Full-DoF Egomotion Estimation for Event Cameras Using Geometric Solvers

Supplementary Material

Appendix

A. Optimization

The comparison of the proposed optimization problems is summarized in Tab. 1.

Problem	Geometry	obs.	#line	var.	matrix
(IncMin)	incidence	raw	1	$\boldsymbol{\omega}$	$\mathbf{M}_{6\times6}$
(IncBat)	incidence	raw	≥ 2	$\boldsymbol{\omega}$	$(\mathbf{M}_i)_{6\times6}$
(CopMin)	coplanarity	flow	1	$\boldsymbol{\omega}$	$\mathbf{N}_{3\times3}$
(CopBat)	coplanarity	flow	≥ 2	$\boldsymbol{\omega}$	$(\mathbf{N}_i)_{3\times3}$

Table 1. Comparison of optimization problems. obs: observations; var: variable; raw: raw events; flow: normal flow; Min: minimal configuration (a single line); Bat: a batch of lines.

A.1. Remark

Remark 1: In optimization problems (IncBat) (CopBat) with a batch of line observations, the objectives are summation of minimal eigenvalues. It is also possible to replace λ_{min} by a monotonically increasing function with respect to λ_{min}. Usually, we can use λ^p_{min}, where p is an exponent. Popular selection of p includes 1, 2, and $1/2$. The optimal selection of p is determined by the noise level of observations. In this work, we simply set p as 1.

Remark 2: Given ground truth $\boldsymbol{\omega}_{gt}$ and exact rotation parametrization, the smallest eigenvalue $\lambda_{\text{min}}(\mathbf{M}_i(\boldsymbol{\omega}))$ is about 10^{-16}. Given ground truth $\boldsymbol{\omega}_{gt}$ and approximated rotation parametrization, $\lambda_{\text{min}}(\mathbf{M}_i(\boldsymbol{\omega}))$ is about $10^{-12} \sim 10^{-9}$. Since the smallest eigenvalues of $\mathbf{M}_i(\boldsymbol{\omega})$ in our problems are significantly small, the floating number calculation causes insufficient accuracy and the termination conditions of optimization methods are easily triggered. To mitigate these issues, we multiply matrices $\{\mathbf{M}_i(\boldsymbol{\omega})\}_i$ by a factor of 10^6. This does not have any influence on the optimum of $\boldsymbol{\omega}$, and it only increases the objective λ_{min} by a certain factor.

Remark 3: If an event has an associated weight, we can simply multiply the weight by the corresponding row of the matrix $\mathbf{A}$ or $\mathbf{B}$.

B. Egomotion Estimation for Pure Rotation Cases

Pure rotational motion is often a degenerate case in relative pose estimation, especially for sparse geometric solvers. In this section, we will discuss how to deal with it.

Proposition 1. *Pure rotation leads to*

$$\text{rank}([\mathbf{f}'_1, \cdots, \mathbf{f}'_N]) = 2 \tag{1}$$

for unique events $\{\mathbf{f}'_j\}_{j=1}^N$ of a 3D line. Bearing vectors $\{\mathbf{f}'_j\}_{j=1}^N$ lies in a same plane whose normal is $\mathbf{e}_2$.

Proof. Since the motion is pure rotation, the origin of any $\mathbf{f}'_j$ is the optical center. Meanwhile, any $\mathbf{f}'_j$ intersects with a 3D line. Thus bearing vectors $\{\mathbf{f}'_j\}_{j=1}^N$ lie on a same plane spanned by the optical center and the 3D line whose normal is $\mathbf{e}_2$. □

When rotation is known, Eq. (1) provides a criterion to determine whether the motion is pure rotation, i.e., linear velocity $\mathbf{v} = \mathbf{0}$. We can identify the pure rotation cases and immediately know the linear velocity $\mathbf{v}$ is zero.

Proposition 2. *When pure rotation occurs, we have* $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{M}) = 4$. *The null space of* $\mathbf{A}$ *has two orthogonal basis* $[\mathbf{e}_2; 0; 0; 0]$ *and* $[0; 0; 0; \mathbf{e}_2]$.

Proof. According to Proposition 1, bearing vectors $\{\mathbf{f}'_j\}_{j=1}^N$ lies in a same plane whose normal is $\mathbf{e}_2$, so does the vector set $\{t_j\mathbf{f}'_j\}_{j=1}^N$. Note that $[t_1\mathbf{f}'_1, \cdots, t_N\mathbf{f}'_N]$ and $[\mathbf{f}'_1, \cdots, \mathbf{f}'_N]$ are the transpose of $\mathbf{A}_{:,1:3}$ and $\mathbf{A}_{:,4:6}$. In summary, both the left and right three columns of $\mathbf{A}$ has a null space basis $\mathbf{e}_2$. So $\mathbf{A}$ has two null space basis $[\mathbf{e}_2; 0; 0; 0]$ and $[0; 0; 0; \mathbf{e}_2]$. Since $\mathbf{M} = \mathbf{A}^\intercal\mathbf{A}$, $\mathbf{M}$ has the same rank as that of $\mathbf{A}$. □

In [4], the proposed solver can only deal with cases when $\text{rank}(\mathbf{A}) \geq 5$. In Proposition 2, we have proved that $\text{rank}(\mathbf{A}) = 4$. Thus it cannot deal with the pure rotation cases. Our pure rotation identification method provides a supplement to the solver in [4]. In addition, we can use $\mathbf{A}_{:,1:3}$ or $\mathbf{A}_{:,4:6}$ to recover $\mathbf{e}_2$. For $\mathbf{e}_1$ and $\mathbf{e}_3$, they span a plane whose normal is $\mathbf{e}_2$, but they cannot be uniquely determined. So the complete line parameters cannot be recovered.

Additionally, we can recover the angular velocity $\boldsymbol{\omega}$ for pure rotation cases within the incidence formulation. According to Proposition 2, the top-left and bottom-right 3×3 minor of $\mathbf{M}$ have a rank deficiency. So the angular velocity $\boldsymbol{\omega}$ for pure rotation cases can be optimized by

$$\arg\min_{\boldsymbol{\omega}} \sum_{i=1}^{M} \lambda_{\text{min}}(\widehat{\mathbf{M}}_i(\boldsymbol{\omega})) \tag{2}$$

where $\widehat{\mathbf{M}}$ is bottom-right 3×3 minor of $\mathbf{M}$, i.e., $\widehat{\mathbf{M}} = \mathbf{M}_{4:6,4:6}$.

The proposed coplanarity formulation can recover the angular velocity $\boldsymbol{\omega}$ natively for pure rotation cases.

C. Experiments

C.1. Simulation

In the simulation tests, we observed that the recovered angular velocities were unsatisfactory when only a single line ($M = 1$) was available. To investigate the underlying reasons for this phenomenon, we conducted additional experiments.

The runtime and numerical stability for the minimal configuration ($M = 1$ and $N = 100$) are presented in Tab. 2. While the objective values λ_{min} are quite small, the accuracy of all solvers remains inadequate, as evidenced by the significantly low success rates. This phenomenon can be intuitively explained. With only one line, it is possible to orbit around that line while simultaneously adjusting the translational velocity to compensate for the added rotational velocity about an axis parallel to the line. This creates a form of rotation-translation ambiguity. Although this ambiguity might not be perfect, the constant velocity assumption in our motion model makes approximations in any case. Thus, it is reasonable to conclude that such ambiguity significantly impacts the performance.

To further validate this observation, we visualize the objective landscapes in Fig. 1. When only a single line is used, the objective functions lack a clear global minimum. When two lines are used, the objective functions still have relatively large convergence regions. However, when more than three lines are included, the objective functions exhibit distinct global minima.

In the synthetic experiments, the success rates of solvers do not reach 100% even without noise. There are several reasons for this. First, for the approximative rotation parametrization, an approximation of the objective is implicitly introduced. As a result, even its global minimum may deviate slightly from the ground truth. This also explains why we used the results from approximative rotations as an initialization of the cascade method. Second, the objectives are non-convex, and the local optimization method may get trapped in local minima. Third, the objective's landscape may be relatively flat near the global minimum, causing local optimization methods to converge slowly or fail entirely to converge. This explains why the success rate varies when different thresholds are used to define success.

C.2. Real-World Experiment

As our solvers are the first capable of estimating both rotational and translational parameters, we evaluated their applicability on the VECtor dataset [2]. This dataset includes VGA-resolution event recordings captured by a Gen3 Prophesee camera, 200 Hz ground truth camera poses obtained via a MoCap system, and 200 Hz readings from an XSens MTi-30 AHRS IMU. To improve efficiency, the events within each time interval are downsampled to 5000.

We use the GC-RANSAC [1] as the robust estimation framework. An angular reprojection [5, 6] threshold of $0.2°$ is used for inlier selection, applied consistently throughout both the main iterations and the local refinement stages. The spherical radius r in neighborhood graph construction is set as 50. The main iterations stop once there is a 0.99 probability that at least one outlier-free set has been sampled, or when the maximum of 1000 iterations is reached. In the local refinement stage, the maximum number of inner RANSAC iterations is set to 20.

The results of line cluster extraction are shown in Fig. 2. Events are accumulated into frames, which has a resolution of 640×480. Line segments are then extracted from the undistorted frames, and events located within 5 pixels of these line segments are considered as part of the line clusters.

We have also conducted a comparison with two methods with known angular velocities [3, 4]. Using the same dataset and settings described in Table 2 of our paper, we obtained the comparison results of estimated linear velocities reported in Tab. 3. We can see that the solvers in [3, 4] have better results than our proposed full-DoF solvers. This is reasonable, as the comparison methods leverage IMU readings to solve the rotational parameters of the motion.

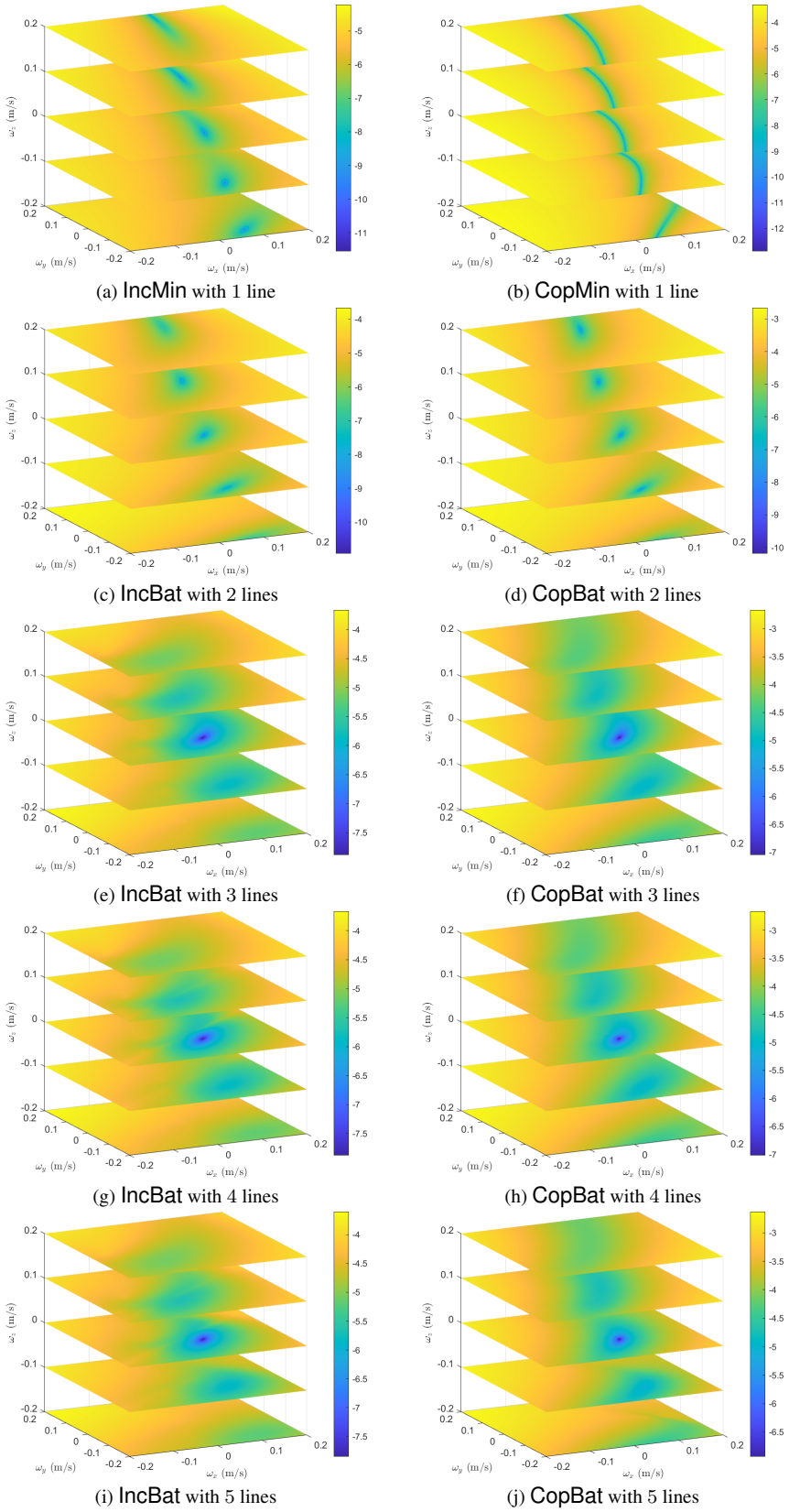
Seq. Name	ICCV23 [3]	CVPR24 [4]	IncBat	CopBat
desk	23.5	21.8	23.0	25.1
mountain	18.2	16.4	17.5	18.7
sofa	19.7	18.9	21.1	20.6

Table 3. Experiment results of the comparison methods. We report the median errors of recovered linear velocities ε_{lin} (unit: degree). This table supplements Table 2 of our paper.

C.3. Discussion

Our method shows promising results, it also has a few limitations. (1) The proposed method is best-suited for scenes that contain long straight lines. (2) The performance of certain solvers heavily relies on the quality of the extracted normal flow. (3) For short time intervals, assuming constant angular and linear velocities is reasonable. However, during aggressive or non-uniform motions, sudden velocity changes may violate this assumption. A more appropriate approach would involve using a continuous-time model, such as a cubic B-spline model [7], to model the $\boldsymbol{\omega}(t)$ and $\mathbf{v}(t)$.

There are several potential strategies to reduce the dependence on line and normal flow extraction. First, it is possible to detect event clusters without identifying line segments, which makes it easier to eliminate the need for frame accumulation. When integrating our solver in a hypothesis-and-test framework (such as RANSAC), we can then verify whether the events within the same cluster originate from a



(a) IncMin with 1 line (b) CopMin with 1 line

(c) IncBat with 2 lines (d) CopBat with 2 lines

(e) IncBat with 3 lines (f) CopBat with 3 lines

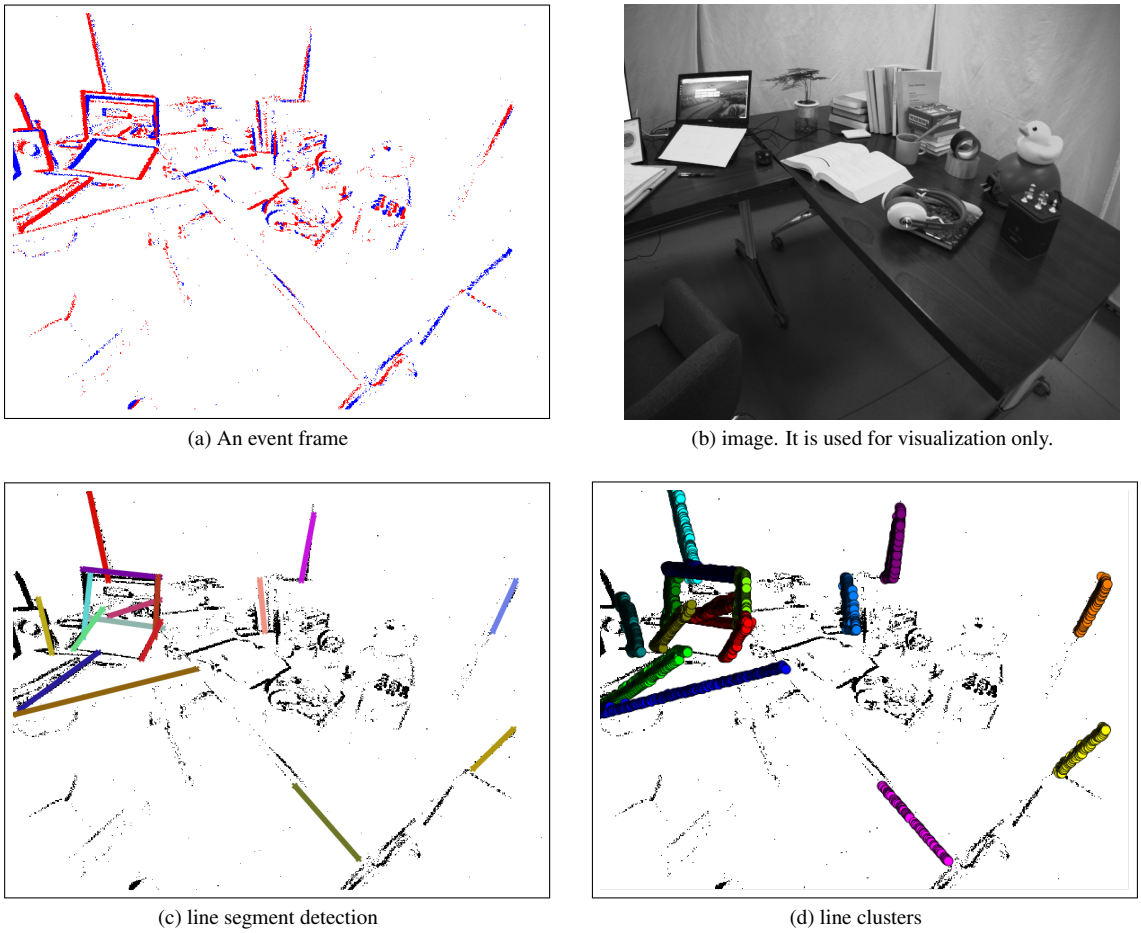
(g) IncBat with 4 lines (h) CopBat with 4 lines

(i) IncBat with 5 lines (j) CopBat with 5 lines

Figure 1. Landscape of the objective functions $\lambda_{\min}$. The events for each line is set as $N = 100$. For better visualization, the pseudocolor and colormap of the objectives use the logarithmic scale.

formulation	rotation	median ε_{ang}	SR1(%)	SR2(%)	median objective	runtime
IncMin	+approx	3.0×10^{-1}	0.8	9.6	5.6×10^{-11}	9.6 ms
	+exact	2.9×10^{-1}	2.0	10.2	5.4×10^{-11}	15.8 ms
	+cascad	2.8×10^{-1}	2.5	11.0	4.0×10^{-11}	15.8 ms
CopMin	+approx	5.8×10^{-1}	0.0	0.8	4.3×10^{-10}	15.7 ms
	+exact	5.7×10^{-1}	0.1	1.3	4.4×10^{-10}	15.9 ms
	+cascad	5.5×10^{-1}	0.0	0.7	2.2×10^{-10}	15.8 ms

Table 2. Runtime and numerical stability for noise-free synthetic data. The configuration is $M = 1$ and $N = 100$. SR1 and SR2 represent success rate (SR) with thresholds of 0.01 and 0.05, respectively. IncMin and CopMin represent the incidence formulation and coplanarity formulation, respectively.



(a) An event frame

(b) image. It is used for visualization only.

(c) line segment detection

(d) line clusters

Figure 2. Line cluster extraction from the *desk-normal* sequence in the VECtor dataset. (a) An event frame generated by accumulating events, where red and blue dots represent events with opposite polarities. (b) The corresponding image. (c) Results of line segment detection. (d) Line cluster extraction by associating events near the line segments. If an event is within 5 pixels of a line segment, draw a circle at the event's position.

line. Second, normal flow can be calculated directly without accumulating frames. The primary focus of our work lies on the relative pose solver, which remains independent of the line/cluster detection and normal flow extraction method. Our method can ultimately be combined with any front-end method.

References

[1] Daniel Barath and Jiri Matas. Graph-cut RANSAC: Local optimization on spatially coherent structures. *IEEE TPAMI*, 44 (9):4961–4974, 2022. 2

[2] Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. VECtor: A versatile

event-centric benchmark for multi-sensor SLAM. *IEEE RAL*, 7(3):8217–8224, 2022. 2

[3] 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 *ICCV*, pages 8049–8059, 2023. 2

[4] 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 *CVPR*, 2024. 1, 2

[5] Laurent Kneip and Paul Furgale. OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In *ICRA*, pages 12034–12043, 2014. 2

[6] Seong Hun Lee and Javier Civera. Closed-form optimal two-view triangulation based on angular errors. In *ICCV*, pages 2681–2689, 2019. 2

[7] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6): 1425–1440, 2018. 2