CONTRAPOSE: GEOMETRY-AWARE CONTRAST MAXIMIZATION FOR EVENT-BASED OBJECT TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate 6DoF object tracking under rapid motion remains challenging for frame-based cameras due to motion blur and limited temporal resolution. Event cameras alleviate these limitations, but their sparse asynchronous measurements provide little direct appearance information, making conventional appearance-based pose estimation difficult. We introduce ConTraPosE, a tracking method based on a geometry-aware contrast formulation for estimating 6DoF object motion directly from raw events. Given a known CAD model and a candidate pose trajectory, we use the object geometry to construct its induced image-plane motion field, warp the observed events accordingly, and evaluate the resulting alignment with a normalized contrast objective. This provides a direct pose-update signal without image reconstruction or explicit event-to-object correspondences. We integrate this objective into continuous event-only tracking with adaptive temporal windows and optionally incorporate depth as an additional geometric constraint for absolute pose correction. Experiments on Event6D and SPADES show strong performance in both event-only and event–depth settings, with consistent robustness across different object geometries, appearances, motion regimes, event corruption, occlusion, and initialization perturbations. These results demonstrate that geometry-induced motion consistency provides an effective alternative to appearance-based comparison for event-driven 6DoF object tracking.

1 INTRODUCTION

Estimating the six-degrees-of-freedom (6DoF) pose of an object relative to a camera is a fundamental computer vision problem, crucial for applications such as robotic manipulation, augmented reality, autonomous systems, and on-orbit rendezvous and proximity operations (Guan et al., 2024; Opromolla et al., 2017). These applications require pose estimates that are both precise and available at high temporal resolution. In highly dynamic scenes, however, frame-based cameras are constrained by their limited sampling rate, and fast motion can cause motion blur and missed pose changes between frames. Event cameras are particularly well suited to such settings, offering a favorable trade-off between speed and energy efficiency (Gallego et al., 2022; Plizzari et al., 2022). Unlike conventional cameras, which capture full frames at fixed intervals, they report per-pixel brightness changes asynchronously with microsecond latency, providing high temporal resolution and reduced motion blur. These properties make event cameras a promising modality for 6DoF pose estimation over time, opening avenues for event-based 6DoF tracking, refinement, and related tasks (Kim et al., 2016; Glover et al., 2024; Liu et al., 2024; Rebecq et al., 2017).

To fully leverage these opportunities, it is essential to develop robust methods for optimizing the 6DoF pose of an object over time directly from observed events. As with standard RGB cameras (Li et al., 2020; Labbé et al., 2023; Krull et al., 2014), the literature has explored several ways to formulate this optimization when the object geometry is known from a CAD model. One line of work identifies common geometric primitives between the observed events and the known object geometry, and solves for the pose from these correspondences (Liu et al., 2024; 2026; Li et al., 2026; Wang et al., 2026; Liu et al., 2025). However, these approaches assume that such shared primitives can be reliably extracted and remain persistent over time to maintain correspondences, which is not straightforward given the sparse nature of event observations. A second, more widely adopted line of work avoids explicit correspondences by shifting the problem toward direct representation comparison. It projects the known object geometry under a candidate 6DoF pose and directly com-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

pares the resulting visual with the observed object appearance, and optimizes the pose based on this image-level comparison (Glover et al., 2024; Kang et al., 2026; 2024; Hay et al., 2026; Nehvi et al., 2021). This render-and-compare strategy is natural for RGB cameras, where rendered and observed images share a similar appearance. However, with event cameras, the comparison is less direct, since the rendered object appearance does not resemble the sparse brightness-change measurements produced by the sensor. Existing event-based render-and-compare methods address this representation gap by constructing a representation shared by the rendered hypothesis and the event observation. For example, EDOPT (Glover et al., 2024) compares an event-derived edge representation with the edges of the rendered object. Beyond edges, other approaches use the pose velocity and a generative event model to simulate individual events (Nehvi et al., 2021) or brightness increments (Kang et al., 2024) from CAD-rendered frames. More recent methods use learned models to bridge the domain gap: Event6D (Kang et al., 2026) reconstructs intensity images from events to reduce their mismatch with CAD renderings, while AEE (Hay et al., 2026) learns a latent representation that aligns real event observations with synthetic renderings. Although these strategies enable image-level comparison between the two domains, their effectiveness depends on the quality and similarity of the intermediate representations. Variations in illumination, reflectance, motion, and sensor characteristics strongly affect event distributions, making object appearance difficult to predict and breaking the similarity between rendered and observed representations. These challenges motivate a formulation that directly exploits spatio-temporal event information, without relying on intermediate appearance-based representations of either the events or the rendering.

In this work, we retain the hypothesis-and-comparison principle of render-and-compare but replace event-to-CAD appearance matching with motion consistency. Given a candidate 6DoF pose trajectory, instead of rendering the CAD model over multiple frames, we render the image-plane motion field induced by the object geometry. Inspired by contrast maximization (Gallego et al., 2018), we warp the observed events according to this motion and score the candidate trajectory by the sharpness of the resulting Image of Warped Events (IWE). This moves the comparison from heterogeneous appearance representations to a shared geometric motion domain. Because the motion field is constrained by the known object geometry, the formulation directly optimizes the 6DoF pose update and supports continuous tracking of arbitrary rigid objects given only their 3D models. Figure 1 illustrates the difference between appearance-based render-and-compare methods and our motion-consistency formulation. In summary:

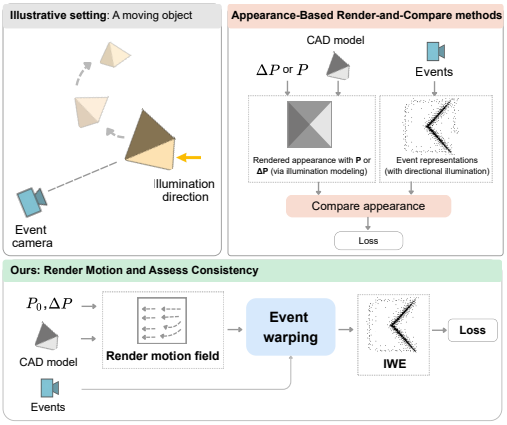


Figure 1: **Appearance consistency vs. motion consistency.** Existing render-and-compare methods rely on appearance-based representations that may suffer from representation mismatch. Our approach instead renders geometry-induced motion, warps the observed events, and directly evaluates motion consistency for 6DoF pose updates.

- We introduce a geometry-aware contrast maximization formulation for 6DoF object motion estimation that evaluates candidate pose trajectories from raw events by aligning them with geometry-induced image motion, without appearance representations or event-object correspondences.
- We integrate this formulation into a 6DoF tracking method that uses event pose updates, adapts the temporal window to object motion, and optionally uses depth for absolute pose correction.
- We evaluate the method in event-only and event–depth settings on two publicly available datasets, demonstrating strong accuracy, generalization across motion and object conditions, and robustness to appearance, event, and initialization perturbations.

2 Related Work

Monocular object pose estimation and tracking. Monocular 6D object pose methods can be broadly categorized into geometric model-based and learning-based approaches. Classical methods track edges, contours, keypoints, or image regions and estimate the pose by aligning these observa-

tions with a known 3D model (Choi & Christensen, 2010; Prisacariu & Reid, 2009; Tjaden et al., 2017). Correspondence-based methods establish 2D–3D point associations and recover the object pose using PnP or related geometric solvers (Lepetit et al., 2009; Rad & Lepetit, 2017; Peng et al., 2019; Chen et al., 2022). Learning-based methods either regress the pose directly, predict 2D–3D correspondences, or iteratively refine an initial estimate by comparing the observed image with a rendering of the object (Labbé et al., 2023; Li et al., 2020). For tracking, temporal methods propagate or refine the pose across consecutive frames (Wen et al., 2024). Although these approaches achieve strong performance, their accuracy can degrade under rapid motion because of motion blur and large inter-frame displacements.

Event-based object pose tracking. Given the sparse and asynchronous nature of event streams, many event-based 6DoF methods rely on geometric primitives or motion cues extracted from events. Some approaches directly associate event-derived edges or lines with known 3D object geometry and recover pose through geometrically constrained optimization (Liu et al., 2024; 2025). Their performance therefore depends on reliable feature extraction and data association, while the use of straight-line primitives can limit their applicability to objects with more complex or curved geometry. Liu et al. (2025), Wang et al. (2026), and Li et al. (2026) use event-based optical flow or tracked keypoints to establish correspondences or propagate pose estimates over time. In these approaches, errors in the intermediate motion or correspondence estimates can directly affect the recovered pose. Another line of work follows a render-and-compare formulation, where candidate object poses are evaluated by comparing model-derived hypotheses with representations constructed from the event stream. These methods differ mainly in the representation used to make the rendered model and event observations comparable, ranging from geometric templates to learned or reconstructed representations (Glover et al., 2024; Kang et al., 2024; 2026; Hay et al., 2026). While avoiding explicit 2D–3D correspondence estimation, they still rely on the quality of the intermediate representation used for pose comparison. In contrast, our formulation evaluates the geometry-induced motion of a candidate 6DoF trajectory through the spatio-temporal alignment of the observed events.

Contrast maximization. (CMax) was introduced as a general framework for estimating motion from events by searching for the motion parameters that maximize the alignment of motion-compensated events (Gallego et al., 2018). When the motion parameters are correct, events generated by the same scene structures align, producing a sharper IWE. Formulated as

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{C}\left(\mathrm{IWE}(\boldsymbol{\theta})\right), \tag{1}$$

where $\boldsymbol{\theta}$ parameterizes the event trajectories and $\mathcal{C}$ measures their alignment. Early CMax formulations applied this principle to several event-based estimation problems, including event-based optical flow, camera rotation, and scene depth (Gallego et al., 2018), followed by work investigating alternative contrast objectives (Stoffregen & Kleeman, 2019), optimization strategies (Liu et al., 2020), and failure modes such as event collapse (Shiba et al., 2022). More recent methods extend CMax to joint motion and geometry estimation. Shiba et al. (2024) jointly estimate depth and ego-motion, CMax-SLAM (Guo & Gallego, 2024) considers rotational motion in SLAM, and event-based 3D reconstruction methods use contrast maximization to refine camera motion and/or scene geometry (Liao et al., 2025; Kohyama et al., 2026). In this work, we are the first to exploit the CAD geometry to formulate a geometry-aware event warp induced by candidate 6DoF object motion.

3 Problem Statement

Given a known CAD model $\mathcal{M}$ and a calibrated camera, let $P_0 = P(t_0) \in SE(3)$ be the known initial object pose in the camera frame. For an event window $[t_0, t_1]$ of duration $\Delta t = t_1 - t_0$, our goal is to estimate the pose $P(t)$ for every $t \in [t_0, t_1]$.

Each event is represented by $e_i = (\mathbf{u}_i, t_i, p_i)$, where $\mathbf{u}_i$ is its image location, t_i its timestamp, and $p_i \in \{-1, +1\}$ its polarity. The event set $\mathcal{E} = \{e_i\}$ contains the events triggered during $[t_0, t_1]$ within an object-centered region of interest. This region is obtained by projecting $\mathcal{M}$ at pose P_0 and adding pixel padding to each side to account for the expected motion during the window.

Assuming that the time window $[t_0, t_1]$ is sufficiently short, we model the object motion using a constant body-frame twist. Let $\Delta t = t_1 - t_0$, and parameterize the motion over the window by the six-dimensional Lie-algebra coordinates:

$$\Delta\boldsymbol{\xi} = \begin{bmatrix}\Delta\boldsymbol{\omega}^\mathsf{T} & \Delta\mathbf{v}^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^6, \qquad P(t) = P_0 \exp\left(\frac{t - t_0}{\Delta t}\Delta\boldsymbol{\xi}^\wedge\right), \quad t \in [t_0, t_1]. \tag{2}$$

Here, $\Delta\boldsymbol{\omega}$ and $\Delta\mathbf{v}$ denote the rotational and translational components of the motion increment, respectively. More precisely, $\Delta\boldsymbol{\omega} \in \mathbb{R}^3$ and $\Delta\mathbf{v} \in \mathbb{R}^3$ are the rotational and translational coordinates of the motion increment. The hat operator $(\cdot)^\wedge : \mathbb{R}^6 \to \mathfrak{se}(3)$ maps these six coordinates to their matrix representation in the Lie algebra, $\boldsymbol{\xi} = \Delta\boldsymbol{\xi}/\Delta t$. This parameterization satisfies $P(t_0) = P_0$ and $P(t_1) = P_0 \exp(\Delta\boldsymbol{\xi}^\wedge)$.

Under the constant-twist assumption, estimating the pose update at any time $t \in [t_0, t_1]$ therefore reduces to estimating these six motion parameters. We formulate this as an optimization problem in which a loss function evaluates each candidate motion increment using the observed events, the known object geometry, and the initial pose as follows

$$\Delta\boldsymbol{\xi}^\star = \underset{\Delta\boldsymbol{\xi}\in\mathbb{R}^6}{\arg\min} \mathcal{L}(\Delta\boldsymbol{\xi}; \mathcal{E}, \mathcal{M}, P_0)\,. \tag{3}$$

The problem is thus to define an event-based loss $\mathcal{L}$ that provides a suitable optimization signal for recovering the motion increment $\Delta\boldsymbol{\xi}$. In the following section, we introduce our geometry-aware contrast loss and describe how we integrate it into 6DoF object pose tracking.

4 METHODOLOGY

In this section, we present ConTraPosE. We first introduce its core component, the geometry-aware contrast loss (Section 4.1), which evaluates a candidate 6DoF motion by constructing its induced image motion, warping the observed events accordingly, and measuring their alignment after warping. We then describe how this loss is integrated into a continuous 6DoF tracking framework, together with adaptive temporal windows and optional depth-based pose correction (Section 4.2).

4.1 GEOMETRY-AWARE CONTRAST LOSS

The proposed contrast loss evaluates the consistency between a candidate $SE(3)$ trajectory, parameterized by $\Delta\boldsymbol{\xi}$, and the observed events. Directly comparing the candidate pose trajectory with the events is not possible, since they belong to different domains: the trajectory is defined in $SE(3)$, whereas events lie in a spatio-temporal image-plane domain. We therefore use the known object geometry to express each candidate trajectory through the motion it induces over the projected object surface. This geometry-induced motion field provides the link between 3D rigid motion and the observed spatio-temporal event pattern. Conceptually, this mapping can be written as

$$\Delta\boldsymbol{\xi} \;\longrightarrow\; P(t) \in SE(3) \;\longrightarrow\; \mathbf{F}_{\Delta\boldsymbol{\xi}}(\mathbf{u}), \tag{4}$$

where $\mathbf{F}_{\Delta\boldsymbol{\xi}}(\mathbf{u})$ denotes the projected image-space motion induced by the candidate trajectory. This projected motion representation provides a common domain in which the candidate $SE(3)$ trajectory can be assessed against the observed events.

Object motion field. To construct the geometry-induced motion field $\mathbf{F}_{\Delta\boldsymbol{\xi}}$, the candidate trajectory is first sampled uniformly at $N+1$ poses as follows

$$P_j = P_0 \exp\left(\frac{j}{N}\Delta\boldsymbol{\xi}^\wedge\right), \qquad j = 0, \ldots, N, \tag{5}$$

where consecutive poses are separated by $\Delta t/N$, with $P_0 = P(t_0)$ and $P_N = P_0 \exp(\Delta\boldsymbol{\xi}^\wedge)$.

For each pair of poses (P_j, P_{j+1}), the projected image-plane displacement of a CAD mesh vertex $\mathbf{X}$ is

$$\mathbf{d}_j(\mathbf{X}) = \pi_K(P_{j+1}\mathbf{X}) - \pi_K(P_j\mathbf{X}), \qquad j = 0, \ldots, N-1, \tag{6}$$

where π_K is the pinhole projection with known camera intrinsics K.

To obtain a dense, differentiable motion field, we attach $\mathbf{d}_j(\mathbf{X})$ to each mesh vertex as an attribute and rasterize the CAD mesh geometry at pose P_j using *nvdiffrast* (Laine et al., 2020). Barycentric interpolation of the visible triangle attributes produces a dense displacement field $\mathbf{D}_j(\mathbf{u})$ over the projected object surface. Rasterization also yields a support map $S_j(\mathbf{u})$ indicating the corresponding object coverage.

The support map weights events associated with the object. Since events near the silhouette may enter or leave the projected support as the candidate pose changes, we apply anti-aliasing to S_j to obtain smooth boundary weights in $[0, 1]$. We do not apply it to $\mathbf{D}_j$, as this would attenuate motion at boundary pixels. Thus, boundary events retain the same induced motion while their contribution to the image of warped events varies smoothly. Repeating this over consecutive pose pairs produces N geometry-constrained displacement fields.

Each segment displacement $\mathbf{D}_j$ is then converted into a velocity over its sub-interval $\Delta t/N$ and fused using the projected object support S_j yielding

$$\mathbf{F}_{\Delta\boldsymbol{\xi}}(\mathbf{u}) = \frac{N \sum_{j=0}^{N-1} S_j(\mathbf{u})\, \mathbf{D}_j(\mathbf{u})}{\Delta t \left(\sum_{j=0}^{N-1} S_j(\mathbf{u}) + \varepsilon \right)}, \tag{7}$$

where $\varepsilon > 0$ ensures numerical stability. The field is zero at pixels outside the support of all sampled projections. This fused representation provides a compact approximation of the projected motion over the event window.

The resulting object motion field provides a direct way to evaluate how well a candidate 6DoF motion explains the observed events, by applying it to the events and measuring their alignment after motion compensation.

Event warping. Using the fused flow field, we assume constant image velocity over the event window and warp each event to a common reference time $t_r \in [t_0, t_1]$ as follows

$$\mathbf{u}'_i = \mathbf{u}_i + (t_r - t_i)\mathbf{F}_{\Delta\boldsymbol{\xi}}(\mathbf{u}_i). \tag{8}$$

Under the correct motion, events generated by the same object structures align at t_r. We use the temporal midpoint $t_r = (t_0 + t_1)/2$; multiple reference times may improve robustness at additional computational cost.

Each event is weighted by the antialiased object support of its temporal sub-interval indexed by j as follows

$$w_i = S_{\Delta\boldsymbol{\xi}}(\mathbf{u}_i, t_i), \qquad S_{\Delta\boldsymbol{\xi}}(\mathbf{u}, t) = S_j(\mathbf{u}) \quad \text{for} \quad t_j \le t < t_{j+1}, \quad j = 0, \ldots, N-1. \tag{9}$$

Events inside the projected object thus receive weights near one, while those near its silhouette are smoothly down-weighted. Warped events are accumulated into a signed IWE via bilinear splatting

$$I_{\Delta\boldsymbol{\xi}}(\mathbf{x}) = \sum_i w_i p_i B(\mathbf{x} - \mathbf{u}'_i)\,, \tag{10}$$

where B is the bilinear splatting kernel, $p_i \in \{-1, +1\}$ is the event polarity, and $\mathbf{x}$ denotes an IWE pixel within the object region of interest.

Rather than using the spatial variance of the IWE, we measure alignment with the following normalized sum of squares

$$C(\Delta\boldsymbol{\xi}) = \frac{\sum_{\mathbf{x}} I_{\Delta\boldsymbol{\xi}}(\mathbf{x})^2}{M_{\Delta\boldsymbol{\xi}}^2 + \varepsilon}, \qquad M_{\Delta\boldsymbol{\xi}} = \sum_{\mathbf{x}} \sum_i w_i B(\mathbf{x} - \mathbf{u}'_i)\,, \tag{11}$$

where $M_{\Delta\boldsymbol{\xi}}$ is the total accumulated unsigned event weight and $\varepsilon > 0$ ensures numerical stability. This normalization reduces sensitivity to variations in the number of selected events across temporal windows. The corresponding contrast loss is defined as the inverse alignment score and is given by

$$\mathcal{L}_c(\Delta\boldsymbol{\xi}) = 1 - C(\Delta\boldsymbol{\xi}), \tag{12}$$

where lower values indicate better alignment of the motion-compensated events.

4.2 6DoF Pose Tracking with Geometry-aware Contrast Loss

Using the geometry-aware contrast loss defined in Equation 12 within the optimization objective defined in Equation 3 allows us to solve for the pose increment $\Delta\boldsymbol{\xi}$ given an event stream, an initial pose estimate, and a known CAD model of the object. This objective can serve several purposes such as trajectory pose refinement, motion-hypothesis evaluation, or as a self-supervised signal for learning 6DoF motion. A natural application is 6DoF pose tracking, where it is solved repeatedly as new events arrive. At each tracking step, we estimate the object's motion by minimizing Equation 3 and use the resulting increment to update the current pose. To handle varying object motion, we accumulate events over an adaptive temporal window whose duration is set from the most recent motion estimate. The formulation also extends naturally to other sensing modalities, which can be added as optional loss terms in the objective. In our experiments, we add depth this way, as an auxiliary loss that can be turned on or off depending on whether depth is available.

Adaptive motion window. The choice of temporal window directly affects contrast-loss quality, as it controls the observable image-plane motion induced by the object. Shorter windows better preserve the local constant-twist assumption but may contain insufficient motion for a reliable contrast

signal, while longer windows provide more motion at the expense of this assumption. To address this trade-off, a lightweight adaptive-window module is introduced to adjust the temporal window at each tracking step before we update the pose with the contrast loss. The appropriate duration depends on the object dynamics, projected scale, and distance from the camera. We search for the window using recent pose history and object geometry, with the motion field in Equation 7 providing an image-space criterion for evaluating candidate durations. At each tracking step, we initially select the minimum duration $\Delta t_{\min}$. When two previous motion estimates are available, we form two hypotheses: one keeps the latest 6DoF velocity constant, while the other updates it using the recent change in velocity to account for acceleration. Over a short interval, these two predictions should remain similar; larger differences indicate stronger motion dynamics. Rather than comparing the two motion estimates directly in $SE(3)$, we compare their induced image-plane motion, the domain in which the event camera observes object motion. This also accounts for the effect of object geometry, depth, and camera projection on image motion. For a candidate duration Δt, both hypotheses are converted into image-plane motion fields, and their disagreement is measured as

$$E(\Delta t) = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{u} \in \mathcal{V}} \left\| \mathbf{F}_{\Delta t\, \boldsymbol{\xi}_k^{\text{const}}}(\mathbf{u}) - \mathbf{F}_{\Delta t\, \boldsymbol{\xi}_k^{\text{change}}}(\mathbf{u}) \right\|_2, \tag{13}$$

where $E(\Delta t)$ measures the average image-plane motion disagreement between the two hypotheses over the visible object region $\mathcal{V}$. Starting from $\Delta t_{\min}$, the candidate window is iteratively expanded over a small number of steps while $E(\Delta t) < \tau_{\text{model}}$, up to $\Delta t_{\max}$. A safety factor is then applied to account for uncertainty in the estimated motion. Stable motion therefore allows longer windows, whereas rapidly changing motion keeps the window short and better preserves the constant-twist assumption. If previous motion estimates are not yet available, $\Delta t_{\min}$ is retained.

Depth loss. The event-based contrast loss can independently estimate pose increments from the event stream. However, because each update is relative to the preceding pose estimate, small errors may accumulate over long sequences. When depth is available, we use it as an additional geometric constraint to refine the absolute pose and reduce drift. At the depth timestamp, the candidate trajectory is evaluated, and the CAD mesh is rendered to obtain a predicted depth map $\hat{D}_{\Delta\boldsymbol{\xi}}(\mathbf{u})$. Its agreement with the measured depth $D(\mathbf{u})$ is evaluated using

$$\mathcal{L}_{\text{depth}}(\Delta\boldsymbol{\xi}) = \frac{\sum_{\mathbf{u}} w(\mathbf{u})\, \rho(r_{\Delta\boldsymbol{\xi}}(\mathbf{u}))}{\sum_{\mathbf{u}} w(\mathbf{u}) + \varepsilon}. \tag{14}$$

where $w(\mathbf{u})$ is a soft confidence weight that suppresses measurements affected by occlusions and silhouette uncertainty, while ρ is a bounded robust penalty, and $r_{\Delta\xi}(\mathbf{u}) = D(\mathbf{u}) - \hat{D}_{\Delta\xi}(\mathbf{u})$. The complete optimization objective is

$$\mathcal{L}(\Delta\boldsymbol{\xi}) = \mathcal{L}_{\text{c}}(\Delta\boldsymbol{\xi}) + \lambda_{\text{depth}}\, \mathcal{L}_{\text{depth}}(\Delta\boldsymbol{\xi}), \tag{15}$$

where λ_{depth} controls the strength of depth refinement and is set to zero when depth is not available.

Pose optimization. For each event window, the integrated body-frame twist $\Delta\boldsymbol{\xi} \in \mathbb{R}^6$ is initialized at zero and optimized with Adam using the contrast objective in Equation 12, or the combined objective in Equation 15 when depth is available. The optimal increment $\Delta\boldsymbol{\xi}^\star$ is mapped to $SE(3)$ through the exponential map and multiplied with the initial pose P_k of the current window to obtain the endpoint pose P_{k+1}, which initializes the next window.

5 Experiments

5.1 Evaluation Settings

We evaluate tracking in two settings: a multimodal setting combining events with discrete depth, and an event-only setting. The multimodal setting follows the Event6D protocol (Kang et al., 2026): depth is available only at keyframes, while events are continuously available for intermediate pose updates. No re-initialization is applied, allowing us to evaluate long-term robustness and the ability of depth to limit accumulated drift. In the event-only setting, all pose updates use only events. Since drift accumulates faster without depth, we re-initialize all methods after a fixed number of pose updates to ensure a fair comparison. All methods are provided with the CAD model.

Datasets. Our main evaluation dataset, Event6D (Kang et al., 2026), covers diverse object shapes and high-speed motion. It provides real sequences of 14 objects with a continuous event stream, 120 Hz pose annotations, and 30 Hz depth, making it suitable for evaluating tracking under rapid motion over long horizons. We also evaluate on SPADES (Rathinam et al., 2024), a spacecraft

Table 1: Comparison of multi-rate event–depth 6D object pose tracking methods on Event6D. Inter-frame updates have no new depth measurement; keyframe updates use newly available depth.

Approach	**E2VID + FP**		**ETAP + PnP/ICP**		**FP**		**EventTrack6D**		**ConTraPosE**		**ConTraPosE (FP-refined)**	
Inter-frame (E)	FP + Last depth		PnP		FP + Last depth		FP + Pred. depth		$\mathcal{L}_c$		$\mathcal{L}_c$	
Keyframe (E+D)	FP + New depth		ICP + New depth		FP + New depth		FP + New depth		$\mathcal{L}_c + \lambda_d \mathcal{L}_d$		FP + New depth	
Metric	**ADD-S**	**ADD**	**ADD-S**	**ADD**	**ADD-S**	**ADD**	**ADD-S**	**ADD**	**ADD-S**	**ADD**	**ADD-S**	**ADD**
banana	24.92	9.65	6.09	1.54	13.99	1.70	43.46	16.26	55.00	8.40	28.77	14.98
bowl	46.20	0.57	48.20	1.25	2.70	0.38	54.86	0.34	44.24	4.36	60.90	3.49
cracker	36.65	28.42	1.17	1.04	45.53	27.28	74.65	62.44	72.41	54.78	75.49	65.19
drill	4.24	2.67	1.02	0.53	39.04	6.83	66.13	38.58	60.79	39.24	53.90	32.69
hammer	57.66	40.34	39.61	27.64	23.32	3.20	53.84	39.68	69.76	44.63	77.80	60.39
marker	0.17	0.17	0.17	0.17	12.33	1.02	62.36	28.01	46.38	16.59	69.27	31.54
mouse	25.48	3.31	0.13	0.13	8.60	0.81	34.54	4.93	13.64	0.62	30.73	4.94
mug	21.43	6.34	1.96	0.80	3.46	0.29	38.74	8.08	32.59	2.95	43.41	9.86
mustard	70.79	49.66	5.20	3.62	26.38	4.42	82.21	63.26	68.36	25.26	77.95	57.75
pitcher	72.35	43.29	8.44	5.12	58.15	17.66	71.47	42.86	51.75	23.40	73.97	46.59
scrub	76.67	54.53	1.21	1.01	61.01	22.19	84.55	65.97	84.93	65.68	85.78	67.34
spam	49.74	23.66	8.47	4.01	27.44	8.03	49.03	24.40	61.88	36.77	55.82	30.63
spatula	47.97	33.39	2.68	2.02	5.07	2.23	0.10	0.10	48.87	15.72	53.14	38.88
wine	0.18	0.18	0.18	0.18	16.66	0.94	17.87	0.48	34.00	0.90	53.44	1.91
MEAN	37.24	16.97	7.77	2.22	22.93	4.31	52.79	25.26	54.40	25.84	**60.51**	**35.12**

dataset with real and synthetic sequences. It features slower motion than Event6D, 10 Hz pose annotations, and events at close, mid, and far ranges (approximately 3–12 m). Given the short sequences, no re-initialization is applied. As SPADES provides no depth, we simulate it for the multimodal evaluation with SimSense (Li, 2022). The simulated depth is available at every tracking step, so all multimodal updates use both events and depth, without intermediate event-only updates.

Metrics. We report the Area Under Curve (AUC) of ADD and ADD-S using the standard threshold of $0.1d$, where d denotes the object diameter.

Baselines. We compare against state-of-the-art methods in both settings. We group them into render-and-compare methods, which match the rendered object appearance to the event observations, and correspondence-based methods, which recover the pose from tracked 2D keypoints.

Event-only. We include EDOPT (Glover et al., 2024), a render-and-compare method that matches edges rendered from the object model with edge representations extracted from events. As a correspondence-based baseline, we use ETAP (Hamann et al., 2025) to track 2D object keypoints in events and recover the 6DoF pose with PnP.

Event and depth. We evaluate the RGB-D render-and-compare tracker FoundationPose (FP) (Wen et al., 2024) in two variants: one fine-tuned on intensity frames reconstructed from events with E2VID (Rebecq et al., 2019) (E2VID + FP), and one fine-tuned directly on events (FP). Since FP requires depth at every update, both variants reuse the most recent depth frame until a new one becomes available. We also include EventTrack6D (Kang et al., 2026) which extends FP with a learned model that extrapolates depth between consecutive depth frames. Finally, we extend the correspondence-based ETAP baseline to the event–depth setting: keypoints are tracked between depth frames and used to recover pose with PnP, while at keyframes depth measurements refine the pose with ICP (Besl & McKay, 1992).

All methods are given the object CAD model and the same initial pose, and evaluation is conducted on the same tracking rate for all the methods.

5.2 Results on Event6D Dataset

Evaluation under the event and depth setting. We first evaluate ConTraPosE in the multi-rate setting of Event6D dataset, where events are continuously available for intermediate pose updates, and depth is provided only at keyframes. ConTraPosE uses the contrast loss $\mathcal{L}_c$ between depth observations and the combined event–depth loss when depth becomes available. As shown in Table 1, ConTraPosE outperforms the compared baselines, demonstrating effective pose propagation between sparse depth measurements. Both ConTraPosE variants achieve robust performance across the different Event6D objects, including thin and weakly textured objects such as the spatula and

wine bottle. This consistency across diverse shapes, appearances, and textures highlights the robustness of the geometry-based motion formulation. The FP-refined variant of our method further allows a direct comparison with EventTrack6D, since both methods use the same depth-based refinement at keyframes and differ mainly in how they propagate pose in the intermediate frames. The stronger performance of ConTraPosE in this setting therefore indicates that the proposed contrast-based event tracker more effectively bridges the pose between sparse depth keyframes using events alone.

Evaluation under the event-only setting. We further evaluate ConTraPosE using only the contrast loss $\mathcal{L}_c$, without any depth-based correction. This setting directly evaluates the ability of the proposed objective to propagate 6DoF pose from events alone. As shown in Table 2, ConTraPosE substantially outperforms both the correspondence-based ETAP+PnP and the render-and-compare EDOPT baselines. The improvement is consistent across objects with different geometries and appearances, with particularly large gains on low-texture objects such as the bowl (+44.1 ADD-S) and mug (+39.8 ADD-S) compared with EDOPT. This indicates that geometry-aware contrast provides a robust event-only pose-update signal without relying on appearance matching.

Table 2: Event-only tracking on Event6D.

	EDOPT		**ETAP+PnP**		**ConTraPosE**	
	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD
banana	22.18	9.01	21.02	13.56	60.41	40.96
bowl	12.62	3.32	12.81	8.17	56.67	40.78
cracker	40.95	28.38	31.80	19.15	69.88	55.37
drill	27.23	7.82	29.96	16.19	76.28	59.37
hammer	26.98	13.36	41.43	28.21	73.69	58.73
marker	12.15	6.38	19.46	8.66	58.35	36.34
mouse	6.16	1.66	16.03	7.26	31.52	16.30
mug	7.92	2.32	6.42	2.58	47.71	28.64
mustard	25.55	13.05	24.43	14.69	54.31	37.07
pitcher	47.95	21.88	31.64	16.26	65.62	43.71
scrub	26.65	12.22	23.89	11.05	49.00	25.89
spam	10.09	3.32	11.63	6.86	40.65	25.94
spatula	26.32	14.77	36.23	25.23	56.41	43.84
wine	18.30	5.45	20.94	8.91	45.37	22.78
Mean	25.11	11.44	24.38	13.85	**58.58**	**40.58**

5.3 Results on SPADES Dataset

To further demonstrate the generalization of ConTraPosE, we evaluate it on SPADES, which presents different object geometry and more moderate motion dynamics than Event6D. Pose estimates are evaluated at 10 Hz, making intermediate high-rate pose propagation not needed.

Evaluation under the event and depth setting. Here, simulated depth is available at every pose update, so ConTraPosE uses the combined event–depth objective throughout. As shown in Table 3, ConTraPosE achieves the strongest overall performance on both synthetic and real sequences. The far range is particularly challenging for depth-based methods: the object may be roughly 12 m from the simulated stereo sensor, increasing depth noise and artifacts. Despite this, ConTraPosE remains strong at far range, indicating that combining contrast and depth objectives reduces reliance on depth alone, whereas methods relying more heavily on depth degrade under these artifacts.

Table 3: Comparison on the SPADES synthetic and real datasets across close, mid, and far distance ranges in the event–depth (E+D) and event-only (E) settings. ADD and ADD-S report AUC at $0.1d$.

Input	**Approach**	**SPADES Synthetic**								**SPADES Real**							
		Close		**Mid**		**Far**		**Mean**		**Close**		**Mid**		**Far**		**Mean**	
		ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD
E+D	ETAP + ICP	62.15	54.58	49.67	43.56	45.95	40.43	52.59	46.19	43.20	38.83	65.79	60.75	35.89	30.62	48.29	43.40
	EventTrack6D	84.83	78.34	69.08	56.08	41.79	19.51	62.63	47.80	87.51	83.21	87.24	84.28	76.65	62.44	83.80	76.64
	ConTraPosE ($\mathcal{L}_c + \lambda_{\text{depth}}\mathcal{L}_d$)	89.26	88.82	80.99	78.59	91.70	91.38	**87.32**	**86.26**	70.71	67.84	89.11	83.52	96.38	95.43	**85.40**	**82.26**
E	ETAP + PnP	47.52	35.71	30.84	18.20	21.96	10.83	33.44	21.58	38.73	29.09	50.14	33.05	22.07	8.82	36.98	23.65
	EDOPT	82.36	73.69	61.27	51.95	55.47	35.77	65.83	52.45	29.40	20.12	49.36	15.22	27.20	5.14	40.93	14.18
	ConTraPosE ($\mathcal{L}_c$)	82.42	78.00	64.26	52.50	55.09	40.47	**67.26**	**56.99**	51.04	38.87	34.63	19.82	45.55	21.21	**43.74**	**26.63**

Evaluation under the event-only setting. We additionally evaluate ConTraPosE using only the contrast loss. Table 3 shows that ConTraPosE achieves the strongest performance, demonstrating that the proposed event-based objective generalizes beyond the fast-motion setting of Event6D.

5.4 Additional Experiments

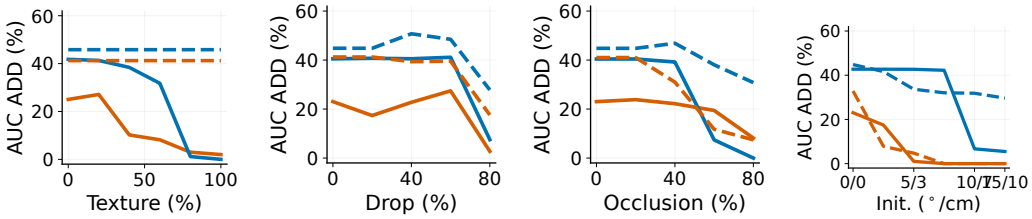
Robustness analysis. We test the robustness of the contrast-based pose update under three perturbation types: object-texture corruption, event-stream corruption, and initial-pose noise. We evaluate both variants of ConTraPosE, event-only ($\mathcal{L}_c$) and event–depth ($\mathcal{L}_c + \lambda_d \mathcal{L}_d$), and compare them with the render-and-compare baselines in the same setting: EDOPT (Glover et al., 2024) (event-only) and EventTrack6D (Kang et al., 2026) (event–depth).

Texture corruption. ConTraPosE remains largely unaffected by texture corruption (Figure 2a), since it does not rely on the rendered object appearance or texture. This makes it particularly suitable

for event cameras, which provide limited appearance information. In contrast, render-and-compare approaches can degrade when the available CAD appearance differs from the observed object; even edge-based matching can depend on texture-induced edges.

Event corruption. We randomly drop events, or remove all events inside a spatial region covering a given fraction of the projected object (spatial occlusion). ConTraPosE remains robust under moderate event loss and occlusion (Figures 2b and 2c). The contrast objective aggregates motion-consistent evidence over the whole visible object support, so the remaining events still constrain the motion. When depth is available, it further stabilizes tracking as the event signal becomes sparse.

Initial-pose noise. The event–depth variant tolerates much larger initialization errors than the event-only variant (Figure 2d). The contrast loss estimates motion relative to the current pose and cannot correct an initial offset on its own. Depth provides an absolute geometric constraint increasing the recovery range under initialization errors.



(a) Texture corruption. (b) Random event loss. (c) Spatial occlusion. (d) Initial pose noise.

Figure 2: Robustness to perturbations. Solid: baselines; dashed: ours. Orange: event-only; blue: event–depth.

Adaptive window. We evaluate the effect of adapting the event-window duration to the current motion dynamics. As shown in Table 4, using the adaptive window clearly improves both ADD and ADD-S. This confirms the importance of balancing sufficient event motion for a reliable contrast signal while keeping the window short enough to preserve the local constant-motion assumption.

Table 4: Adaptive event window.

Adaptive window	ADD	ADD-S
Without	14.28	41.82
With	**25.84**	**54.40**

Table 5: Forward compute time (ms) of the contrast loss.

Substeps	Motion field	Event warp	IWE+contrast	Total
1	2.67	0.94	1.34	**5.22**
2	4.36	0.70	1.03	**6.49**
3	6.36	0.64	1.06	**8.60**

Runtime analysis. We analyze the computational cost of the proposed event-based contrast loss using Event6D data. Table 5 reports the forward loss computation time for different numbers of interpolated poses; we use two interpolation steps for Event6D and three for SPADES. A single loss evaluation takes 5.22 ms with one step and 8.60 ms with three steps. Most of the runtime is spent rendering the motion field and increases with the number of interpolated poses. Event warping and the combined IWE construction and contrast computation each take approximately 1 ms. The appendix provides further details on the complete tracking runtime and optimization strategies.

6 Conclusion, Limitations, and Future Work

We introduced ConTraPosE, a geometry-aware contrast formulation for directly evaluating 6DoF object motion from raw events. By expressing candidate pose trajectories through their geometry-induced image motion and measuring the alignment of motion-compensated events, the proposed loss provides a direct pose-update signal without relying on appearance-based representations or explicit correspondences. Our experiments demonstrate accurate tracking across different objects and motion regimes, together with robustness to appearance changes, event corruption, occlusion, and initialization perturbations.

A current limitation of the tracking method is that pose updates rely on iterative gradient-based optimization, which adds computational cost despite efficient loss evaluation. Beyond tracking, the contrast loss defines a general geometry-aware objective for evaluating 6DoF motion hypotheses from events and could provide a self-supervised signal for training event-based pose or motion models without pose annotations. Another promising direction is to integrate the objective with learned optimization, where a network predicts or refines pose updates using the contrast loss as an optimization objective or input signal. Such a model could amortize iterative optimization into one or a few forward passes, enabling faster pose updates while retaining the geometric constraints of the proposed formulation.

AI USE STATEMENT

Generative AI tools were used to assist with writing and polishing the manuscript, as well as with code development and debugging. All scientific ideas, methodological choices, experimental design, results, analysis, and conclusions were developed, reviewed, and verified by the authors, who take full responsibility for the content of the paper.

REFERENCES

Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. doi: 10.1109/34.121791.

Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2771–2780, 2022. doi: 10.1109/CVPR52688.2022.00280.

Changhyun Choi and Henrik I. Christensen. Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. In *2010 IEEE International Conference on Robotics and Automation*, pp. 4048–4055, 2010.

Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 3867–3876. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00407. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Gallego_A_Unifying_Contrast_CVPR_2018_paper.html.

Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi: 10.1109/TPAMI.2020.3008413.

Arren Glover, Luna Gava, Zhichao Li, and Chiara Bartolozzi. EDOPT: event-camera 6-dof dynamic object pose tracking. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 18200–18206. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611511. URL https://doi.org/10.1109/ICRA57147.2024.10611511.

Jian Guan, Yingming Hao, Qingxiao Wu, et al. A survey of 6dof object pose estimation methods for different application scenarios. *Sensors*, 24(4):1076, 2024. doi: 10.3390/s24041076.

Shuang Guo and Guillermo Gallego. CMax-SLAM: Event-based rotational-motion bundle adjustment and SLAM system using contrast maximization. *IEEE Transactions on Robotics*, 40:2442–2461, 2024. doi: 10.1109/TRO.2024.3378443.

Friedhelm Hamann, Daniel Gehrig, Filbert Febryanto, Kostas Daniilidis, and Guillermo Gallego. ETAP: event-based tracking of any point. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 27186–27196. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02532. URL https://openaccess.thecvf.com/content/CVPR2025/html/Hamann_ETAP_Event-based_Tracking_of_Any_Point_CVPR_2025_paper.html.

Oussama Abdul Hay, Xiaoqian Huang, Muhammad Ahmed Humais, Abdulla Ayyad, Randa Almadhoun, and Yahya H. Zweiri. Learning 6d object pose estimation with event cameras using synthetic data and domain randomization. *IEEE Robotics Autom. Lett.*, 11(2):1690–1697, 2026. doi: 10.1109/LRA.2025.3644151. URL https://doi.org/10.1109/LRA.2025.3644151.

Jae-Young Kang, Hoonehee Cho, Taeyeop Lee, Minjun Kang, Bowen Wen, Youngho Kim, and Kuk-Jin Yoon. Event6d: Event-based novel object 6d pose tracking. *CoRR*, abs/2603.28045, 2026. doi: 10.48550/ARXIV.2603.28045. URL https://doi.org/10.48550/arXiv.2603.28045.

Yufan Kang, Guillaume Caron, Ryoichi Ishikawa, Adrien Escande, Kevin Chappellet, Ryusuke Sagawa, and Takeshi Oishi. Direct 3d model-based object tracking with event camera by motion interpolation. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 2645–2651. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611576. URL https://doi.org/10.1109/ICRA57147.2024.10611576.

Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, volume 9910 of *Lecture Notes in Computer Science*, pp. 349–364. Springer, 2016. doi: 10.1007/978-3-319-46466-4_21. URL https://doi.org/10.1007/978-3-319-46466-4_21.

Kai Kohyama, Yoshimitsu Aoki, Guillermo Gallego, and Shintaro Shiba. Geometric-photometric event-based 3d gaussian ray tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22301–22311, June 2026.

Alexander Krull, Frank Michel, Eric Brachmann, Stefan Gumhold, Stephan Ihrke, and Carsten Rother. 6-dof model based tracking via object coordinate regression. In *Computer Vision – ACCV 2014*, pp. 384–399, 2014.

Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6d pose estimation of novel objects via render & compare. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 715–725. PMLR, 2023.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.*, 39(6): 194:1–194:14, 2020. doi: 10.1145/3414685.3417861. URL https://doi.org/10.1145/3414685.3417861.

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009.

Ang Li. SimSense: A real-time depth sensor simulator. https://github.com/angli66/simsense, 2022. GitHub repository.

Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. *Int. J. Comput. Vis.*, 128(3):657–678, 2020. doi: 10.1007/S11263-019-01250-9. URL https://doi.org/10.1007/s11263-019-01250-9.

Zhichao Li, Chiara Bartolozzi, Lorenzo Natale, and Arren Glover. Event-based motion & appearance fusion for 6d object pose tracking. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Bohao Liao, Wei Zhai, Zengyu Wan, Zhixin Cheng, Wenfei Yang, Yang Cao, Tianzhu Zhang, and Zheng-Jun Zha. EF-3DGS: Event-aided free-trajectory 3d gaussian splatting. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2890.

Daqi Liu, Álvaro Parra Bustos, and Tat-Jun Chin. Globally optimal contrast maximisation for event-based motion estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 6348–6357. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00638. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Liu_Globally_Optimal_Contrast_Maximisation_for_Event-Based_Motion_Estimation_CVPR_2020_paper.html.

Zibin Liu, Banglei Guan, Yang Shang, Qifeng Yu, and Laurent Kneip. Line-based 6-dof object pose estimation and tracking with an event camera. *IEEE Trans. Image Process.*, 33:4765–4780, 2024. doi: 10.1109/TIP.2024.3445736. URL https://doi.org/10.1109/TIP.2024.3445736.

Zibin Liu, Banglei Guan, Yang Shang, Shunkun Liang, Zhenbao Yu, and Qifeng Yu. Optical flow-guided 6dof object pose tracking with an event camera. *CoRR*, abs/2512.21053, 2025. doi: 10.48550/ARXIV.2512.21053. URL https://doi.org/10.48550/arXiv.2512.21053.

Zibin Liu, Shunkun Liang, Banglei Guan, Yang Shang, Qifeng Yu, and Ji Zhao. A geometric framework for absolute pose and velocity estimation with event cameras. *IEEE Trans. Image Process.*, 35:6714–6729, 2026. doi: 10.1109/TIP.2026.3700937. URL https://doi.org/10.1109/TIP.2026.3700937.

Jalees Nehvi, Vladislav Golyanik, Franziska Mueller, Hans-Peter Seidel, Mohamed A. Elgharib, and Christian Theobalt. Differentiable event stream simulator for non-rigid 3d tracking. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pp. 1302–1311. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPRW53098.2021.00143. URL https://openaccess.thecvf.com/content/CVPR2021W/EventVision/html/Nehvi_Differentiable_Event_Stream_Simulator_for_Non-Rigid_3D_Tracking_CVPRW_2021_paper.html.

Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, and Michele Grassi. A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations. *Progress in Aerospace Sciences*, 93:53–72, 2017. doi: 10.1016/j.paerosci.2017.07.001.

Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4561–4570, 2019. doi: 10.1109/CVPR.2019.00469.

Chiara Plizzari, Mirco Planamente, Gabriele Goletto, Marco Cannici, Emanuele Gusso, Matteo Matteucci, and Barbara Caputo. E^2(go)motion: Motion augmented event stream for egocentric action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 19903–19915. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01931. URL https://doi.org/10.1109/CVPR52688.2022.01931.

Victor Adrian Prisacariu and Ian D. Reid. Pwp3d: Real-time segmentation and tracking of 3d objects. In *British Machine Vision Conference*, pp. 1–10, 2009.

Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *IEEE International Conference on Computer Vision*, pp. 3848–3856, 2017. doi: 10.1109/ICCV.2017.413.

Arunkumar Rathinam, Haytam Qadadri, and Djamila Aouada. SPADES: A realistic spacecraft pose estimation dataset using event sensing. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 11760–11766. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611231. URL https://doi.org/10.1109/ICRA57147.2024.10611231.

Henri Rebecq, Timo Horstschaefer, Guillermo Gallego, and Davide Scaramuzza. EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics Autom. Lett.*, 2(2):593–600, 2017. doi: 10.1109/LRA.2016.2645143. URL https://doi.org/10.1109/LRA.2016.2645143.

Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 3857–3866. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00398. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Rebecq_Events-To-Video_Bringing_Modern_Computer_Vision_to_Event_Cameras_CVPR_2019_paper.html.

Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Event collapse in contrast maximization frameworks. *Sensors*, 22(14), 2022. doi: 10.3390/s22145190.

Shintaro Shiba, Yannick Klose, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow, depth and ego-motion estimation by contrast maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7742–7759, 2024. doi: 10.1109/TPAMI.2024.3396116.

Timo Stoffregen and Lindsay Kleeman. Event cameras, contrast maximization and reward functions: An analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12292–12300, 2019. doi: 10.1109/CVPR.2019.01258.

Henning Tjaden, Ulrich Schwanecke, and Elmar Schömer. Real-time monocular pose estimation of 3d objects using temporally consistent local color histograms. In *IEEE International Conference on Computer Vision*, pp. 124–132, 2017.

Zhe Wang, Qijin Song, Zihao Li, Jingyu Xiao, and Weibang Bai. Keypoint-based dynamic object 6-dof pose tracking via event camera, 2026. URL https://arxiv.org/abs/2604.23387.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 17868–17879. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01692. URL https://doi.org/10.1109/CVPR52733.2024.01692.

A Appendix

A.1 Additional Implementation Details

This section complements the method description in the main paper with implementation choices that are useful for reproducibility. We focus on the construction of the motion fields, the numerical implementation of the losses, and the optimization procedure.

A.1.1 Tracking Pipeline

Learning rate and temporal interpolation. The optimization step size depends on the image-plane motion induced by changes in the $\mathrm{SE}(3)$ motion parameters. The same pose perturbation can produce different projected displacements depending on the object depth, geometry, camera focal length, and current pose. We use a smaller learning rate when small pose variations produce large image displacements, which avoids unstable optimization. Conversely, a larger learning rate improves convergence when the projected motion is weak.

Each optimization is initialized with zero incremental motion. Consequently, the first IWE corresponds to the original unwarped event window, while the object pose itself is initialized from the estimate of the preceding tracking update.

The number of temporal interpolation steps is selected per dataset according to the motion accumulated within its adaptive event windows. For highly dynamic Event6D sequences, the adaptive windows are typically short and two interpolation substeps are sufficient. For smoother SPADES trajectories, the window can expand while preserving the constant-twist assumption and may contain a larger accumulated image displacement. We therefore use three interpolation substeps on SPADES to represent the motion field over these longer windows.

Loss reference. The contrast loss can be evaluated at one or several temporal reference points. Multiple references, such as the beginning, midpoint, and end of the event window, produce several motion-compensated IWEs and can increase robustness by evaluating the same motion at different points along the trajectory. However, their computational cost grows with the number of reference IWEs. We use a single temporal midpoint reference in all tracking experiments as a compromise between robustness and efficiency.

A.1.2 Differentiable Motion-Field Rendering

We implement the mesh projection and motion-field construction with `nvdiffrast`. The CAD vertices are transformed by the interpolated object poses and projected using the calibrated camera intrinsics. Rasterization provides the visible triangle index, barycentric coordinates, depth, and object support at each image location. Per-vertex image displacements between consecutive interpolated poses are then interpolated over the visible triangles to obtain dense motion fields.

The CAD geometry and face tensors are stored on the GPU and reused throughout tracking. When the beginning of the event window remains fixed during an optimization, its rasterization is cached and reused across optimizer iterations. Candidate-dependent endpoint rasterizations are recomputed as the pose parameters change. For the discrete solver, independent candidate poses are rendered as an instanced batch whenever possible.

Temporal fusion of the motion field. The motion field is built by interpolating the trajectory in $SE(3)$ and mapping the resulting 3D displacements to the image plane.

$$\mathbf{F}_j(\mathbf{u}) = \frac{\mathbf{D}_j(\mathbf{u})}{\Delta t/N}, \qquad j = 0, \dots, N-1. \tag{16}$$

Keeping these fields separate gives the piecewise time-dependent flow

$$\mathbf{F}_{\Delta\boldsymbol{\xi}}(\mathbf{u}, t) = \mathbf{F}_j(\mathbf{u}), \quad t \in \left[t_0 + \frac{j\Delta t}{N}, t_0 + \frac{(j+1)\Delta t}{N} \right), \quad j = 0, \dots, N-1. \tag{17}$$

Events are therefore warped using the field associated with their temporal sub-interval, allowing the projected motion to vary along the trajectory.

In our experiments, we fused the segment-wise fields into a single average flow:

$$\mathbf{F}_{\Delta\boldsymbol{\xi}}(\mathbf{u}) = \frac{\sum_{j=0}^{N-1} S_j(\mathbf{u})\,\mathbf{F}_j(\mathbf{u})}{\sum_{j=0}^{N-1} S_j(\mathbf{u}) + \varepsilon}, \tag{18}$$

where $\varepsilon > 0$ ensures numerical stability. The field is zero at pixels outside the support of all sampled projections.

The fused representation is simpler and more efficient, but it may neglect the temporal variations caused by the nonlinear perspective projection. This approximation is generally accurate for the short windows considered here, where projected displacements remain small. For larger rotations, depth changes, or image displacements, the piecewise formulation better preserves the time-varying motion.

A.1.3 Contrast-Loss Implementation

All contrast computations are performed in `float32` on the GPU. The fused motion field is sampled once at the event coordinates and reused for the selected temporal reference. Warped events are converted to coordinates local to the rendered region of interest before IWE construction, reducing the number of pixels involved in the loss. We use differentiable bilinear splatting to accumulate the warped events. The four neighboring pixel contributions are computed in parallel and accumulated using indexed scatter operations.

A.1.4 Depth-Loss Implementation

For each available depth frame, we compare the observed depth with the CAD depth rendered at the candidate pose. Since the rendered depth map is already computed during motion-field construction, we reuse it without an additional rendering step.

We analyze the depth differences to identify the main group of pixels that is consistent with the object surface. Pixels likely belonging to foreground occluders, the background, or uncertain object boundaries receive lower weights. The depth loss is computed only from sufficiently reliable object pixels and is disabled when the available depth support is insufficient.

The responsibility estimates are detached before evaluating the differentiable depth residual. This prevents the optimizer from improving the loss by changing the pixel classification rather than correcting the pose. The resulting fixed responsibilities are cached and reused for several optimizer iterations before being refreshed. The depth residual itself remains differentiable with respect to the rendered pose.

We apply a bounded robust penalty to the selected object residuals. The depth term is used only when the effective number of object pixels is sufficiently large, and the dominant residual mode is coherent. Otherwise, the depth term is disabled for that update and tracking continues using the event contrast objective. This reliability check is particularly important for frames containing strong foreground occlusion or limited object-depth coverage.

A.1.5 Optimization

Gradient-based optimization The continuous tracker optimizes an incremental six-dimensional motion with Adam. Rotation is represented in radians and translation in meters. The increment is initialized to zero for every event window, and the optimized transformation is composed with the propagated object pose only after the current optimization terminates.

When multiple objectives are active, their numerical magnitudes alone do not provide a meaningful balance because rotation and translation have different units. We normalize each active loss using its initial gradient norm. For this normalization, the translation-gradient components are scaled by the RMS radius of the CAD model, placing them on a scale comparable to rotational motion. The configured loss weights are applied after this normalization. This is used only to balance the optimization gradients and does not change the physical pose parameterization.

The optimization supports a fixed iteration budget and optional early termination when the objective stops improving. The implementation avoids plotting, video generation, and metric computation inside the timed optimization loop. These operations are performed after tracking and are excluded from the reported pose-update runtime.

The discrete solver uses the same rendering, support, warping, and contrast implementations. It replaces the backward pass and Adam update with batched evaluations of local pose candidates. Candidates within a coordinate poll are evaluated simultaneously, while coarse-to-fine levels remain sequential because each level is centered on the pose selected by the preceding level.

Discrete pose search. We additionally consider a gradient-free discrete solver. Instead of computing derivatives of the contrast objective, the solver searches directly over local perturbations of the six-dimensional pose. At each coordinate poll, it evaluates the current pose together with positive and negative perturbations along the three rotational and three translational axes. This gives 13 candidates per poll. Events are warped according to each candidate motion, and the admissible candidate with the highest contrast is selected; candidates with insufficient projected-object support are rejected.

The search uses three coarse-to-fine levels, with one coordinate poll at each level. The perturbation magnitude is reduced after every level, and the next poll is centered on the pose selected at the preceding level. Candidates within one poll are independent and are rendered and scored in one GPU batch. The levels remain sequential because each level depends on the result of the previous one. Including the initial objective evaluation, one pose update therefore uses $1 + 3 \times 13 = 40$ candidate loss evaluations in four sequential forward calls with batch sizes 1, 13, 13, and 13.

A.2 Data processing: Depth simulation

We generated depth frames using SAPIEN's SimSense stereo sensor model (Li, 2022). For each timestamp, the object CAD model was rendered at its ground-truth pose into a calibrated stereo infrared pair with a $5\,\mathrm{cm}$ baseline. SimSense applied projected-pattern illumination, infrared noise, and stereo matching to obtain depth from disparity. Invalid measurements were set to zero, and valid depths were quantized to millimetres and stored as 16-bit PNG images. The rendering geometry was verified against the available ground-truth depth maps rendered directly from the object pose.

A.3 Detailed Runtime Analysis

Runtime protocol. The complete tracking runtime depends on the selected optimization strategy, the number of optimizer iterations, and the number of interpolated poses used to approximate the object motion. It also depends on whether additional objectives, such as the depth loss, are enabled. We therefore report the contrast loss and optimization costs separately.

The measurements use an NVIDIA RTX A6000 GPU and two motion-interpolation substeps. Unless specified otherwise, only the event-based contrast objective is evaluated. Data loading, metric computation, visualization, and file writing are excluded.

Gradient-based optimization runtime. Our continuous tracker minimizes the contrast objective using Adam. Each iteration consists of a forward loss evaluation, backpropagation, and an Adam parameter update. Table 6 gives the corresponding breakdown. Most of the computation is divided between motion-field construction in the forward pass and gradient computation in the backward pass. The Adam parameter update itself is negligible.

The optimizer generally converges within approximately 160 iterations. At $11.14\,\mathrm{ms}$ per iteration, this corresponds to approximately $1.78\,\mathrm{s}$ of optimization per pose update. Early stopping can reduce this cost when convergence is reached sooner. For example, approximately 86 iterations correspond to about $0.96\,\mathrm{s}$.

Discrete-search runtime. Our discrete search consists of four sequential search stages. At each stage, all candidate motions are evaluated in parallel using a batched implementation that renders their motion fields, warps the events, constructs the IWEs, and computes the contrast scores. The four search stages together require approximately $0.07\mathrm{s}$ per pose update. Event selection, mask

Table 6: Runtime breakdown of one Adam iteration using two interpolation substeps.

Operation	**Runtime (ms)**
Motion-field construction	4.36
Event warping	0.70
IWE and contrast computation	1.03
Other operations	0.41
Complete forward loss	6.49
Backward pass	4.42
Adam parameter update	0.09
Complete Adam iteration	**11.14**

Table 7: Optimization runtime per pose update on an NVIDIA RTX A6000. The discrete runtime is hardware-normalized from the measured full-dataset component timings. A slash-separated batch size lists the candidates evaluated simultaneously in each sequential forward call.

Solver	**Loss evaluations**	**Sequential calls**	**Candidates per call**	**Backward pass**	**Pose update (s)**
Adam	160	160	1	Yes	1.78
Discrete	40	4	1/13/13/13	No	≈ 0.09

construction, and pose update operations add approximately 0.02s, resulting in a total runtime of 0.09s per pose update. Since the search relies only on loss evaluation, no backward pass is required.

The number of objective evaluations has a different computational meaning for the two solvers. Adam performs one forward and one backward pass at every iteration, and all 160 iterations are sequential. The discrete solver evaluates the independent candidates of each coordinate poll simultaneously and performs only four sequential forward calls.

Tracking accuracy. Figure 3 compares the solvers in the event-only setting with the same periodic-redetection protocol. Adam obtains ADD and ADD-S AUC values of 40.58 and 58.58, respectively. The lightweight discrete solver obtains 17.21 and 34.36 on the complete Event6D test set. These results expose the accuracy–runtime trade-off: discrete search removes the backward pass and substantially reduces update time, while continuous optimization achieves higher tracking accuracy.

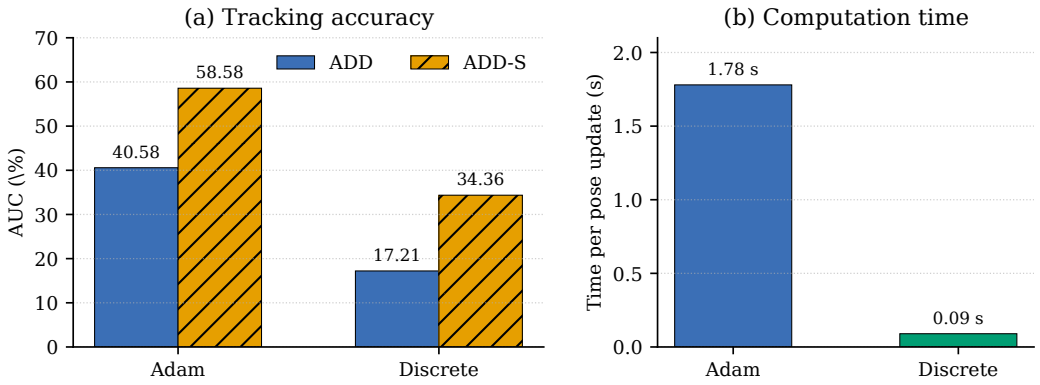


Figure 3: Continuous Adam optimization and discrete search in the Event6D dataset, event-only setting. Left: ADD and ADD-S AUC. Right: computation time per pose update.