33] Zuzana Kukelova, Cenek Albl, Akihiro Sugimoto, and Tomás Pajdla. Linear solution to the minimal absolute pose rolling shutter problem. In *Computer Vision - ACCV 2018 - 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part III*, 2018. 1

[34] Zuzana Kukelova, Cenek Albl, Akihiro Sugimoto, Konrad Schindler, and Tomás Pajdla. Minimal rolling shutter absolute pose with unknown focal length and radial distortion. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, 2020. 1

[35] Yizhen Lao and Omar Ait-Aider. Rolling shutter homography and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2780–2793, 2021. 1

[36] Yizhen Lao, Omar Ait-Aider, and Helder Araújo. Robustified structure from motion with rolling-shutter camera using straightness constraint. *Pattern Recognit. Lett.*, 2018. 1

[37] Yizhen Lao, Omar Ait-Aider, and Adrien Bartoli. Rolling shutter pose and ego-motion estimation using shape-from-template. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, 2018. 1

[38] Viktor Larsson and contributors. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. 6

[39] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *CVPR*, 2017. 2, 4, 5

[40] Chang-Ryeol Lee and Kuk-Jin Yoon. Monocular visual odometry with a rolling shutter camera. *CoRR*, 2017. 1

[41] Chang-Ryeol Lee and Kuk-Jin Yoon. Inertial-aided rolling shutter relative pose estimation. *CoRR*, 2017. 1

[42] Chang-Ryeol Lee, Ju Hong Yoon, Min-Gyu Park, and Kuk-Jin Yoon. Gyroscope-aided relative pose estimation for rolling shutter cameras. *CoRR*, 2019. 1

[43] Moyang Li, Peng Wang, Lingzhe Zhao, Bangyan Liao, and Peidong Liu. Usb-nerf: Unrolling shutter bundle adjusted neural radiance fields. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024. 1

[44] Bangyan Liao, Delin Qu, Yifei Xue, Huiqing Zhang, and Yizhen Lao. Revisiting rolling shutter bundle adjustment: Toward accurate and fast solution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, 2023. 1

[45] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 8

[46] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 7, 8

[47] Ludovic Magerand, Adrien Bartoli, Omar Ait-Aider, and Daniel Pizarro. Global optimization of object pose and motion from a single rolling shutter image with automatic 2d-3d matching. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*, 2012. 1

[48] Marci Meingast, Christopher Geyer, and Shankar Sastry. Geometric models of rolling-shutter cameras. *CoRR*, abs/cs/0503076, 2005. 1

[49] Thanh-Tin Nguyen and Maxime Lhuillier. Adding synchronization and rolling shutter in multi-camera bundle adjustment. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*, 2016. 1

[50] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004. 1, 7, 8, 2

[51] David Nistér, Oleg Naroditsky, and James R. Bergen. Visual odometry. In *CVPR*, 2004. 1

[52] Tomas Pajdla. *Elements of Geometry for Robotics*. Czech Technical University in Prague, 2021. 2

[53] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[54] Pulak Purkait and Christopher Zach. Minimal solvers for monocular rolling shutter compensation under ackermann motion. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, 2018. 1

[55] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in manhattan world. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

[56] Delin Qu, Yizhen Lao, Zhigang Wang, Dong Wang, Bin Zhao, and Xuelong Li. Towards nonlinear-motion-aware and occlusion-robust rolling shutter correction. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, 2023.

[57] Delin Qu, Bangyan Liao, Huiqing Zhang, Omar Ait-Aider, and Yizhen Lao. Fast rolling shutter correction in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023. 1

[58] Zhongyang Ren, Bangyan Liao, Delei Kong, Jinghang Li, Peidong Liu, Laurent Kneip, Guillermo Gallego, and Yi Zhou. Motion and structure from event-based normal flow. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[59] Vijay Rengarajan, A. N. Rajagopalan, and Rangarajan Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016. 1

[60] Erik Ringaby and Per-Erik Forssén. Efficient video rectification and stabilisation for cell-phones. *Int. J. Comput. Vis.*, 2012. 1

[61] Felix Rydell, Elima Shehu, and Angélica Torres. Theoretical and numerical analysis of 3d reconstruction using point and line incidences. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3725–3734. IEEE, 2023. 4

[62] Olivier Saurer, Kevin Köser, Jean-Yves Bouguet, and Marc Pollefeys. Rolling shutter stereo. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 2013. 1

[63] Olivier Saurer, Marc Pollefeys, and Gim Hee Lee. A minimal solution to the rolling shutter pose estimation problem. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, 2015. 1

[64] Olivier Saurer, Marc Pollefeys, and Gim Hee Lee. Sparse to dense 3d reconstruction from rolling shutter images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016. 1

[65] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 8

[66] David Schubert, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. Direct sparse odometry with rolling shutter. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, 2018. 1

[67] Wei Shang, Dongwei Ren, Chaoyu Feng, Xiaotao Wang, Lei Lei, and Wangmeng Zuo. Self-supervised learning to bring dual reversed rolling shutter images alive. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, 2023. 1

[68] Hang Su, Yunlong Feng, Daniel Gehrig, Panfeng Jiang, Ling Gao, Xavier Lagorce, and Laurent Kneip. A linear n-point solver for structure and motion from asynchronous tracks. In *ICCV*, 2025. 1, 4

[69] Yufen Sun, Gang Liu, and Yue Sun. An affine motion model for removing rolling shutter distortions. *IEEE Signal Process. Lett.*, 2016. 1

[70] Subeesh Vasu, Mahesh Mohan M. R., and A. N. Rajagopalan. Occlusion-aware rolling shutter rectification of 3d scenes. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018. 1

[71] Ke Wang, Bin Fan, and Yuchao Dai. Relative pose estimation for stereo rolling shutter cameras. In *IEEE International Conference on Image Processing, ICIP 2020, Abu Dhabi, United Arab Emirates, October 25-28, 2020*, 2020. 1

[72] Bo Xu, Ziao Liu, Mengqi Guo, Jiancheng Li, and Gim Hee Lee. Urs-nerf: Unordered rolling shutter bundle adjustment for neural radiance fields. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXV*, 2024. 1

[73] Weilong Yan, Robby T. Tan, Bing Zeng, and Shuaicheng Liu. Deep homography mixture for single image rolling shutter correction. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, 2023. 1

[74] Dehao Yuan, Levi Burner, Jiayi Wu, Minghui Liu, Jingxi Chen, Yiannis Aloimonos, and Cornelia Fermüller. Learning normal flow directly from event neighborhoods. *arXiv preprint arXiv:2412.11284*, 2024. 1

[75] Yongcong Zhang, Bangyan Liao, Yifei Xue, Chen Lu, Peidong Liu, and Yizhen Lao. RSL-BA: rolling shutter line bundle adjustment. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LIX*, 2024. 1

[76] Ji Zhao, Banglei Guan, Zibin Liu, and Laurent Kneip. Full-dof egomotion estimation for event cameras using geometric solvers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 11515–11524, 2025. 1

[77] Bingbing Zhuang and Quoc-Huy Tran. Image stitching and rectification for hand-held cameras. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII*, 2020. 1

[78] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017. 1

[79] Bingbing Zhuang, Quoc-Huy Tran, Pan Ji, Loong-Fah Cheong, and Manmohan Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019. 1