2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Fusion of Events and Frames using 8-DOF Warping Model for Robust Feature Tracking

Min Seok Lee[1], Ye Jun Kim[2], Jae Hyung Jung[1], and Chan Gook Park[1]

***Abstract*— Event cameras are asynchronous neuromorphic vision sensors with high temporal resolution and no motion blur, offering advantages over standard frame-based cameras especially in high-speed motions and high dynamic range conditions. However, event cameras are unable to capture the overall context of the scene, and produce different events for the same scenery depending on the direction of the motion, creating a challenge in data association. Standard camera, on the other hand, provides frames at a fixed rate that are independent of the motion direction, and are rich in context. In this paper, we present a robust feature tracking method that employs 8-DOF warping model in minimizing the difference between brightness increment patches from events and frames, exploiting the complementary nature of the two data types. Unlike previous works, the proposed method enables tracking of features under complex motions accompanying distortions. Extensive quantitative evaluation over publicly available datasets was performed where our method shows an improvement over state-of-the-art methods in robustness with greatly prolonged feature age and in accuracy for challenging scenarios.**

I. Introduction

Standard cameras collect light intensity measurements of all pixels and create a frame at fixed rates, whereas event cameras or Dynamic Vision Sensors (DVS) [1] generate events whenever logarithmic brightness changes are detected at each pixel. Thanks to their asynchronous nature with low latency (1 μs), the output, an event, is not affected by motion blur and provides higher dynamic range (HDR) of 120dB compared to standard cameras. Thus, many studies have focused on exploiting such merits of the event camera in highly dynamic motion and HDR scenarios [2], [3], [4], [5], [6], [7].

We specifically focus on solving visual feature tracking among numerous computer vision problems event camera presents. Regarding event-only feature tracking, since the appearances of events vary depending on the moving direction of the sensor [7], resolving data association between newly created events and already established features is a challenge. Hence, several existing studies [7], [8], [9] have tackled such problem by using both events and frames in a complementary manner. To facilitate the two different outputs together, the most commonly used sensor is the Dynamic and Active-pixel Vision Sensor (DAVIS) [10], which generates both frames and events, sharing a pixel array. With the help of this hybrid sensor, [7], [8], [9] utilizes both data types through an optimization framework that registers events to features derived from a frame. When registering the data points (events) to a template feature, a transformation, or warping, model should be chosen. Abovementioned works all employ rigid warping of 3-DOF, which is insufficient to account for all transformation in 2D space.

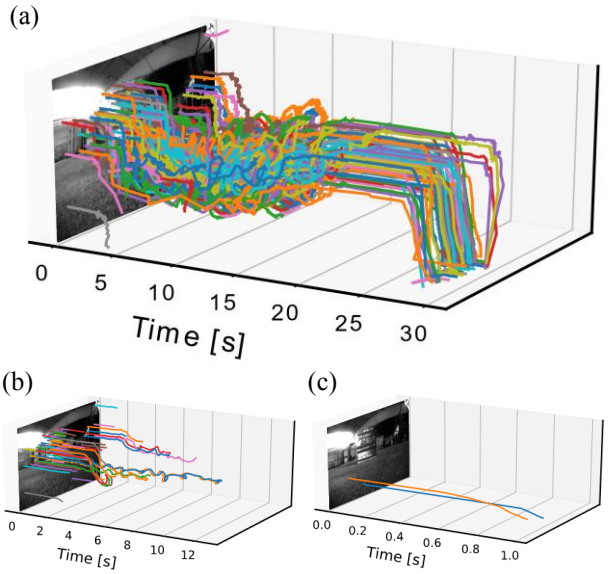

Fig. 1. Feature tracks visualized for (a) the proposed method, (b) EKLT [7], and (c) Zhu *et al*. [3] on the sequence *indoor_forward_5* of the UZH-FPV Drone Racing Dataset [12]. The maximum number of initiating features is set to 100, and tracking is terminated when no features are alive. Our method initiates more features and keeps tracked features alive for much prolonged time compared to the two state-of-the-art methods.

With such motivation, we propose a robust feature tracking method that is capable of explaining general motion of the tracked features by resolving 8-DOF in warping events to a frame. The proposed method does so by adopting homography, or projective, warping in a two-step manner when registering accumulated events on to brightness increment patches predicted by a preceding frame. The first step optimizing only the translation parameters is repeated until features travel more than a threshold pixel distance. Upon exceeding the threshold, the algorithm enters the second step where the homography parameters are optimized. By separating the optimization process, we attempt to achieve higher computational efficiency and optimization stability. Our method is compared against state-of-the-art event-based and event-and-frame-based feature trackers on two datasets with real sequences: the Event-Camera Dataset [11] and the UZH-FPV Drone Racing dataset [12]. Extensive evaluation proves robustness of our method by showing higher number of tracked features with longer feature ages, and demonstrates

[1] Navigation and Electronic System Laboratory, Department of Aerospace Engineering, Seoul National University, Seoul 08826, Republic of Korea, {mslee1996, lastflowers, chanpark}@snu.ac.kr
[2] Hyundai motor group, Seoul 06182, Republic of Korea, yejun@hyundai.com

higher accuracy for most sequences when comparing normalized tracking errors. Our main contributions are as follows:

- We present a robust feature tracker that incorporates both events and frames through registering brightness increment patches using 8-DOF warping model.
- Warping model of high DOF is implemented in a two-step manner to avoid computational hazard and instability in optimization.
- We perform extensive quantitative evaluation on two real datasets, total of 28 sequences, some with challenging situations. To assess the efficacy of our algorithm, we modify one of the state-of-the-art methods and apply 8-DOF warping model without our two-step process.
- Our method proves robustness by showing considerably longer feature ages compared to existing state-of-the-art methods, as in Fig. 1. Our method also shows improved feature tracking accuracy of normalized tracking errors.

This paper is based on our previous work [13], which we extend in several ways: we broaden the context of the method to general situations not limited to a specific scenario; we perform feature tracking evaluation on real datasets with plentiful sequences instead of few synthetic sequences; we perform quantitative evaluation with a new error metric against the state-of-the-art methods and provide graphical comparisons.

II. Related Work

Previous works on applications such as object tracking [14], [15], [16], [17], [18] and localization [19], [20], [21], [22], [23], [24], [25] adopt event-based or event- and frame-based feature tracking methods. Although event cameras are relatively new compared to frame-based cameras, numerous studies have presented novel approaches of utilizing events in tracking features. We briefly introduce a few that carry importance to our work.

The work of [3] presents an event-only feature tracker that resolves data association through optical flow estimation via Expectation-Maximization (EM) scheme. Estimated optical flow is used to propagate a set of events, which is then registered on to a previous set using EM-ICP [26] method. [7] proposes the state-of-the-art event-and-frame-based feature tracker EKLT (Event-based Kanade-Lucas-Tomasi tracker). This seminal work extends the KLT tracker [27], [28] to brightness patches of events and frames. The optimization process tries to minimize the difference between the patches, by estimating the warping parameters. Here, rigid warping is adopted as the warping model. [8] presents similar feature tracking method as [7] in the sense that rigid model is used as a warping model and both events and frames are used. The difference is that the warping model in [7] refers to feature patches of $t = 0$ and current, whereas in [8] it refers to patches of current features and features just before a new event. However, it has the same problem that the rigid model cannot reflect all shape changes of the patch over time. [5] has the same optimization framework as [7] but tracks 6-DOF pose of the camera in a given photometric 3D map. In [5], the frame and event patch pairs in [7] are replaced by a photometric 3D map and every pixel. Since there is a photometric 3D map, the warping model on the 2D image plane is replaced by a pin-hole camera model that projects a 3D point on to an image. [6] uses a filter to track camera pose, and has a similar scenario as [5]. The work also presents a photometric depth map that does not need a warping model. [29] performs tracking on objects defined with a set of events. A modified affine model with 5-DOF is used as the warping model in registration process. Whenever an event occurs, the x and y-axis scales are updated compared to the set of points from a given object. [30] presents a general framework for event cameras called contrast maximization that can be used in applications of estimating motion, depth, and optical flow. [30] uses homography as the geometric model to estimate ego-motion in planar scenes. [30] inspired us to adopt homography warping model as our registration method.

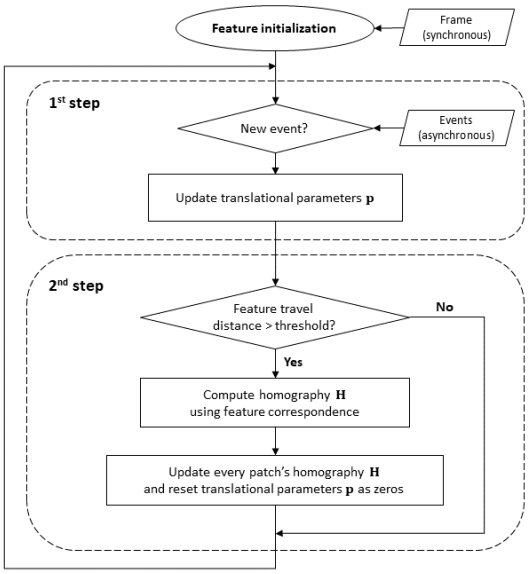

Fig. 2. Schematic overview of our two-step optimization process

III. Method

In this section, we shed light on our robust feature tracking method, starting with the definition of event measurements. Then, we elaborate on how events and frames are used to initialize feature patches of their own. Next, the 8-DOF warping model that we devised to register event patches on to template patches predicted from frames is presented. Lastly, we present our two-step optimization process that enables adoption of our high DOF warping model in an efficient and stable manner. An overview of our optimization process is demonstrated in Fig. 2.

A. Event Measurements

Standard cameras measure the absolute brightness of light at each pixel and combine measurements of all pixels to generate a frame at a fixed rate. However, as for an event camera, each pixel operates asynchronously. At each pixel, an event $e_k = (x_k, y_k, t_k, p_k)$ occurs whenever $\boldsymbol{L}$ (logarithmic brightness) changes by a threshold C, where $\mathbf{u}_k = (x_k, y_k)$ is the pixel coordinate, t_k is the time, and $p_k \in \{-1, +1\}$ is the

polarity indicating the increase or decrease of brightness. For an ideal event camera, threshold C is constant. Thus, all e_k satisfy the following equation:

$$\Delta \boldsymbol{L}(\mathbf{u}_k, t_k) \triangleq \boldsymbol{L}(\mathbf{u}_k, t_k) - \boldsymbol{L}(\mathbf{u}_k, t_k - \Delta t_k) = p_k C, \quad (1)$$

where Δt_k is the time elapsed from the latest event in pixel $\mathbf{u}_k$.

B. Patch Initialization

The proposed method begins with initializing patches created with events and frames separately. Initialization process is similar to that of [7]. For events, the polarities of incoming events are accumulated over a time window $\Delta\tau$ to create a brightness increment patch $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$, which can be computed as

$$\Delta\bar{\boldsymbol{L}}(\mathbf{u}) = \sum_{t_k \in \Delta\tau} p_k C \delta(\mathbf{u} - \mathbf{u}_k), \quad (2)$$

where δ is the Kronecker delta.

For frames, we first extract Harris corners [31] from a frame denoted by $\hat{\boldsymbol{L}}$. Then, intensity patches are created from the extracted corners to compute the intensity gradient $\nabla\boldsymbol{L}(\mathbf{u})$. With this brightness gradient patch, we attempt to create an estimate of $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$ by finding a warping parameter $\mathbf{p}$ and an optical flow $\mathbf{v}$ that minimizes the difference between the estimate and $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$. The estimate can be computed as

$$\Delta\hat{\boldsymbol{L}}(\mathbf{u}; \mathbf{p}, \mathbf{v}) \doteq -\nabla\hat{\boldsymbol{L}}\big(\boldsymbol{W}(\mathbf{u}; \mathbf{p})\big) \cdot \mathbf{v}\Delta\tau, \quad (3)$$

where $\boldsymbol{W}$ is the warping model. A simple demonstration of warping is shown in Fig. 3.

C. Warping Model

In this paper, we propose an 8-DOF warping model modified from the homography warping model. The following is a summary of several warping models including our own:

- **Rigid model**: Rigid model has 3-DOF warping parameters: $\mathbf{p} = \{\theta, t_x, t_y\}$. Therefore, it can express rotation and translation of a patch.
- **Homography model**: When the same plane is observed from two viewpoints, transformation between the two images can be expressed as a homography matrix $\mathbf{H}$. Therefore, arbitrary warping of patches on a plane can be expressed as the following model:

$$\boldsymbol{W}(\mathbf{u}; \mathbf{p}) = \boldsymbol{H}\big(\mathbf{u}; \mathbf{H}(\mathbf{p})\big)$$

$$\mathbf{p} = \{h_i | 1 \le i \le 8\}$$

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix}, \quad (4)$$

where $\mathbf{p}$ is composed of elements of $\mathbf{H}$. Homography transformation $\boldsymbol{H}$ is defined as

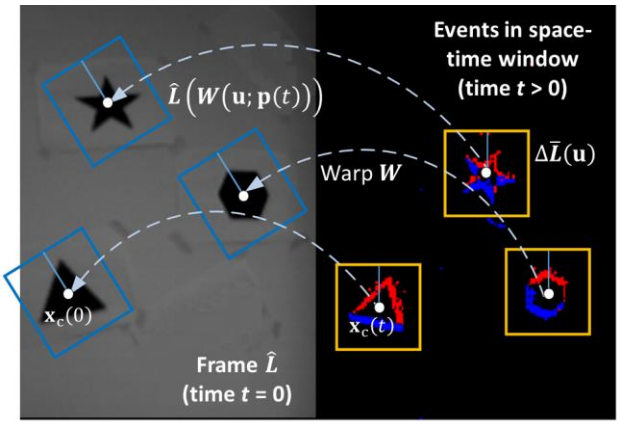

Fig. 3. Illustration of warping of the patches. Yellow patches present events in space-time window at time $t > 0$. Blue patches on $\hat{\boldsymbol{L}}(t = 0)$ are result after warping $\boldsymbol{W}$ of yellow patches. Red and blue points represent events with $p_k = 1$ and -1, respectively.

$$\boldsymbol{H}\big(\mathbf{u}; \mathbf{H}(\mathbf{p})\big) = \frac{1}{h_7 x + h_8 y + 1} \begin{bmatrix} h_1 x + h_2 y + h_3 \\ h_4 x + h_5 y + h_6 \end{bmatrix} \quad (5)$$

- **Proposed warping model**: The proposed warping model can be expressed as

$$\boldsymbol{W}(\mathbf{u}; \mathbf{p}, \mathbf{H}) = \boldsymbol{H}(\mathbf{u}; \mathbf{H}) + \mathbf{t}(\mathbf{p})$$

$$\mathbf{p} = \{t_x, t_y\}. \quad (6)$$

Similar to the homography model, (6) can express arbitrary warping of patches on a plane. The biggest difference from the homography model is that the translation term $\mathbf{t}(\mathbf{p})$ is added. Elements of $\mathbf{H}$ are omitted from $\mathbf{p}$ and are only left with 2-DOF translation parameters $\{t_x, t_y\}$.

D. Two-step Optimization

Simply applying a warping model with high-DOF causes degradation in computational efficiency and stability of optimization. To solve this problem, we present a two-step optimization strategy using the proposed warping model. The naming refers to the fact that $\mathbf{H}$ and $\mathbf{p}$ of the proposed warping model are solved separately in two-fold.

- **1st step**: $\mathbf{p} = \{t_x, t_y\}$ is updated by solving the optimization problem,

$$\min_{\mathbf{p},\mathbf{v}} \left\| \frac{\Delta\bar{\boldsymbol{L}}(\mathbf{u})}{\|\Delta\bar{\boldsymbol{L}}(\mathbf{u})\|} - \frac{\Delta\hat{\boldsymbol{L}}(\mathbf{u}; \mathbf{p}, \mathbf{H}, \mathbf{v})}{\|\Delta\hat{\boldsymbol{L}}(\mathbf{u}; \mathbf{p}, \mathbf{H}, \mathbf{v})\|} \right\|^2 \quad (7)$$

which aims to minimize the difference between $\Delta\bar{\boldsymbol{L}}(\mathbf{u})$ and $\Delta\hat{\boldsymbol{L}}(\mathbf{u})$. The important point is that when solving (7), $\mathbf{H}$ is fixed and only $\mathbf{p}$ and $\mathbf{v}$ are optimized.

- **2nd step**: $\mathbf{H}$ in the proposed warping model is updated as follows:

$$\mathbf{x}_c^i(t) = \boldsymbol{W}^{-1}(\mathbf{x}_c^i(0); \mathbf{p}(t), \mathbf{H}) \ \ (i = 1, \dots, N) \quad (8)$$

$$\min_{\mathbf{H}} \sum_{i} \left\| \mathbf{x}_c^i(0) - \boldsymbol{H}(\mathbf{x}_c^i(t); \mathbf{H}) \right\|^2, \qquad (9)$$

where $\mathbf{x}_c$ is the center of features. After the 1st step, $\mathbf{p} = \{t_x, t_y\}$ is updated and the locations of the features are changed. For N features that have not been lost, new locations can be obtained using (8). At this time, $\mathbf{H}$ is a fixed value used in the 1st step. Then, since we know the correspondence between initial features and current features, we can obtain $\mathbf{H}$ that minimizes (9). When doing so, outliers are excluded by using RANSAC instead of using all matching pairs of features. Ultimately, $\mathbf{H}$ of each feature is updated with a newly calculated value, and $\mathbf{p} = \{t_x, t_y\}$ is reset to zeros.

The proposed algorithm is based on the fact that the shape changes of the patches is relatively slower than the translation of the patches. In Fig. 2, the 1st step in which $\mathbf{p} = \{t_x, t_y\}$ is updated is executed every time a new event comes in. Therefore, it can be seen that $\mathbf{p} = \{t_x, t_y\}$ is updated online and event-by-event. However, the 2nd step, which updates $\mathbf{H}$ that reflects the change in the shapes of the patches, is only performed when pixels have travelled for more than a certain distance, or a threshold. The threshold is a user-set parameter, where in this paper, we set it as 1px. Namely, when the mean travel distance of the features is more than 1px, the 2nd step is entered. As a result, $\mathbf{H}$ is updated relatively slower than $\mathbf{p} = \{t_x, t_y\}$. Because the warping parameter $\mathbf{p}$ estimated in the optimization process of the 1st step is reduced to 2-DOF, the probability of converging to a local minimum in the optimization process decreases.

IV. Experiments

To validate the performance of our feature tracker, we perform quantitative evaluation on real datasets [11] and [12] against three methods: EKLT [7], Zhu *et al*. [3], and h-EKLT. EKLT and Zhu *et al*. provide open-source implementations written in C++ and MATLAB, respectively. When running the baselines, we use all default settings provided by their authors. We modified EKLT to employ homography warping model instead of rigid warping model, creating h-EKLT. This is to accurately assess the efficacy of our two-step optimization framework alone: by comparing our method to h-EKLT, the effect of higher DOF warping model on feature tracking performance is ablated.

For all feature trackers on all evaluated dataset sequences, maximum number of tracked features is set as 100 and tracking is terminated when all features are lost. In other words, no new features are initialized after the first image.

A. Real Data

We evaluate our method on two real datasets: the Event-Camera Dataset [11] and the UZH-FPV Drone Racing dataset [12]. Event and frame outputs of [11] is recorded with DAVIS 240C [10] and the dataset is comprised of 20+ sequences including 6-DOF, rotation-only, translation-only, dynamic and HDR scenarios on various scenes. For comparison, we choose 15 sequences that are most frequently benchmarked.

[12] is the most aggressive event camera dataset to date, with large acceleration and fast-changing motions. Sequences are taken on two environments, indoor and outdoor, with a quadrotor facing two-ways: forward and 45-degree downward. The drone racing quadrotor is equipped with miniDAVIS 346, providing events and frames. We choose 13 sequences that vary in difficulty and sceneries for evaluation.

B. Tracking Error

For continuous tracking and visual odometry, being able to track features for long period of time is a huge advantage. However, feature trackers with longer feature ages tend to accumulate tracking errors due to drift. Hence, it is unfair to compare only the mean tracking errors of the trackers when evaluating their performances. To take feature age into consideration, we compare normalized tracking errors instead, which are computed by dividing the mean tracking errors with corresponding feature ages. The best result for each sequence is highlighted in bold.

The normalized tracking errors on sequences of [11] is shown in Table I, and the left column of Fig. 4 shows graphical comparison of the tracking errors over time on three sequences of [11], with ours in blue. Our method outperforms all compared methods on 14 out of 15 sequences, showing average of 33% lower normalized error than the runner-up EKLT. When comparing mean tracking errors only, which are the numbers in brackets, EKLT shows better performance overall; yet, after normalizing, our method outperforms the state-of-the-art as longer feature ages were taken into account. We find that even after normalizing, EKLT show better performance in "Poster Translation" and "Shapes Translation" sequences, both comprised of translational motions only. This indicates that for situations with only such simple motion, rigid model is sufficient for explaining the feature distortions. However, in reality, this is rarely true.

The normalized tracking errors on sequences of [12] is shown in Table II, and the left column of Fig. 6 shows graphical comparison of the tracking errors over time on three sequences of [12], with ours in blue. Similar to results on [11], our method outperforms all compared methods on 10 out of 13 sequences, and show comparable errors for the rest. Compared to EKLT, our method shows average of 23% lower normalized tracking error.

C. Feature Age

In proving robustness of our method, feature age is an important criterion to inspect. Mean feature age on datasets [11] and [12] are shown in Table I and Table II, respectively. For both datasets, our method shows remarkably longer feature age, outperforming all compared methods by far on all 28 sequences except one. Results show that our approach achieves 8.26, 2.70, and 2.15 times longer feature ages compared to Zhu *et al*., EKLT, and h-EKLT, respectively, for [11] dataset. As for the drone dataset [12], our method tracks features for 11.67, 3.18, and 1.79 times longer period than Zhu *et al*., EKLT, and h-EKLT, respectively. The reasons for prolonged feature age of our feature tracker are two-fold:

TABLE I
Quantitative Comparison of Feature Tracking Performance on the Event-Camera Dataset [11]

Dataset	Normalized tracking error (px/s)				Mean feature age (s)			
	Zhu *et al.* [3]	EKLT [7]	h-EKLT	Ours	Zhu *et al.* [3]	EKLT [7]	h-EKLT	Ours
Boxes 6DOF	2.30 (3.33)	0.52 (0.86)	0.96 (2.13)	**0.47** (1.27)	1.45	1.64	2.23	**2.68**
Boxes Rotation	3.32 (3.95)	0.32 (0.55)	0.41 (0.89)	**0.17** (0.73)	1.19	1.71	2.16	**4.28**
Boxes Translation	3.69 (3.21)	0.40 (0.72)	0.64 (1.61)	**0.28** (1.06)	0.87	1.80	2.50	**3.83**
Poster 6DOF	2.87 (3.79)	0.19 (0.66)	0.15 (1.02)	**0.12** (1.06)	1.32	3.55	6.96	**8.73**
Poster Rotation	8.19 (2.13)	0.45 (0.61)	0.29 (0.76)	**0.12** (0.48)	0.26	1.35	2.66	**4.14**
Poster Translation	5.11 (3.58)	**0.17** (0.53)	0.27 (0.90)	**0.17** (0.69)	0.70	3.19	3.39	**3.98**
Shapes 6DOF	12.88 (5.41)	0.21 (0.86)	0.29 (1.04)	**0.12** (0.65)	0.42	4.07	3.54	**5.21**
Shapes Rotation	22.50 (3.60)	0.39 (0.64)	0.90 (1.33)	**0.20** (0.58)	0.16	1.63	1.48	**2.91**
Shapes Translation	12.90 (4.00)	**0.24** (0.59)	0.75 (1.51)	0.25 (0.93)	0.31	2.50	2.02	**3.69**
Dynamic 6DOF	3.45 (3.04)	0.50 (0.53)	0.61 (1.54)	**0.41** (1.24)	0.88	1.05	2.51	**3.04**
Dynamic Rotation	2.29 (2.34)	0.41 (0.62)	0.59 (1.01)	**0.20** (0.60)	1.02	1.51	1.70	**2.98**
Dynamic Translation	4.68 (3.79)	0.50 (0.63)	0.64 (1.28)	**0.41** (1.11)	0.81	1.26	2.00	**2.70**
HDR Boxes	2.82 (3.53)	0.71 (0.76)	1.08 (2.35)	**0.47** (1.12)	1.25	1.07	2.17	**2.40**
HDR Poster	1.62 (1.20)	**0.16** (0.61)	0.19 (0.90)	**0.16** (0.79)	0.74	3.73	4.64	**5.09**
Checkerboard	3.34 (4.61)	0.37 (1.04)	0.59 (1.70)	**0.06** (2.04)	1.38	2.79	2.90	**34.22**

* Numbers in brackets are the mean tracking errors [px].

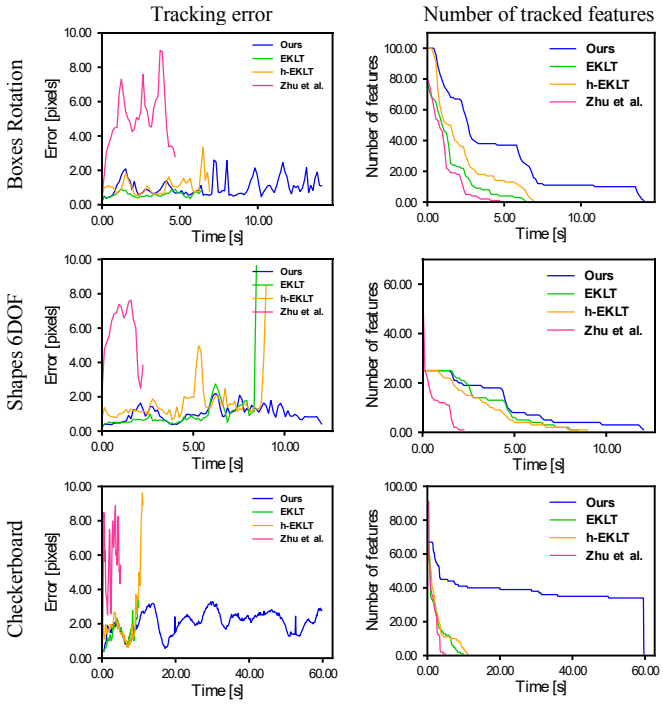

Fig. 4. Graphical comparison of feature tracking results on three sequences from the Event-Camera Dataset [11]

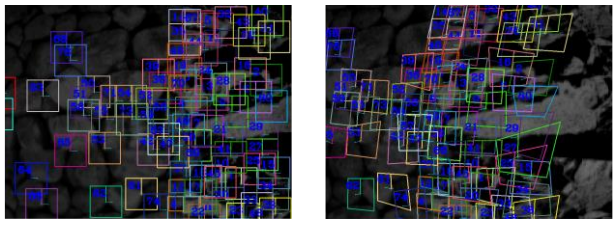

Fig. 5. Visualization of feature patches tracked with our method (a) at initiation and (b) during tracking, on *poster_6dof* sequence of [11]. It is clearly seen that feature patches do warp in 8-DOF when allowed.

higher DOF warping model adopted by our method allows for explanation of general deformation of the feature patches, as shown in Fig. 5, and thus is resilient to challenging motions; our two-step optimization framework provides stability in optimizing the high-DOF warping parameters by not overcomplicating every movement of the feature patches.

The right column of Fig. 4 and Fig. 6 are graphical comparison of number of tracked features over time on three sequences of [11] and [12], respectively. The graphs corroborate the superiority of our method in feature longevity. Our method, in blue, not only tracks features for the longest but also tracks the largest number of features at almost any given time of the sequences. The most striking sequence is the "checkerboard" sequence from [11], with its results shown in the third row of Fig. 4. Graph on the right column shows that our robust feature tracker keeps nearly 40 features out of 66 initiated features until the very end of the sequence, which is 59.79 seconds. This is also reflected in the graph on the left column where our method does not drift and keeps error bounded until the end of the sequence, whereas other methods quickly drift away and terminate their tracking.

D. Computational Efficiency

We evaluate computational efficiency of our method by comparing normalized computational factor against other approaches. Computational factor measures real-time capability by measuring time spent to process a second of real-time; computational factor less than 1 indicates real-time performance. We normalize computational factor since processing time depends on the number of features that the tracker is currently tracking. Normalized computational factor is computed by dividing computational factor with average feature ages of each tracker. We run all experiments

TABLE II
Quantitative Comparison of Feature Tracking Performance on the UZH-FPV Drone Racing Dataset [12]

Dataset	Normalized tracking error (px/s)				Mean feature age (s)			
	Zhu *et al.* [3]	EKLT [7]	h-EKLT	Ours	Zhu *et al.* [3]	EKLT [7]	h-EKLT	Ours
Indoor Forward 3	6.63 (3.58)	**0.16** (0.69)	0.61 (1.44)	0.22 (1.14)	0.54	4.40	2.35	**5.13**
Indoor Forward 5	11.09 (3.77)	0.15 (0.56)	0.22 (1.22)	**0.14** (1.43)	0.34	3.81	5.50	**10.35**
Indoor Forward 6	10.51 (4.10)	**0.39** (0.68)	0.72 (1.89)	0.43 (1.56)	0.39	1.73	2.61	**3.65**
Indoor Forward 7	6.97 (4.46)	0.32 (0.87)	0.32 (1.70)	**0.05** (0.85)	0.64	2.75	5.24	**16.92**
Indoor Forward 9	7.33 (4.62)	0.45 (0.73)	0.60 (1.69)	**0.15** (1.57)	0.63	1.62	2.80	**10.48**
Indoor Forward 10	3.01 (4.93)	0.15 (0.61)	0.26 (1.49)	**0.08** (1.19)	1.64	3.98	5.72	**14.88**
Indoor 45 2	10.06 (3.62)	1.25 (0.79)	1.60 (1.71)	**1.21** (1.36)	0.36	0.63	1.07	**1.12**
Indoor 45 4	6.93 (1.04)	**0.36** (0.49)	0.68 (1.06)	0.57 (0.92)	0.15	1.35	1.56	**1.61**
Indoor 45 9	9.70 (4.17)	0.93 (0.71)	1.02 (1.29)	**0.91** (1.20)	0.43	0.76	1.27	**1.32**
Indoor 45 12	9.81 (4.22)	0.65 (0.80)	2.60 (4.92)	**0.56** (0.98)	0.43	1.23	**1.89**	1.74
Outdoor Forward 1	8.68 (2.17)	2.78 (1.64)	0.53 (1.43)	**0.42** (1.39)	0.25	0.59	2.68	**3.28**
Outdoor Forward 3	4.85 (4.17)	0.40 (0.61)	0.42 (1.41)	**0.24** (1.25)	0.86	1.54	3.33	**5.20**
Outdoor Forward 5	3.08 (3.08)	0.62 (0.68)	0.55 (1.76)	**0.32** (1.43)	1.00	1.09	3.20	**4.42**

* Numbers in brackets are the mean tracking errors [px].

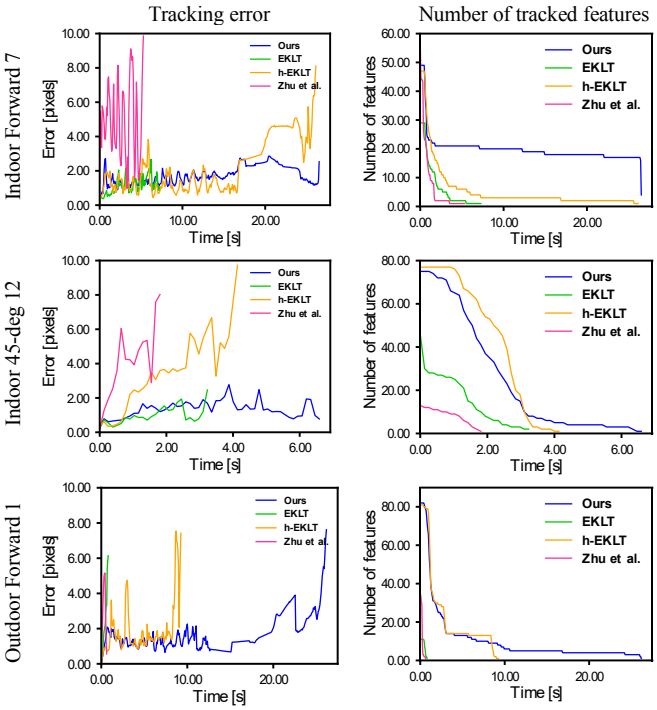

Fig. 6. Graphical comparison of feature tracking results on three sequences from the UZH-FPV Drone Racing Dataset [12].

on a PC with an Intel i7-9700K processor.

Table III shows normalized computational factors of the feature trackers, averaged for all sequences of each dataset of [11] and [12]. Numbers in brackets show mean computational factor before normalization, and results on Zhu *et al.* are grayed out as the source code is written in MATLAB while others are in C++. For both datasets, our method outperforms the compared methods. Although EKLT shows lower mean computational factors, number of features that our approach tracks per epoch are greater, hence our feature tracker shows higher computational efficiency when normalized. Both h-EKLT and our method adopt homography warping of

TABLE III
Quantitative Comparison of Computational Efficiency

Dataset	Normalized computational factor (Hz)			
	Zhu et al. [3]	EKLT [7]	h-EKLT	Ours
[11]	23.65 (20.12)	5.63 (12.32)	5.10 (14.57)	**2.19** (13.14)
[12]	58.47 (34.45)	1.59 (3.11)	3.99 (12.03)	**1.42** (8.78)

* Numbers in brackets are the mean computational factor.

8-DOF, and yet our method prevails. This proves that our two-step optimization process, which optimizes all 8 parameters only when necessary, has great impact on lowering any added computational burden due to high-DOF warping model.

V. Conclusion

In this paper, we present a robust feature tracker that fuses events and frames using 8-DOF warping model via a two-step optimization process. Unlike existing methods with only 3-DOF, we devise a warping model of higher DOF to fully explain the difference between brightness increment patches created by events and frames. When applying the high-DOF model, we separate optimization process into two-fold, and only optimize all 8 parameters in the second step when feature travel distance exceeds a threshold. We conduct a thorough quantitative evaluation on two real datasets, a total of 28 sequences, and demonstrate superiority of our method over state-of-the-art event-utilizing feature trackers on tracking error, feature age, and computational efficiency. For future work, we plan to develop visual odometry framework based on our robust feature tracker proposed in this paper.

Acknowledgment

This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT, the Republic of Korea. (NRF-2022R1A2 C2012166)

REFERENCES

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-state Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[2] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-dof pose tracking for high-speed maneuvers," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2761–2768.

[3] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based feature tracking with probabilistic data association," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4465–4470.

[4] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1964-1980, 2021.

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

[6] 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, 2018.

[7] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "Eklt: Asynchronous photometric feature tracking using events and frames," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 601– 618, 2020.

[8] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 16–23.

[9] Y. Dong and T. Zhang, "Standard and Event Cameras Fusion for Feature Tracking," in *International Conference on Machine Vision Application*, 2021, pp. 55-60.

[10] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[11] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[12] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? The UZH-FPV drone racing dataset," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6713-6719.

[13] Y. J. Kim, "Event and frame based feature tracking for lunar landing navigation," M.S. thesis, Dept. of Aerospace Engineering, Seoul National University, Seoul, Republic of Korea, 2021.

[14] T. -H. Wu, C. Gong, D. Kong, S. Xu, and Q. Liu, "A novel visual object detection and distance estimation method for HDR scenes based on event camera," in *2021 7th International Conference on Computer and Communications (ICCC)*, 2021, pp. 636-640.

[15] A. Tomy, A. Paigwar, K. S. Mann, A. Renzaglia, and C. Laugier, "Fusing event-based and RGB camera for robust object detection in adverse conditions," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 933-939.

[16] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1-9.

[17] H. Cao, G. Chen, J. Xia, G. Zhuang, and A. Knoll, "Fusion-based feature attention gate component for vehicle detection based on event camera," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24540-24548, 2021.

[18] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," *Advances in Neural Information Processing Systems*, *33*, pp. 16639-16652, 2020.

[19] F. Mahlknecht, D. Gehrig, J. Nash, F. M. Rockenbauer, B. Morrell, J. Delaune, and D. Scaramuzza, "Exploring event camera-based odometry for planetary robots," arXiv preprint arXiv:2204.05880v2, 2022.

[20] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5391-5399.

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

[22] 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.

[23] D. Liu, A. Parra, and T. -J. Chin, "Spatiotemporal registration for event-based visual odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4937-4946.

[24] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1425-1440, 2018.

[25] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 593-600, 2017.

[26] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *European Conference on Computer Vision (ECCV)*, 2002, pp. 418-432.

[27] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International joint conference on artificial intelligence (IJCAI)*, pp. 674–679, 1981.

[28] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221– 255, 2004.

[29] Z. Ni, S.-H. Ieng, C. Posch, S. Regnier, and R. Benosman, "Visual tracking using neuromorphic asynchronous event-based cameras," *Neural Computation*, vol. 27, no. 4, pp. 925–953, 2015.

[30] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3867–3876.

[31] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the fourth Alvey Vision Conference*, 1988, vol. 15, pp. 147–151.