Direct 3D model-based object tracking with event camera by motion interpolation

Y. Kang[1,2], G. Caron[1,3], R. Ishikawa[2], A. Escande[4], K. Chappellet[1,5], R. Sagawa[1], T. Oishi[2]

***Abstract*— Event cameras are recent sensors that measure intensity changes in each pixel asynchronously. It is being used due to lower latency and higher temporal resolution compared to traditional frame-based camera. We propose a method of 3D model-based object tracking directly from events captured by event camera. To enable reliable and accurate tracking of objects, we use a new event representation and predict brightness increment images with motion interpolation. Results of object tracking show the new methods significantly improves tracking duration and robustness, both for perspective and fisheye cameras. Our implementation succeeds in tracking objects when the camera speed is reaching 2 m/s.**

I. Introduction

Event cameras are bio-inspired vision sensors that do not capture frames of color (RGB) pixels but instead measure pixel brightness changes asynchronously [1]. Compared to frame-based cameras, their latency and power consumption are much lower, whereas their dynamic range and temporal resolution are higher. Though recent, they have made successful the fast visual control of robot flight [2] and are gaining interest for autonomous vehicles [3], visual odometry [4], 3D reconstruction [5] and, with additional sensors, 3D model-based tracking [6] and Simultaneous Localization And Mapping (SLAM) [7]. The recent efforts of the scientific community in sharing comprehensive datasets [8] develop the adoption of event cameras.

This work studies the task of high-speed 6 degrees of freedom (DoF) object tracking problem with event cameras, which is essential for real-time object manipulation with robots, too. We deal with a challenging scenario of tracking objects in first person view. To this end, we leverage [9] to solve three interleaved challenges. First, when generating brightness increment images by accumulating events, motion blur is unavoidable in scenes where depth is various. Indeed, several events at different image coordinates accumulated in the same brightness increment image can be triggered by the same 3D point. However, this accumulation blur was never considered in predicting brightness increment images, decreasing the tracking accuracy and success rate ([9] and Sec. II). Although recent works [10] generate brightness increments by weighting to avoid motion blur, such approaches are at a cost of information loss.

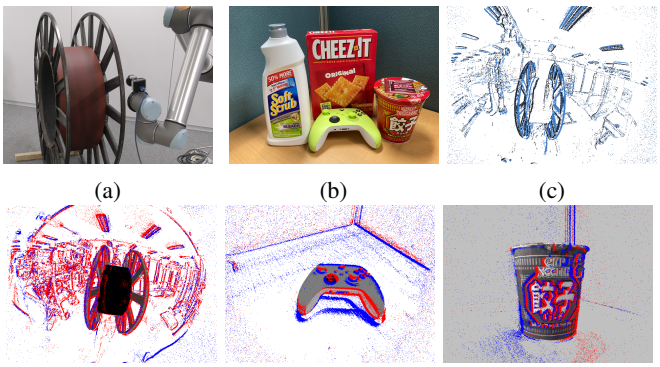


Fig. 1: (a) Event camera with a fisheye lens on the robot wrist. (b) Data capturing with UR10 robot. (c) Events captured with a fisheye lens of a 130 cm diameter coil to track. (d) - (f) Events plot over rendered intensity images of 3 target objects. Blue and red points represent positive and negative events respectively.

Second, we consider the 3D tracking of an object within the camera field-of-view. Basically, it implies the camera field-of-view might not always be filled of information, contrary to the camera tracking inside a room. A more important difference is that we can not predict the events accurately since the background and the light condition are unknown. The event polarities can be very different when the same object moves in different background or when the environment changes from bright to dark. Therefore, the brightness increment image predicted from the 3D model and the poses will not align well with the observed image.

Third, besides tracking moving objects while the camera is static, we also deal with a more challenging problem where the camera moves around the objects. In this case there can be a lot of events triggered by the background instead of the target object.

To solve these challenges, we propose some new methods based on a 3D-2D registration framework. We not only use a new event presentation of absolute brightness increments but also introduce motion interpolation to the prediction of brightness increment images from 3D model.

In summary, we make the following contributions in this paper:

- A novel method for 6-DoF object pose tracking which

[1]Y. Kang, G. Caron, K. Chappellet, R. Sagawa are with CNRS-AIST JRL (Joint Robotics Laboratory), IRL, and the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, 305-8568, Japan. {kang.yufan, ryusuke.sagawa}@aist.go.jp

[2]Y. Kang, R. Ishikawa, T. Oishi are with the Institute of Industrial Science, the University of Tokyo, Tokyo 153-8505, Japan. {ishikawa, oishi}@cvl.iis.u-tokyo.ac.jp

[3]G. Caron is also with Université de Picardie Jules Verne, MIS laboratory, 80000 Amiens, France. guillaume.caron@u-picardie.fr

[4]A. Escande is with Inria Center at Grenoble Alpes University, 38334 Montbonnot Cedex, France. adrien.escande@inria.fr

[5]K. Chappellet is also with Université de Montpellier, France. chappellet.kevin@gmail.com

takes only events and 3D model as input. To solve the problem of accumulation blur, this method leverages motion interpolation in the prediction of brightness increment image to simulate accumulation blur.
- We use a new event generation model which considers absolute brightness increments. We show this new representation is necessary for successful object tracking especially when the background is not pure color.
- We use a region of interest (ROI) from rendering module to generate new event frames and exclude the irrelevant events.

To the best of our knowledge, we are the first to track object poses from events when camera moves, where events are also triggered by the background. We present a compelling evaluation of the proposed method on real data on a new object tracking dataset. We show that our system reach accurate and stable tracking in five different objects with unified camera model.

Outline: The rest of the paper is organized as follows. Section II reviews the related work on event-based object tracking. Section III describes our object tracking methods. Section IV evaluates the method on a new object tracking dataset for first person view tracking. Finally, Section V conludes the paper.

II. Related work

Event-based ego-motion estimation has been developing step-by-step in scenarios during the past years [1]. Started from pure rotation [11], [12] or planar motion [13], [14], state-of-the-art methods are able to estimate 6-DoF camera pose in real world scenes where the photometric maps are given. Existing works estimate the camera poses either by event-to-event processing or by creating grids from multiple events that are spatially or temporally related. They can also be categorized according to whether intensity images are used to help with the estimation. Here we review the works related to the problem of event-based object tracking in recent years.

1) Event-based Camera Tracking: Tracking the camera pose in a scene where the photometric map is given can be seen as inside-object tracking. [15] was the first work to track 3-DoF object rotation in a panoramic setting. The tracking was formulated as an optimization problem given the panoramic map. Gallego *et al.* [16] proposed a probabilistic method to estimate the 6-DoF pose of camera in a indoor scene where the photometric map is given. They estimate the camera pose with a robust likelihood function upon the arrival of each single event. [17], [18] were also event-by-event methods that dealt with event-based pose tracking problem in Simultaneous Localization And Mapping (SLAM) systems. [17] estimates the relative motion from events and low-frequency grayscale frames of Dynamic Vision Sensors (DVS) by defining the likehood that an event can be observed given a certain motion. [18] estimates the camera motion with an Extended Kalman Filter (EKF). The main advantage of event-by-event methods is extreme high time resolution which could easily reach 100 MHz.

Most state-of-the-art works convert groups of events into grids for pose estimation, which is usually at cost of some information loss. Bryner *et al.* [9] proposed a direct brightness increment alignment approach to track a camera in scenes where the photometric appearance is provided. The 6-DoF poses and velocities are estimated by comparing two intensity variation images, one accumulated from the events and the other rendered by the photometric map and motion parameters. [10] combined the alignment with a photometric bundle adjustment back-end to minimize brightness errors. It also tried to avoid accumulation blur in event frame generation by emphasizing the central part of the window of events. Intensity images are used as keyframes to predict the intensity variation from the motion. [19] tackles the problem of event-based visual odometry with the *Time Surface* representation where each pixel stores the timestamp of the last event at that pixel. The inherent distance field nature of *Time Surface* is leveraged in the 3D-2D registration based camera tracking.

2) Event-based Object Tracking: Event-based object tracking requires the ability to discriminate the target objects with respect to the background. Ramesh *et al.* [20] designed a framework with a combination of tracking and detection to track object positions in camera field of view. They use an event-based *local sliding window* technique to deal with the cluttered and textured background. Similarly, Jiang *et al.* [21] combined an offline-trained detector with an online-trained tracking which complement each other to track multiple objects in the scene like vehicles. A particle filter method was used to track a moving ball while the event camera also moves [22]. Works of event-based 6-DoF object tracking combine event camera with other cameras [6], [23]. Dubeau *et al.* [6] proposed a deep learning based framework to handle high speed object tracking by combining an existing RGB-D network with a novel event-based network in a cascade way. Li and Stueckler [23] derived a probabilistic generative model from high rate events and refined the object trajectory in slower rate image frames through direct image alignment. However, they both deal with the easier case where the camera is static and the target object moves, indicating most events are related to the object itself.

3) Wrap-up: There are only a few works solving the problem of 6-DoF object tracking with event camera, most of which also use the information from other sensors like RGBD cameras. What's more, the only two existing works of event-based 6-DoF object tracking dealt with the easier case of static camera, where most events are triggered by the moving objects. Different from the existing works, we tackle the 6-DoF object tracking using only events and the 3D model as input, making our approach applicable to various scenes and tasks. What's more, we tackle the object tracking in noisy background where plenty of irrelevant events are triggered.

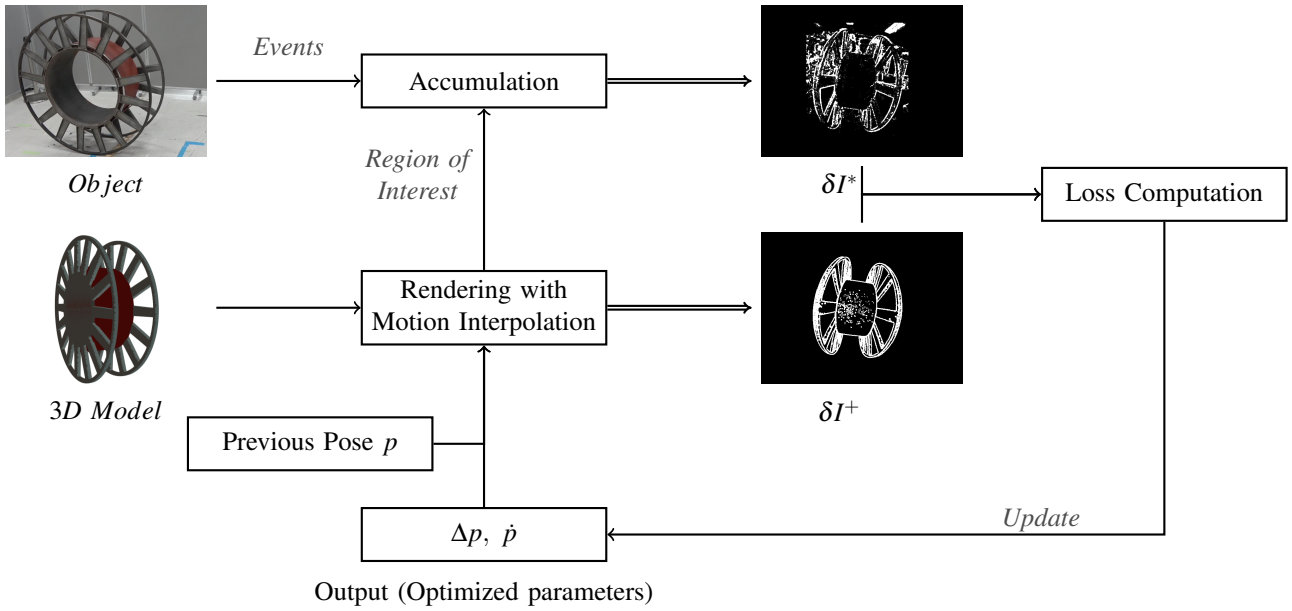


Fig. 2: Overview of BIAM. (Brightness increments in δI^* and δI^+ enlarged for visualization.)

III. Direct brightness increment alignment with motion interpolation: BIAM

This section describes the direct brightness increment alignment with motion interpolation (BIAM) we propose. It is summarized in the diagram of Figure 2. First, we review how event cameras work and present our new event frame generation method (Sec. III-A). Then we describe the method of predicting brightness increment images from 3D model and camera poses with motion interpolation (Sec. III-B). Finally, we describe the whole workflow (Sec. III-C).

A. Event Frame Generation

An event camera raises an event $e_k = (\mathbf{u}_k, t_k, p_k)$ at time $t_k \in \mathbb{R}_+$ as soon as a change of the logarithmic brightness L reaching the contrast sensitivity C is detected at the photosite $\mathbf{u}_k$ (pixel coordinates, for convenience):

$$\delta I(\mathbf{u}_k, t_k) \doteq I(\mathbf{u}_k, t_k) - I(\mathbf{u}_k, t_k - \Delta t_k) = p_k C, \qquad (1)$$

where polarity $p_k \in \{+1, -1\}$ is the sign of the brightness change and Δt_k is the time elapsed since the last event at the same pixel. Unlike traditional frame-based cameras that output images at a constant rate, event cameras output a stream of asynchronous events e_k in space-time.

A *brightness increment image* δI^* is the pixel-wise collection of polarities p_k of the $N_e \in \mathbb{N}$ events e_k, $k \in [\![1, N_e]\!]$ captured during a period of time $\Delta t = t_{N_e} - t_1$:

$$\delta I^*(\mathbf{u}) = \sum_{t_k \in T} p_k C \delta(\mathbf{u} - \mathbf{u}_k), \qquad (2)$$

with $T = \{t_1, t_2, ..., t_{N_e}\}$, in a simplified writing. Here the Kronecker δ selects the appropriate pixel. In first person view object tracking, the event polarities at contours between target object and the background are unpredictable as the background information is unknown. Imagine a gray cube object moves in front of an event camera. The event polarities at the scene contours will be inversed when the background changes from white to black. What's more, although there is no background to consider in inside-object tracking, the light reflection can also make the polarities unpredictable. Due to these reasons, we use the absolute increments when generating the observed brightness increment image.

In addition, we generate the brightness increment image δI^* by loading new events in the region-of-interest (ROI) to preserve the events triggered by the target object and exclude those relative to the background. The ROI is the smallest bounding box which contains the object in the intensity image I rendered from the hypothesized pose $\mathbf{p}$ of the previous frame or the initial pose. A small padding of a few pixels is also applied to compensate the motion and increase robustness. The event number N_e in each frame is determined by a certain fraction of the ROI size to assure there are always enough events for tracking and avoid too much motion blur at the same time. Therefore, our new δI^* is generated by accumulating N_e new events in the region-of-interest:

$$\delta I^*(\mathbf{u}) = |\sum_{t_k \in T, \mathbf{u}_k \in ROI} p_k C \delta(\mathbf{u} - \mathbf{u}_k)|, \qquad (3)$$

where $k \in [\![1, N_e]\!]$ and N_e is determined by a certain ratio of the pixel number in the ROI.

B. Predicting Brightness Increments

a) Basis: When the number of events N_e in (3) spans a small delta time $\Delta t = t_{N_e} - t_1$, the increment (1) can be approximated with Taylor's expansion. By substituting the brightness constancy assumption, we have

$$\delta I(\mathbf{u}) \approx |-\nabla I(\mathbf{u}) \cdot \Delta \mathbf{u}| = |-\nabla I(\mathbf{u}) \cdot \mathbf{v}(\mathbf{u}) \Delta t|, \qquad (4)$$

indicating that δI is caused by brightness gradients ∇I moving with velocity $\mathbf{v}$ on the image plane.

In the object tracking problem, we assume the edges make main contributions to successful tracking. Thus we use the pixels $\mathbf{u}$ where $\nabla I(\mathbf{u}) > \varepsilon$, where ε is a pre-set threshold for gradient. Since ∇I is rendered from the 3D model at virtual camera pose $\mathbf{p} \in \mathbb{R}^6$ and $\mathbf{v}(\mathbf{u})$ can be calculated with the point sensitivity matrix $\mathbf{L_u}$ and the camera velocity $\dot{\mathbf{p}} \in \mathbb{R}^6$, we have

$$\delta I(\mathbf{u}) = \begin{cases} |-\nabla I(\mathbf{u})\ \mathbf{L_u}\ \dot{\mathbf{p}}\ \Delta t|, & \text{if } \nabla I(\mathbf{u}) \geq \varepsilon) \\ 0, & \text{if } \nabla I(\mathbf{u}) < \varepsilon) \end{cases}. \quad (5)$$

b) Camera Model: In this work, we leverage the unified central camera projection model (UCM) [24] which is compatible with both perspective and omnidirectional cameras. The sensitivity matrix of a digital point $\mathbf{u}$ with respect to the camera pose is:

$$\mathbf{L_u} = \begin{bmatrix} \alpha_u & 0 \\ 0 & \alpha_v \end{bmatrix} \mathbf{L_x}, \quad (6)$$

where $\alpha_u \in \mathbb{R}^*$, $\alpha_v \in \mathbb{R}^*$ are the generalized focal length and:

$$\mathbf{L_x} = \begin{bmatrix} -\frac{1+x^2(1-\xi(\gamma+\xi))+y^2}{\rho(\gamma+\xi)} & \frac{\xi xy}{\rho} \\ \frac{\xi xy}{\rho} & -\frac{1+y^2(1-\xi(\gamma+\xi))+x^2}{\rho(\gamma+\xi)} \\ \frac{\gamma x}{\rho} & \frac{\gamma y}{\rho} \\ xy & \frac{(1+y^2)\gamma-\xi x^2}{\gamma+\xi} \\ -\frac{(1+x^2)\gamma-\xi y^2}{\gamma+\xi} & -xy \\ y & -x \end{bmatrix}^T, \quad (7)$$

where $\mathbf{x} = [x, y]^\top \in \mathbb{R}^2$ is the projection of a 3D point $\mathbf{X} = [X, Y, Z]^\top \in \mathbb{R}^3$ on the normalized image plane, $\xi \in \mathbb{R}$ is a parameter associated to the shape of lens, $\rho = \sqrt{X^2+Y^2+Z^2}$ and $\gamma = \sqrt{1+(1-\xi^2)(x^2+y^2)}$ [25].

c) Motion Interpolation: In Sec. I we introduced the problem of accumulation blur when generating δI^*. To produce similar distribution in the predicted brightness increment image δI, we propose the motion interpolation method.

Suppose the relative camera pose between two desired frames δI^*_m and δI^*_{m+1} is $\Delta\mathbf{p}$ in $SO(3)$. A visible point on the 3D model moves from ${}^{c_m}\mathbf{X} \in \mathbb{R}^3$ to ${}^{c_{m+1}}\mathbf{X} \in \mathbb{R}^3$ in 3D space. Because of this motion, we have:

$${}^{c_{m+1}}\mathbf{X} = {}^{c_{m+1}}\mathbf{M}_{c_m}\ {}^{c_m}\mathbf{X}, \quad (8)$$

where ${}^{c_{m+1}}\mathbf{M}_{c_m} \in SE(3)$ is the Rodrigues' formulae expression from $\Delta\mathbf{p}$. On the 2D image plane, the 3D points are projected to $\mathbf{u}_m \in \mathbb{R}^2$ and $\mathbf{u}_{m+1} \in \mathbb{R}^2$ respectively, as shown in Figure. 3. Assuming the depth $\rho \in \mathbb{R}_+$ constant for each pixel, we have:

$$\mathbf{u}_{m+1} = \pi(\frac{1}{\rho}(\ {}^{c_{m+1}}\mathbf{M}_{c_m}\rho\ \pi^{-1}(\mathbf{u}_m))) \quad (9)$$

where $\pi^{-1} : \mathbb{R}^2 \Rightarrow \mathbb{R}^3$ back-projects 2D pixel coordinate to unit line of sight, $\pi : \mathbb{R}^3 \Rightarrow \mathbb{R}^2$ projects the unit line of sight to 2D plane.

Instead of calculating the intensity variation images only at $\mathbf{u}_{m+1}$ as in [9], we interpolate the motion by predicting the absolute intensity variation image at a series of intermediate

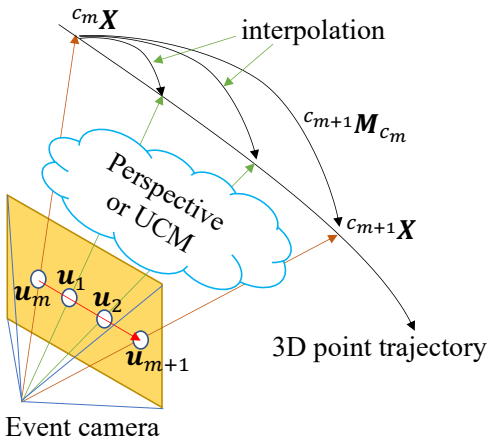


Fig. 3: Motion Interpolation

pixels on the 2D trajectory from $\mathbf{u}_m$ to $\mathbf{u}_{m+1}$. Formally, we predict the absolute brightness increment image by:

$$\delta I^+ = \frac{1}{n}\sum_{i=1}^{n} \delta I(\mathbf{u}_i), \quad (10)$$

where $\delta I(\mathbf{u}_i)$ are computed by (5) at the intermediate pixels $\mathbf{u}_i$ on the 2D trajectory between $\mathbf{u}_m$ and $\mathbf{u}_{m+1}$:

$$\mathbf{u}_i = \pi(\frac{1}{\rho}({}^i\mathbf{M}_{c_m}\ \rho\ \pi^{-1}(\mathbf{u}_m))). \quad (11)$$

where ${}^i\mathbf{M}_{c_m}$ is the transformation matrix from ${}^{c_m}\mathbf{X}$ to intermediate point ${}^i\mathbf{X}$ computed by taking $\frac{i}{n}$ of both translation and rotation components in ${}^{c_{m+1}}\mathbf{M}_{c_m}$. The interpolation parameter n is determined by rounding the length of 2D segment between $\mathbf{u}_m$ and $\mathbf{u}_{m+1}$ which can be approximated by assuming linear motion in 2D space.

C. The Optimization Problem for Alignment

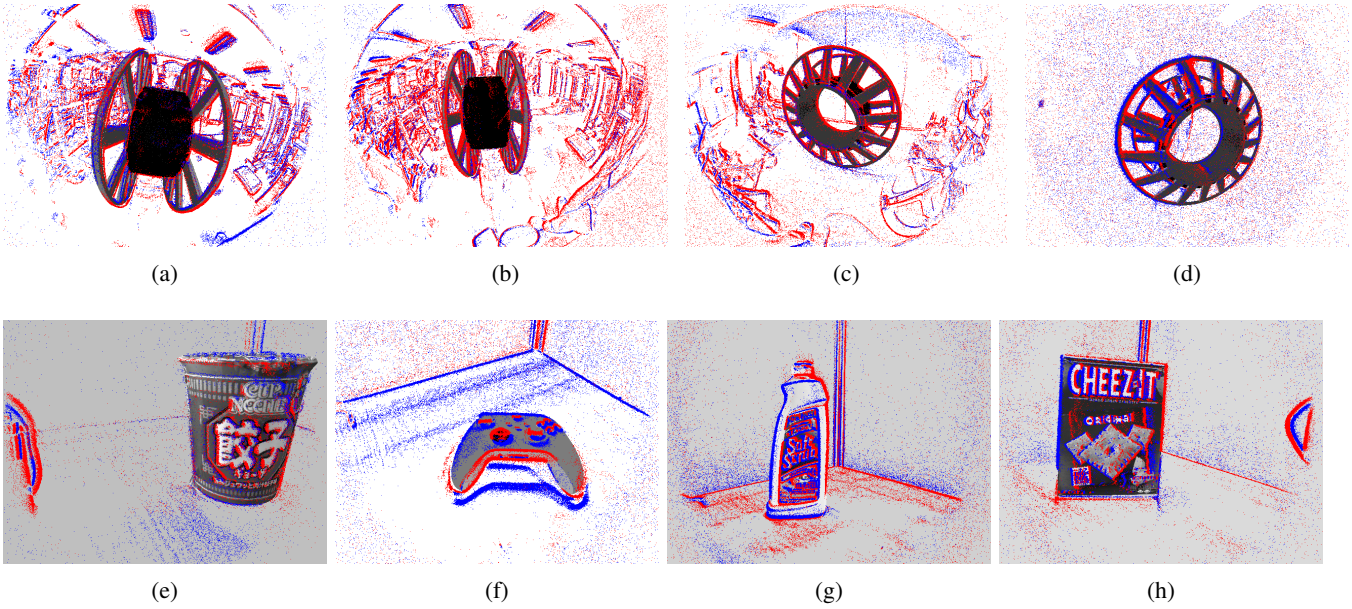
Aligning δI^* with the object 3D model to estimate the pose $\mathbf{p}$ and the velocity $\dot{\mathbf{p}}$ of the event camera is done by minimizing the Sum-of-Squared-Differences (SSD) between δI^* in (3) and the predicted δI^+ in (10). $\mathbf{p}$ is calculated from the optimal pose of the previous frame and the relative pose $\Delta\mathbf{p}$ to reach the current camera pose, thus solving the below optimization problem:

$$[\widehat{\Delta\mathbf{p}}, \widehat{\dot{\mathbf{p}}}]^\top = \arg\min_{\Delta\mathbf{p},\dot{\mathbf{p}}} || \frac{\delta I^+}{||\delta I^+||_2} - \frac{\delta I^*}{||\delta I^*||_2} ||^2. \quad (12)$$

The tracker takes only the 3D model of the object and events as input. The initial pose is manually set before the tracking starts. After the optimization of each frame, the new pose $\mathbf{p}$ is used as the initial pose of the next frame. $\mathbf{p}$ is also used to calculate the bounding box for event loading.

IV. Experiments

We evaluate the performance of BIAM on a new event-based object tracking dataset. First, we introduce the new dataset and the experimental setup (Sec. IV-A). After that we introduce the metrics we use for evaluation (Sec. IV-B). Then we present the object tracking experiments with BIAM (Sec. IV-C). Finally, we present complementary experiments and discussions (Sec. IV-D).



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4: Qualitative results of object tracking with BIAM on the new dataset. The images depict the events over the rendered intensity images. (a) to (d) are the results of the large coil, (e) to (h) are the results on the other four objects.

A. Dataset and Experimental Setup

We evaluate the performance of BIAM by applying it to the tracking of 5 different objects: a large coil, a noodle box, a controller, a bleach cleanser and a cracker box. The large coil (Fig. 1b) has a diameter of 130 cm and weighs 140 kg. It is used in industries to carry cables, paper, ruber. The noodle box and the controller are common objects. The bleach cleanser and the cracker box are from the Yale-CMU-Berkeley dataset [26]. Their 3D models are dense point clouds sampled from featured Computer Aided Design (CAD) mesh or scanned mesh. We present quantitative evaluation on the large coil. For the other objects, we present qualitative to show that our method is general.

Considering the size of the coil, we use a fisheye lens so that the camera could always see the whole object at similar viewpoints that an operator or a robot moving such an object would have. We also use pinhole lens for ablation study. For the other 4 objects, we place them on the table and move the perspective camera at a distance of about 30cm.

Our dataset consists of 14 *perspective* sequences captured with a conventional lens and 15 *omni* sequences captured with a fisheye lens, in which *perspective* 1-6 and *omni* sequences 1-15 are about the coil, *perspective* 7-8 for the noodle box, *perspective* 9-10 for the controller, *perspective* 11-12 for the bleach cleanser and *perspective* 13-14 for the cracker box. *perspective* 1-2 and *omni* 14-15 sequences were captured by moving the coil in front of static camera. The others were captured by moving the camera with a Universal Robots UR10 robot arm or human hands in front of the static objects. The duration of the sequences vary from 2s to 20s. The camera-robot calibration was performed before recording the sequences with the robot arm to record reliable ground truth. The sequences acquired with the robot arm show various trajectories from pure translations to complicated 6-DoF motions. When using the robot arm, the camera poses in robot base frame were recorded from the robot side with the extrinsic parameters acquired from hand-eye calibration.

The events are captured with a Prophesee Gen3.1 event camera (640×480 pixels of $k = 15$ μm pitch) with a pinhole lens and a fisheye lens. In our experiments, both cameras are well calibrated with a blinking checkboard pattern. In the hand-held sequences, we shake the camera against real world background. We will share the dataset in the future.

Emprically, we set once for all N_e (Section III-A) to 0.15 times of the pixel number in the ROI in order to ensure high SNR and avoid too much blur at the same time. In all the experiments the initial pose is manually set.

B. Evaluation Metrics

We use the evaluation metrics in [9], where the position error is calculated by the Euclidean distance between the ground truth and the estimated position and orientation error measured with the geodesic distance in the rotation group $SO(3)$. Although [9] presented the quantitative results with median errors, we also give root-mean-square (RMS) errors which have been used by more state-of-the-art works. Since the true object pose in robot frame is hard to get, we calculate the transformation matrix from the initial camera frame c_0 to the current camera frame c:

$$ {}^{c}\mathbf{M}_{c_0} = {}^{c}\mathbf{M}_{b}\ {}^{c_0}\mathbf{M}_{b}^{-1}, \tag{13} $$

where ${}^{c}\mathbf{M}_{b}$ and ${}^{c_0}\mathbf{M}_{b}$ are the transformation matrices from the robot base frame b to a camera frame. We extract the current camera pose relative to the initial frame from the transformation matrix and use its position and orientation components in the evaluation.

TABLE I: Comparison between BIA and BIAM on *omni* recordings in RMS Position and Rotation error.

	BIA		BIAM	
	Pos. [mm]	Rot. [°]	Pos. [mm]	Rot. [°]
omni Traj. 1	28.4	1.83	**19.9**	**0.86**
omni Traj. 2	39.7	2.21	**33.0**	**1.54**
omni Traj. 3	**35.8**	1.35	36.8	**1.18**
omni Traj. 4	15.8	2.26	**15.5**	**2.19**
omni Traj. 5	22.4	1.50	**20.1**	**1.40**
omni Traj. 6	227	14.4	**12.9**	**2.11**
omni Traj. 7	502	30.9	**28.0**	**2.50**
omni Traj. 8	348	22.1	**16.6**	**2.56**
omni Traj. 9	/	/	**18.7**	**1.17**
omni Traj. 10	/	/	**16.3**	**1.59**

C. Results

We start BIAM on all the *omni* sequences with manually set initial poses. Quantitatively (Tab. I), BIAM tracked the position and rotation poses at the best accuracy on all the sequences, regardless of the trajectory or the viewpoint. The tracking of BIAM is as good on the fast motion sequences (1.0 to 2.0 m/s, Traj. 9 - Traj. 12) as on those with slower camera motion (0.1 to 0.5 m/s, Traj. 1 - Traj. 8). The maximum RMS error is 36.8 mm for position and 3.87 ° for rotation. Figure 5 shows the estimated and ground truth trajectories of BIAM on Traj. 9.

Qualitative results of BIAM on all the objects are reported in Figure 4a to 4h. BIAM reports small difference between the events captured and the image rendered in all the *omni* sequences. Even if there are many irrelevant objects and patterns in the background like Figure 4a and 4b, the tracking still runs stable.

BIAM is capable of estimating $\mathbf{p}$ accurately in fast motion. The average frames per second (FPS) of tracking is 351.39 Hz on Traj. 9 and 403 Hz on Traj. 10. For the fast hand-held sequences (Traj. 11 and Traj. 12), the FPS reached up to 750 Hz due to the super fast camera motion.

For more details, please refer to the accompanying video.

D. Discussion

a) Ablation study: For comparison, we also run the method without motion interpolation (BIA). The quantitative results on the coil can be found in Table. I. On most sequences the accuracy of BIA is worse than BIAM, and on the fast motion sequences BIA always encounters failure (*omni* Traj. 9 - Traj. 12). On 4 of the sequences of the other objects, BIA also fails to track until the end.

When we do not use the absolute brightness increments and follow the event generative model in [9], the tracking fails on more than 70% of the sequences, mainly due to the failure in predicting the event polarities in real world scenes.

We also tried to track the large coil with perspective camera on *perspective* 1-6. However, all the experiments failed in 1s because the object could only show part of itself.

b) Camera tracking on public dataset: As there is about no public event-based object tracking dataset, we evaluate BIAM on the *room* dataset where the method in [9] was evaluated on. Since there are a lot of similarities between camera tracking and object tracking, we believe this evaluation is meaningful. On *room* Traj. 1, [9] reached a median position error of 9.95cm and a median orientation error of 3.08°. While using BIAM, the median position error is 2.34cm and the median orientation is 0.783° on the same sequence. Both the position error and the orientation error are decreased significantly with our new method.

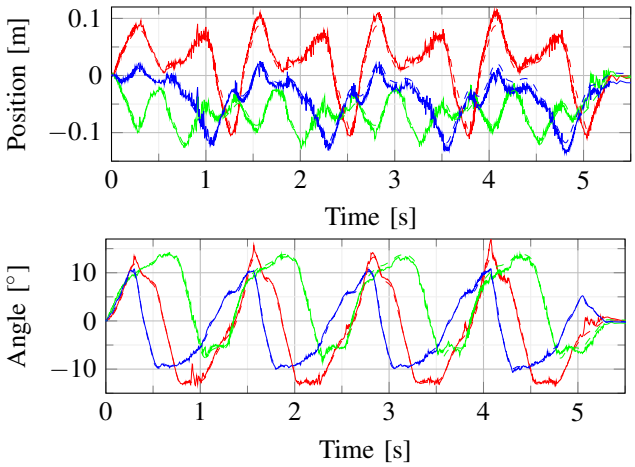


Fig. 5: Object pose tracking results of BIAM with position and rotation variations on *omni* Traj. 9. The poses of X, Y, Z axes are plot as red, green and blue curves (dash lines: ground truth, dense lines: estimated poses).

c) Computational cost: Compared to [9] which took 22.7s to process each frame, we realized faster processing. Although we also built our method with ceres-solver, we made a better implementation in multi-thread and point selection. To track each event frame, BIAM takes 0.63s to track each event frame when tracking the large coil.

V. Conclusion

This paper proposed the 3D tracking of objects described by a 3D model within events captured with a event camera. For this, we extended the state-of-the-art method capable of directly tracking in brightness increment images the pose of an event camera in a room, by introducing motion interpolation and a new event frame representation. We generate the desired image by accumulating the events in the region-of-interest and ignoring the polarity information while render the predicted image with motion interpolation to produce similar accumulation blur.

The motion interpolation is proved a key contribution in the evaluation on both public camera tracking dataset and our new object tracking dataset, especially when the camera motion is fast. We also evaluated our proposed method BIAM with objects closely observed with a event camera. It shows much more reliable tracking for significantly longer times than the previous state-of-the-art approach especially in fast relative motion between the camera and the target objects.

References

[1] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, *et al.*, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[2] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, 2020.

[3] Y. Hu, J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2020, pp. 1–6.

[4] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5816–5824.

[5] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *Proc. of European Conference on Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016, pp. 349–364.

[6] E. Dubeau, M. Garon, B. Debaque, R. d. Charette, and J.-F. Lalonde, "RGB-D-E: Event camera calibration for fast 6-DOF object tracking," in *Proc. of IEEE International Symposium on Mixed and Augmented Reality*, 2020, pp. 127–135.

[7] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

[8] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "TUM-VIE: The TUM stereo visual-inertial event dataset," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[9] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, "Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization," in *Proc. of IEEE International Conference on Robotics and Automation*, 2019, pp. 325–331.

[10] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, "Event-aided direct sparse odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5781–5790.

[11] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger, "Interacting maps for fast visual interpretation," in *IEEE International Joint Conference on Neural Networks*, 2011, pp. 770–776.

[12] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 632–639, 2017.

[13] D. Weikersdorfer and J. Conradt, "Event-based particle filtering for robot self-localization," in *IEEE International Conference on Robotics and Biomimetics*, 2012, pp. 866–870.

[14] D. Weikersdorfer, R. Hoffmann, and J. Conradt, "Simultaneous localization and mapping for event-based vision systems," in *Proc. of 9th International Conference Computer Vision Systems, St. Petersburg, Russia, July 16-18, 2013. 9*, pp. 133–142.

[15] C. Reinbacher, G. Munda, and T. Pock, "Real-time panoramic tracking for event cameras," in *IEEE International Conference on Computational Photography*, 2017, pp. 1–9.

[16] G. Gallego, J. E. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-dof camera tracking from photometric depth maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2402–2412, 2017.

[17] A. Censi and D. Scaramuzza, "Low-latency event-based visual odometry," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 703–710.

[18] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *Proc. of European Conference on Computer Vision*, 2016, pp. 349–364.

[19] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1433–1450, 2021.

[20] B. Ramesh, S. Zhang, Z. W. Lee, Z. Gao, G. Orchard, and C. Xiang, "Long-term object tracking with a moving event camera." in *the British Machine Vision Conference*, 2018, p. 241.

[21] R. Jiang, X. Mou, S. Shi, Y. Zhou, Q. Wang, M. Dong, and S. Chen, "Object tracking on event cameras with offline–online learning," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 165–171, 2020.

[22] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 3769–3776.

[23] H. Li and J. Stueckler, "Tracking 6-dof object motion from events and frames," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 14 171–14 177.

[24] J. Barreto, F. Martin, and R. Horaud, "Visual servoing/tracking using central catadioptric images," in *Experimental Robotics VIII*, 2003, pp. 245–254.

[25] G. Caron, E. Marchand, and E. Mouaddib, "Photometric visual servoing for omnidirectional cameras," *Autonomous Robots*, vol. 35, no. 2-3, pp. 177–193, Oct. 2013.

[26] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *IEEE International Conference on Advanced Robotics*, 2015, pp. 510–517.