BlinkTrack: Feature Tracking over 80 FPS via Events and Images

Yichen Shen[1*] Yijin Li[1,2*] Shuo Chen[1] Guanglin Li[1]
Zhaoyang Huang[2] Hujun Bao[1] Zhaopeng Cui[1] Guofeng Zhang[1†]
[1]State Key Lab of CAD&CG, Zhejiang University [2]Avolution AI
{shenyichen, zhangguofeng}@zju.edu.cn

Abstract

Event cameras, known for their high temporal resolution and ability to capture asynchronous changes, have gained significant attention for their potential in feature tracking, especially in challenging conditions. However, event cameras lack the fine-grained texture information that conventional cameras provide, leading to error accumulation in tracking. To address this, we propose a novel framework, BlinkTrack, which integrates event data with grayscale images for high-frequency feature tracking. Our method extends the traditional Kalman filter into a learning-based framework, utilizing differentiable Kalman filters in both event and image branches. This approach improves single-modality tracking and effectively solves the data association and fusion from asynchronous event and image data. We also introduce new synthetic and augmented datasets to better evaluate our model. Experimental results indicate that BlinkTrack significantly outperforms existing methods, exceeding 80 FPS with multi-modality data and 100 FPS with preprocessed event data. Codes and dataset are available at https://github.com/ColieShen/BlinkTrack.

1. Introduction

Feature tracking aims to estimate the trajectories of query points from a reference timestamp over subsequent periods. It serves as the cornerstone for many computer vision tasks, including structure from motion[54, 55], simultaneous localization and mapping(SLAM)[8, 41, 47–49], object tracking[68, 69, 76], and multi-view stereo[15, 71].

In recent years, event cameras[19, 21, 35, 37, 38] have attracted significant attention within the research community, particularly in the field of feature tracking. Event cameras are innovative sensors that asynchronously detect changes in a scene with very high temporal resolution, capturing events as they occur rather than recording frames at fixed intervals. This capability allows event cameras to

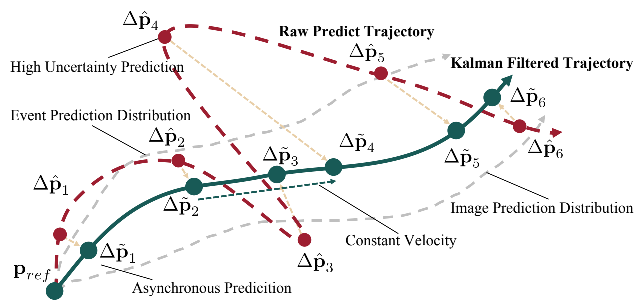

Figure 1. Our method employs the idea of the Kalman filter to associate the asynchronous data from event and image modality and learn the uncertainty-aware fusion.

track features at high frequencies, even in challenging lighting conditions or with fast-moving objects, where conventional cameras are easy to fail[73, 74].

Nevertheless, the event cameras have inherent limitations. For example, event cameras are unable to capture detailed, fine-grained texture information like conventional cameras, and the spatial signal of event cameras could be very sparse in slow-moving scenes. As a result, event-only could lead to error accumulation and inhibit tracking performance. Therefore, in this paper, we aim to explore efficient and powerful feature tracking by fusing the event data and the image frame from conventional cameras.

To achieve our objective, we must address three primary challenges: (1) Data association. Event cameras and standard cameras do not operate in a synchronized manner, with standard cameras capturing at fixed frequency(e.g., 30 Hz[1]) and event cameras detecting changes asynchronously. As a result, it needs an effective method to associate these asynchronous data streams from different modalities. (2) Uncertainty-aware fusion. Event cameras and standard cameras exhibit different distributions of uncertainty due to their distinct sensor characteristics[20]. To achieve effective fusion, the method should be capable of adaptively weighting the contributions of each modality based on confidence estimates. In addition, the uncertainty caused by occlusion

*Yichen Shen, Yijin Li contributed equally to this work.
†Corresponding author.

[1]In this paper, we use FPS as the frequency of algorithm inference, and Hz as the frame rate of the input data.

and other environmental factors also needs to be considered. (3) Runtime efficiency. A promising event-based framework requires minimal runtime overhead to fully exploit the low latency and high temporal resolution from event camera data in scenarios demanding rapid response times, such as autonomous driving and drones.

Previous learning-based methods for feature tracking either adopt a naive fusion strategy[44], resulting in marginal performance gains, or align event data with low-frame-rate image data before fusion, thereby sacrificing the high temporal resolution inherent to event data. In other fields[36, 73, 74], attention-based[62] modules have been explored for implicit alignment and fusion. While these methods can be effective, they do not meet the efficiency requirements for high-frame-rate feature tracking. On the other hand, traditional techniques such as the Kalman filter[31] offer efficient tools for the fusion of asynchronous data. However, they typically require manually tuned parameters, such as carefully adjusting the uncertainty of observations, yet still underperform compared to recent learning-based methods.

Based on these observations, we propose a framework leveraging the Kalman filter technique and extending it into a learning-based paradigm. In our framework, an event module and an image module are employed to independently generate initial tracking predictions from the input data. Additionally, we estimate the prediction uncertainty for each module. Then the Kalman filter integrates these network predictions(treated as observations) according to the respective uncertainties. Since the Kalman filter maintains an internal motion state that evolves dynamically and can be updated with observations at any timestamp, it naturally supports associating the asynchronous predictions from two modules. We train these networks coupled with the Kalman filter in an end-to-end fashion. Supervised by ground-truth tracking labels and proxy visibility labels, the networks learn to generate initial predictions and estimate uncertainties, minimizing the loss by dynamically weighting from historical and current observations of different modalities and obtaining the reliable fused state with the Kalman filter (Fig. 1).

A remaining challenge is computational efficiency: while the Kalman filter is lightweight, the event and image module constitutes the primary computational bottleneck when computing the initial prediction. To this end, we carefully modulate the event and image modules based on state-of-the-art methods[28, 44, 60], ensuring robust tracking while maintaining high computational efficiency.

We also find that existing datasets and benchmarks are overly simple. As a result, they do not fully enable our model to reach its potential or provide a comprehensive evaluation. To address these, we generate a more complex synthetic dataset for training. Additionally, we augment two existing evaluation datasets to introduce more challenging scenarios and ensure a more thorough evaluation.

Our contributions can be outlined as follows. First, we propose a Kalman-filter-based framework for feature tracking that effectively solves the data association and fusion from asynchronous event data and image data. Second, we introduce the carefully modulated event and image module that strikes a good balance between efficiency and performance. Third, we generate new datasets for training and evaluating event-based feature tracking. Exclusive experiments show that the proposed methods outperform existing methods by a large margin and are much more robust in handling occlusions. Besides, it can run at 80 FPS on multi-modality data and over 100 FPS on preprocessed event data.

2. Related Work

Event-based Feature Tracking. Most early works[1, 11, 12, 30, 56, 66] are based on hand-crafted features or optimization. Among them, HASTE[2] and EKLT[21] are more well-known and open source. Deep-EV-Tracker[44] is the first to apply neural networks to event-based feature tracking. It is trained on MultiFlow[26] after deriving the tracking labels from the ground truth optical flow, which leads to inaccurate tracking labels due to quantization error and occlusion. A concurrent work Deep-Ev-Tracker(E2VID)[45] enhances Deep-Ev-Tracker by training with E2VID[51, 52] data but requires significant preprocessing time. Another concurrent work, ETAP[27], achieves excellent tracking performance by applying a CoTracker-like[33] network. However, its heavy model results in a long runtime, diminishing the benefits of high temporal resolution and low latency of event data. Our event module is inspired by Deep-EV-Tracker[44] but modulates the network architecture to achieve a balance between accuracy and efficiency. Besides, we propose a new training dataset that provides accurate tracking ground truth for both occluded and non-occluded pixels, along with visibility labels.

Frame-based Feature Tracking. PIPs[28] is the first learning-based method that directly predicts the tracking trajectory given an RGB sequence. PIPs has inspired a lot of follow-up work[5, 14, 77], among which the CoTracker[33] series is currently the most robust and widely used. However, these methods are time- and resource-consuming, which are incompatible with the low-latency nature of event data. To address this, we design a lightweight image module that can seamlessly cooperate with our event module.

Event and Frame Fusion. The fusion of event and image data has proven effective across various fields[9, 22, 50, 73, 74]. In the area of depth estimation, Gehrig et al.[22] asynchronously encode event or image data into a shared latent feature space; the encoded features are then fused implicitly using an RNN-like predictor. However, the results are not satisfactory, which may be due to the trade-off between the fusion network's capacity and computational efficiency. In

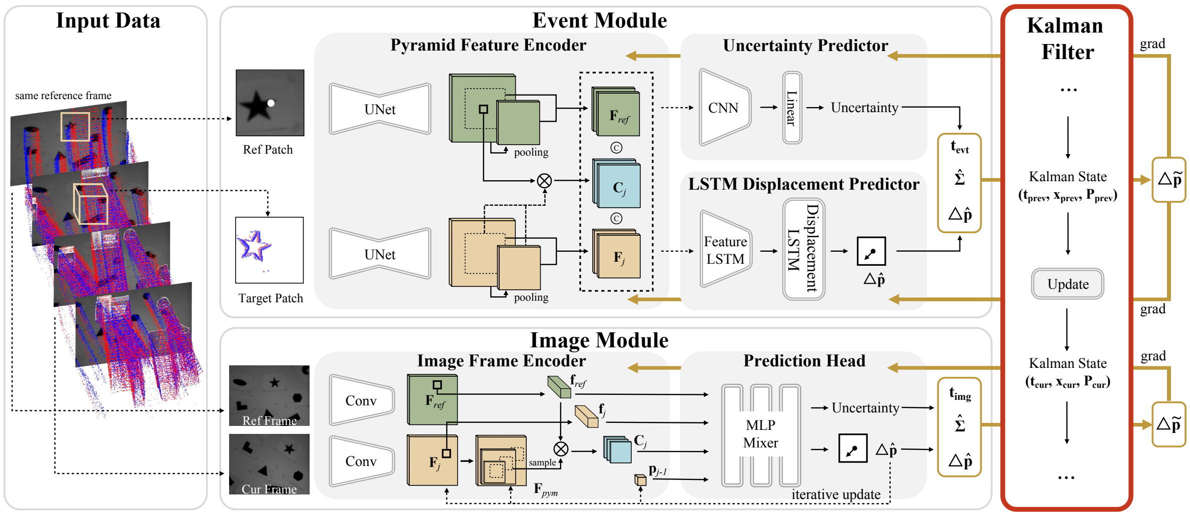

Figure 2. **Overview of the proposed framework.** BlinkTrack employs a Kalman filter to integrate an event module and an image module. The event module takes a reference patch $\mathbf{P}_{evt_{ref}}$ extracted from a grayscale image $\mathbf{I}_{ref}$ and an event patch $\mathbf{P}_{evt_j}$ preprocessed from the event stream at timestamp $\mathbf{t}_{evt_j}$ as input. Separate pyramid encoders first process the two patches to generate pyramid feature maps, which are then used to compute pyramid correlation maps. These feature and correlation maps are concatenated and fed to prediction heads(an uncertainty predictor and an LSTM displacement predictor) to obtain the uncertainty $\hat{\mathbf{\Sigma}}_{evt_j}$ and relative displacement $\Delta\hat{\mathbf{p}}$. The image module operates similarly but encodes the full frame and utilizes an MLP to aggregate diverse data for prediction. Both module predictions are fed into the Kalman filter, which generates the final results $\Delta\tilde{\mathbf{p}}$ with supervised optimal uncertainty. The final integration achieves both high accuracy and efficiency, supporting asynchronous data fusion.

the area of feature tracking, Deep-EV-Tracker[44] naively combines event data and images by initializing the tracking position with results from another modality, leading to limited improvements on its pure event tracking. A concurrent work, FF-KDT[64], accumulates event data to align it with the frame rate of image frames before applying a neural network for fusion. This approach avoids the need for data association but sacrifices the high temporal resolution of event data and results in a low frame rate output. Instead, our method is inspired by the idea of the Kalman filter that effectively and efficiently solves the data association and fusion problem.

Kalman Filter. Kalman filter has been widely used for object tracking[6, 17, 70, 75] and optical flow[10, 18] because it can build an explicit motion model, benefiting the tracking task. The differentiability of the Kalman filter makes it compatible with neural networks. Bao et al.[3, 4] add a Kalman filter on existing optical flow methods, gaining performance promotion. Also, the Kalman filter holds the advantage of fusing different signals, which is used by Wang et al.[65] to reconstruct video using LDR images and event data. This paper investigates Kalman filter techniques in feature tracking and couples them with two modulated modules to develop a powerful and efficient framework.

3. BlinkTrack

Given a query point $\mathbf{p}_{ref}$ at a particular timestamp, BlinkTrack attempts to track the points in subsequent timestamps by matching them with the event stream and the image data. The initial tracks are predicted using the event module and image module, and then the asynchronous predictions from both modules are associated and fused using the Kalman filter technique with timestamp and supervised uncertainty, predicting final position $\tilde{\mathbf{p}} = \mathbf{p}_{ref} + \Delta\tilde{\mathbf{p}}$. An overview of the proposed method, BlinkTrack, is presented in Fig. 2.

3.1. Event Module

Our event module follows previous methods[21, 44, 45] to take a grayscale image patch as the reference and predict the track position by matching with the event patch. The use of image patches as references ensures reference consistency with the image module.

Pyramid Feature Encoder. To minimize computation, we extract the reference patch $\mathbf{P}_{evt_{ref}}$ from the reference grayscale image $\mathbf{I}_{ref}$ at query position $\mathbf{p}_{ref}$, and the event patch $\mathbf{P}_{evt_j}$ from the preprocessed event frame at last predicted position $\mathbf{p}_{j-1}$. Details of the preprocessing steps are provided in the supplementary materials. The reference and event patches are then encoded using two shallow U-Nets, which results in two patch feature maps. To enhance the feature, inspired by DPVO[60], we extract feature maps at different scales from the center of the full feature maps to construct a two-level feature pyramid, enabling multi-scale feature perception. The feature pyramid is then reshaped to the same scale by applying average pooling and concatenated, generating the reference feature map $\mathbf{F}_{evt_{ref}}$ and the event patch feature map $\mathbf{F}_{evt_j}$. Additionally, we compute the correlation $\mathbf{C}_{evt_j}$ correlated between the reference feature vector $\mathbf{f}_{evt_{ref}}$ and the event patch feature map $\mathbf{F}_{evt_j}$, explicitly comparing the similarity between the reference and event features. This reference feature vector $\mathbf{f}_{evt_{ref}}$ is extracted from the center of the reference feature map $\mathbf{F}_{evt_{ref}}$ as shown in Fig. 2.

LSTM Displacement Predictor. Previous methods[23, 44] demonstrated that LSTMs significantly enhance event-based tasks by efficiently extracting temporal features and producing smoother results. Building on this, we employ dual LSTM modules for feature and displacement information propagation. In the feature LSTM module, the $\mathbf{F}_{evt_{ref}}$, $\mathbf{F}_{evt_j}$, and $\mathbf{C}_{evt_j}$ are concatenated and convolved to enhance information exchange before being processed through the ConvLSTM block[58]. The ConvLSTM block propagates previous feature information, with its output subsequently convolved and flattened into the displacement feature vector $\mathbf{f}_{evt_j}$. In the displacement LSTM module, each $\mathbf{f}_{evt_j}$ is concatenated with the hidden displacement feature vector $\mathbf{h}_{j-1}$, leveraging displacement information from preceding frames, and passed through an MLP to obtain a merged displacement vector $\mathbf{m}_j$. The $\mathbf{h}_{j-1}$ and $\mathbf{m}_j$ are then added using a predicted gate weight, yielding the current hidden displacement feature vector $\mathbf{h}_j$. Finally, the $\mathbf{h}_j$ is processed by a linear layer to estimate the displacement $\Delta\hat{\mathbf{p}}$, representing the offset from the query position $\mathbf{p}_{ref}$.

Uncertainty Predictor. This module predicts the uncertainty associated with the estimated displacement, which is incorporated into the Kalman filter to produce an optimal estimation. General uncertainty values range from zero to positive infinity, resulting in a distribution that is too broad and unstable for learning. To address this, the network is designed to learn a normalized uncertainty value within the range [0, 1]. Specifically, the predictor also takes $\mathbf{F}_{evt_{ref}}$, $\mathbf{F}_{evt_j}$ and $\mathbf{C}_{evt_j}$ as input, concatenates them, and processes them through CNN with five convolution layers to transform the feature patch to an uncertainty feature vector $\mathbf{f}_{uncert_j}$. A linear layer generates two scores representing "certain" and "uncertain," followed by a Softmax function to obtain the normalized uncertainty $\hat{\sigma}$, which is subsequently remapped by a parabolic function to range [0, 10], and then repeated and placed on the diagonal of the covariance matrix $\hat{\mathbf{\Sigma}}_{evt_j} \in \mathbb{R}^{2\times 2}$. This matrix is fed into the Kalman filter to represent uncertainty.

3.2. Image Module

Event cameras are not good at measuring fine-grained texture information like traditional cameras, which can easily lead to error accumulation and track loss. To enhance long-term tracking performance and minimize cumulative errors, we developed a lightweight image frame relocalization module with vast receptive fields. The design of our module is inspired by recent works on point tracking, such as PIPs[28]. However, these existing methods operate slowly, limiting overall processing speed despite the high speed of the event module. To address this, we simplified the architecture design, enabling our module to achieve operation speeds exceeding 50 FPS.

Image Frame Encoder. The image module takes a reference frame $\mathbf{I}_{ref}$ and a target frame $\mathbf{I}_j$ input, both images are encoded by a Pyramid Encoder like RAFT[59] feature map $\mathbf{F}_{img_{ref}}$ and $\mathbf{F}_{img_j}$. This encoder effectively reduces computational resources and time while simultaneously increasing the receptive field through convolution layers, which significantly aids in relocalization. We apply bilinear sampling to the reference feature map $\mathbf{F}_{img_{ref}}$ at the given reference point $\mathbf{p}_{ref}$ to obtain the reference feature vector $\mathbf{f}_{img_{ref}}$. The feature vector $\mathbf{f}_{img_j}$ is also sampled from $\mathbf{F}_{img_j}$ at last prediction $\mathbf{p}_{j-1}$.

To evaluate feature similarity explicitly, the feature map $\mathbf{F}_{img_j}$ is processed through several average pool layers to multiple scales and correlated with reference feature vector $\mathbf{f}_{img_{ref}}$, constructing the correlation pyramid $\mathbf{C}_{pym_j}$. From last predicted position $\mathbf{p}_{j-1}$, we sample patches $\mathbf{P}_{img_j}$ from multiple scale correlation pyramid $\mathbf{C}_{pym_j}$ and concatenate them to the final correlation map $\mathbf{C}_{img_j}$. This approach facilitates the acquisition of multi-scale information, further broadening the receptive field and ensuring both accuracy and breadth of predictions.

Prediction Head. To involves most information, $\mathbf{f}_{img_{ref}}$, $\mathbf{f}_{img_j}$, $\mathbf{C}_{img_j}$, $\mathbf{p}_{j-1}$ and the embedded $\mathbf{p}_{j-1}$ are gathered together as input to a MLP-Mixer[61] inspired from PIPs[28] to predict displacement $\Delta\hat{\mathbf{p}}$, "certain" and "uncertainty" scores. The scores would be processed the same as the event module to calculate $\hat{\mathbf{\Sigma}}_{img_j}$.

Given the potential distance between the current and actual positions, this procedure can be iteratively performed by extracting $\mathbf{f}_{img_j}$, $\mathbf{C}_{img_j}$ using the newly predicted $\mathbf{p}_j$ from $\mathbf{C}_{pym_j}$. This allows for progressively obtaining the most accurate and confident prediction. Finally, the output $\Delta\hat{\mathbf{p}}$ and $\hat{\mathbf{\Sigma}}_{img_j}$ are integrated into the Kalman filter to generate the final prediction.

3.3. Kalman Filter

We employ a Kalman filter to associate and fuse asynchronous data from event and image modality. The Kalman filter design offers several advantages. First, it allows measurements from any module at any timestamp, enabling the seamless fusion of multi-modal data arriving at different rates. Second, its process is fully differentiable and allows the network to learn to minimize estimation errors by effectively integrating reliable measurements while mitigating the influence of noise(e.g., occlusion) through uncertainty supervision. Third, it requires minimal computational resources and maintains low coupling between the event and image modules, enabling modular architectures that better align with the data and support efficient fusion.

In our framework, the Kalman filter observation requires only the position $\Delta\hat{\mathbf{p}}$, covariance matrix $\hat{\mathbf{\Sigma}}$, and timestamp t as inputs. Consequently, measurements can originate from either the event or image module, seamlessly fusing both modules and predicting the filtered final position $\Delta\tilde{\mathbf{p}}$. We assume a simple constant velocity model in the Kalman filter by defining state x, state covariance matrix P.

$$\begin{aligned} x &= (\mathrm{x}, \mathrm{y}, v_x, v_y)^{\mathrm{T}} \\ P &= \begin{pmatrix} \rho_{xx} & 0 & \rho_{xv_x} & 0 \\ 0 & \rho_{yy} & 0 & \rho_{yv_y} \\ \rho_{v_x x} & 0 & \rho_{v_x v_x} & 0 \\ 0 & \rho_{v_y y} & 0 & \rho_{v_y v_y} \end{pmatrix} \end{aligned} \tag{1}$$

The state x contains the position and velocity of x and y, and the state covariance matrix P represents the uncertainty of the internal state. The process in the Kalman filter is traditional[32, 67]. As a new measurement arrives, the Kalman filter performs a prediction step to propagate the internal state and uncertainty to the current timestamp, relying solely on historical information. The internal state and uncertainty are then updated using the current measurement, incorporating $\Delta\hat{\mathbf{p}}$ and $\hat{\mathbf{\Sigma}}$ from our event or image module.

$$\begin{aligned} \hat{x}_{k|k-1}, P_{k|k-1} &= \text{Predict}(\hat{x}_{k-1|k-1}, P_{k-1|k-1}), \\ \hat{x}_{k|k}, P_{k|k} &= \text{Update}(z_k, R_k, \hat{x}_{k|k-1}, P_{k|k-1}). \end{aligned} \tag{2}$$

z is the new measurement(our predicted $\Delta\hat{\mathbf{p}}$), R is the measurement noise covariance(our predicted $\hat{\mathbf{\Sigma}}$), and our final prediction $\Delta\tilde{\mathbf{p}} = (\mathrm{x}, \mathrm{y})$ is from the first two elements in the vector $\hat{x}_{k|k}$. Throughout the Kalman filter full process, the measurements $\Delta\hat{\mathbf{p}}_k$ are fused with previous state $\hat{x}_{k-1|k-1}$ weighted by $\hat{\mathbf{\Sigma}}$ and $P_{k-1|k-1}$ to get final state $\hat{x}_{k|k}$(i.e., $\Delta\tilde{\mathbf{p}}_k$ and velocity), while simultaneously maintaining velocity (v_x, v_y) and the state covariance matrix $P_{k|k}$. Both modules are expected to predict high uncertainty in challenging scenarios, such as occlusions, and low uncertainty in confident conditions. When $\hat{\mathbf{\Sigma}}$ indicates high uncertainty, the Kalman filter relies more on its internal constant velocity model to predict $\Delta\tilde{\mathbf{p}}$. Conversely, when $\hat{\mathbf{\Sigma}}$ indicates low uncertainty, $\Delta\hat{\mathbf{p}}$ is assigned greater weight in determining $\Delta\tilde{\mathbf{p}}$. Moreover, our predicted $\hat{\mathbf{\Sigma}}$ can be directly supervised through this process, rather than being traditionally handcrafted, significantly improving the quality of the uncertainty. This process filters out unreliable predictions and non-uniform noise, ensuring a stable and accurate trajectory with minimal computation. More details of operation and definitions in the Kalman filter are provided in the supplementary material.

3.4. Supervision

Joint training of the event and image modules would require excessive computational resources, so we opted to train the two modules separately. Moreover, we empirically observe that training the entire event module from scratch is unstable and slow, due to cumulative error and the limited receptive field of the event module, which covers only a few dozen pixels. Therefore, we initially train the event module without the uncertainty predictor and Kalman filter, using only the event displacement loss calculated from the direct predictions $\Delta\hat{\mathbf{p}}$ of the event module. After that, we freeze the encoder and displacement predictor, then train the uncertainty predictor by incorporating the differentiable Kalman filter, using both displacement loss calculated from Kalman filter prediction $\Delta\tilde{\mathbf{p}}$ and event uncertainty loss. The training is conducted on our proposed dataset, MultiTrack (see Sec. 4.1). While MultiTrack provides a ground truth trajectory for any pixel, we use SuperPoint[13] to extract keypoints as the query points. In the following, we introduce the two loss functions used to train the event module. The training loss for the image module is similar and detailed in the supplementary material.

Event Displacement Loss. The displacement loss measures the L1 distance between the predicted displacement $\Delta\hat{\mathbf{p}}$ and ground truth displacement $\Delta\mathbf{p}$. It is important to note that due to error accumulation, the ground truth tracking position may fall outside the search region(i.e., the event patch $\mathbf{P}_{evt_j}$). In such cases, we truncate the loss, formulated as follows:

$$\mathcal{L}_{\hat{disp}} = \begin{cases} \|\, \Delta\hat{\mathbf{p}} - \Delta\mathbf{p} \,\|_1, & \|\, \Delta\mathbf{p} \,\|_1 < r \\ 0. & \text{else} \end{cases} \tag{3}$$

Additionally, we perform on-the-fly augmentation[44], applying random affine transformations to the event patch.

Event Uncertainty Loss. The uncertainty can be indirectly supervised through displacement loss on $\Delta\tilde{\mathbf{p}}$ by incorporating the Kalman filter. To boost convergence, we also apply visibility loss as a proxy directly on the predicted normalized uncertainty, as visibility probability is strongly correlated with uncertainty. Specifically, we use cross-entropy loss:

$$\mathcal{L}_{vis} = \text{CrossEntropy}(1 - \hat{\sigma}, g), \tag{4}$$

where g is the ground truth visibility. We use these two loss functions to train the uncertainty predictor: $\mathcal{L}_{event} = \mathcal{L}_{\tilde{disp}} + w_1 \mathcal{L}_{vis}$, where $w_1 = 2$.

4. MultiTrack, EC-occ and EDS-occ

As manually labeling visibility ground truth is highly challenging, we introduce a data synthesis pipeline, see Fig. 3. Our pipeline can synthesize color images, events, occluded tracks, and visibility status with adjustable parameters, or generate these data from existing datasets, which require only raw color images and specified pixel trajectories.

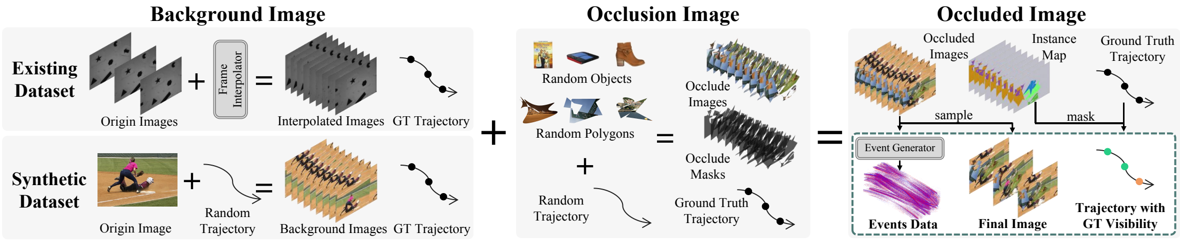

Figure 3. **Overview of occluded data synthetic pipeline.** The high frame rate occlusion images are overlaid on the high frame rate background images to compose occluded images, which are used to generate synthetic event data.

4.1. MultiTrack

Following MultiFlow[26], our generated dataset is based on the premise that the background exhibits regular motion, whereas certain foreground objects present rapid motion and occlusion. The background images are selected from the Flickr 30k dataset[72], while the foreground images are of two types to provide more comprehensive features and occlusions. The first type consists of objects with transparent backgrounds from Google Scanned Objects[16], the other type has random polygons or smoothed shapes with random images hatched in these shapes. The shapes are inspired by AutoFlow[43], and the hatched images are from COCO2014[40]. With random translation, rotation, and scale of each image, the foreground images are overlaid together on background images, producing synthetic high frame rate(1000 Hz) images, instance maps, and dense complex tracks with visibility. The high frame rate images are taken as input to DVS-Voltmeter[39] to generate synthetic event data while they are also sampled in a normal frame rate to get a color image. By using this pipeline, we generate MultiTrack as our training data.

4.2. EC-occ, EDS-occ

When applying occlusion on an existing dataset, we first interpolate the images using FILM[53], which iterates 4 times, inserting 15 frames for each color frame interval(final 384 Hz for EC[46] and 1200 Hz for EDS[29]). With interpolated high frame rate background images, the same procedures are applied to create occlusion on the existing dataset, obtaining occluded event data from DVS-Voltmeter[39] and visibility of given tracks from the instance map. By applying this pipeline, EC[46] and EDS[29] are occluded and processed to EC-occ and EDS-occ.

5. Experiment

Dataset. We test our method in a commonly used real captured dataset, Event Camera dataset(EC)[46], and Event-aided Direct Sparse Odometry dataset(EDS)[29]. EC provides 240×180 pixels 24Hz APS frames and events, which is recorded by a DAVIS240C camera[7] while EDS contains 640×480 pixels 75Hz image data and event data captured with a beam splitter setup[29]. The events from EC and EDS are pre-processed to SBT-Max[63] with intervals of 10ms and 5ms, so that the input event data are in frequency of 100Hz and 200Hz, respectively. These datasets do not originally offer the ground truth for feature tracking, and Messikommer et al.[44] extend it by calculating the ground truth through KLT[57] tracking and triangulation. We also test in our synthetic dataset with occlusion, EC-occ, and EDS-occ introduced at Sec. 4. Another widely used event-based dataset, DSEC[24, 25], is also used in qualitative comparison to evaluate performance on large outdoor scenes.

Baselines. We conducted experiments against the state-of-the-art method Deep-EV-Tracker[44], which tracks event frames using grayscale image references with smaller patches. We also obtained experimental data from the concurrent method Deep-Ev-Tracker(E2VID)[45]. We also compare with EKLT[21], which has a similar pipeline with Deep-EV-Tracker but without using a neural network. Additionally, HASTE[2], which demands no image but uses pure event data, is included. Additionally, to demonstrate our novel multi-modality combination technique, we follow Deep-EV-Tracker[44], combining it with KLT[57], a well-established color frame tracker, by replacing the initial position with predictions from another module. Our fusion method without a Kalman filter follows the same approach. We also evaluate FF-KDT[64]; however, it produces estimates only at image timestamps, whereas Deep-Ev-Tracker[44] and our method generate significantly more predictions at event timestamps.

Metric. The EC[46] and EDS[29] only provide ground truth trajectories with 24Hz and 75Hz. To evaluate the high-rate event predictions with 100Hz and 200Hz, we interpolate the event predictions to continuous trajectories and sample points on those timestamps that have ground truth data following Messikommer et al.[44]. We evaluate feature age(FA), defined as the duration for which a feature exceeds a specified distance(1, 2, 3 ..., 32) from the ground truth, and expect feature age(Exp FA or Expect FA), which is calculated as the product of feature age and the ratio of stable tracks—those maintaining errors below the threshold across all frames. These two metrics are common in event-based feature tracking benchmarks[44]. We also evaluate

Data	Methods	EC FA↑	EC Exp FA↑	EC N_p↑	EDS FA↑	EDS Exp FA↑	EDS N_p↑
E	HASTE[2]	0.442	0.427	-	0.096	0.063	-
	EKLT[21]	0.811	0.775	-	0.325	0.205	-
	D-Tracker[44]	0.795	0.787	100	0.549	0.451	200
	D-Tracker(E2VID)[45]	0.793	0.781	100	**0.579**	**0.482**	200
	Ours(E)	$\underline{0.833}$	$\underline{0.819}$	100	0.568	0.474	200
	Ours(E w. K)	**0.835**	**0.820**	100	$\underline{0.569}$	$\underline{0.475}$	200
E + I	FF-KDT[64]	0.852*	0.846*	24	0.526*	0.431*	75
	D-Tracker[44]+KLT[57]	0.735	0.730	124	0.594	0.503	275
	Ours(E+I)	$\underline{0.786}$	$\underline{0.780}$	124	$\underline{0.634}$	$\underline{0.532}$	275
	Ours(E+I w. K)	**0.851**	**0.845**	124	**0.653**	**0.550**	275

Table 1. **Performance evaluation on EC and EDS.** "D-tracker" denotes Deep-EV-Tracker[44]. "E" denotes event, "I" denotes image and "K" denotes Kalman. N_p denotes the number of predictions per second of data. HASTE[2] and EKLT[21] operate on raw event data without frames, resulting in an unstable number of predictions. * FF-KDT[64] results are not compared due to their limited prediction number.

accuracy through δ_{avg}[14], which is the average fraction of points tracked within 1, 2, 4, 8, and 16 pixels, also widely used in point tracking benchmarks[33, 77].

Implementation Details. As mentioned in Sec. 3.4, our event module is pre-trained on the MultiTrack dataset on 30000 feature tracks from 2000 sequences employing a continual learning approach[44]. We use ADAM[34] optimizer with a learning rate of 1×10^{-4} and 140000 training steps in total with batch size 32. To initialize training, the input sequences are clipped to 4 frames, which would increase to 12 and 23 after 80000 and 120000 steps[44]. Then we train our uncertainty predictor on the same dataset with the same hyperparameters for 80000 steps. At the same time, we train our image module on MultiTrack, which is generated for image training with 10000 sequences using ADAMW[42] with a learning rate of 5×10^{-4} and batch size 16. The sequence length is fixed to 10 in all 50000 training steps. We run both modules' supervision on two NVIDIA RTX3090 24GB GPUS for 48 hours and experiments on one NVIDIA RTX3090 GPU and Intel(R) Xeon(R) Gold 6139M CPU.

5.1. Experiments Results

Real Data Experiment. We compare our method with other state-of-the-art methods in terms of the input signals, i.e., event-only, and both event and image, in Tab. 1. Note that even though both EC[46] and EDS[29] only contain static scenarios scenes with almost no occluded trajectories, which are rather simple and do not fully reveal the superiority of our learnable Kalman filter, our method still presents state-of-the-art performance on both tracks and obtains high-frequency predictions when fusing the event and image signals. Due to their simplicity, a naive LSTM within the event module can capture contextual information across frames implicitly, and our proposed learnable Kalman filter does not bring a prominent performance gain.

Occlusion Data Experiment. To better demonstrate the superiority of the Kalman filter, experiments are conducted

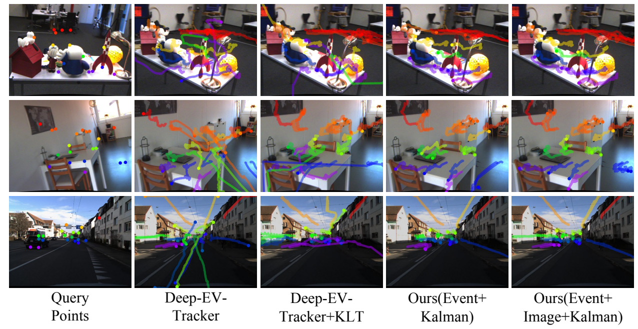

Figure 4. **Qualitative comparison on EDS[29](top two rows) and DSEC[24](bottom row).**

on more challenging datasets with occlusions, EC-occ, and EDS-occ, see Tab. 2. The improved performance of modules incorporating the Kalman filter demonstrates its effectiveness in stabilizing tracking. This is particularly evident in occluded scenarios, where occluded points exhibit the most significant performance gains, confirming that the Kalman filter mitigates instabilities and corrects erroneous measurements caused by noise.

Both experiments demonstrate that naively combining the two modules by replacing initial points does not yield satisfactory results and may even degrade performance due to the compounding limitations of both. However, with the aid of the Kalman filter, significant improvements are observed, demonstrating its effectiveness in combining and optimizing asynchronous multimodal predictions. In detail, uncertainty increases in challenging scenarios for event or traditional cameras, such as extreme lighting or motion blur for traditional cameras and low-texture environments for event cameras. In these cases, the Kalman filter mitigates the impact of biased predictions by accounting for high uncertainty. Conversely, when cameras operate in optimal conditions, uncertainty remains low, allowing predictions to have a greater influence on the Kalman filter's output. This process harnesses the strengths of both event and image cameras while mitigating their limitations, resulting in a more balanced and accurate final prediction.

Qualitative Comparison. We visualize the tracks estimated by our methods and other methods in Fig. 4. As shown, all event-based methods exhibit some degree of error; however, our event module produces fewer lost tracks compared to the Deep-EV-Tracker[44], indicating greater robustness. With the integration of the image module, our method achieves improved performance, demonstrating that the image module is effective in enhancing relocalization and accuracy. When comparing the multi-modal results, a simple fusion approach, such as replacing initial points, results in a higher number of lost tracks, whereas Kalman fusion proves to be more stable and reliable.

Runtime Analysis. Our event module takes less than 9 ms to track one event patch across one preprocessed event

Data	Metric	EC-occ					EDS-occ				
		FF-KDT[64]	D-tracker[44]	**Ours(E)**	**Ours(E+K)**	Incr(%)↑	FF-KDT[64]	D-tracker[44]	**Ours(E)**	**Ours(E+K)**	Incr(%)↑
E	$\delta^{vis}\uparrow$	-	26.8	29.3	**30.5**	3.92	-	31.3	35.1	**36.7**	4.50
	$\delta^{occ}\uparrow$	-	20.2	23.1	**24.1**	**4.41**	-	21.3	24.6	**25.9**	**5.05**
	$\delta^{all}\uparrow$	-	26.3	28.7	**29.8**	3.95	-	30.9	34.6	**36.2**	4.51
		FF-KDT[64]	D-tracker[44] + KLT[57]	**Ours (E+I)**	**Ours (E+I w. K)**	Incr(%)↑	FF-KDT[64]	D-tracker[44] + KLT[57]	**Ours (E+I)**	**Ours (E+I w. K)**	Incr(%)↑
E+I	$\delta^{vis}\uparrow$	28.2	32.1	37.2	**44.5**	19.73	24.0	35.0	43.1	**52.0**	20.64
	$\delta^{occ}\uparrow$	14.5	5.8	11.4	**28.5**	**149.06**	11.5	10.7	18.6	**26.8**	**43.61**
	$\delta^{all}\uparrow$	27.1	30.3	35.5	**43.4**	22.32	23.4	33.7	41.6	**50.6**	21.51

Table 2. **Performance increase from module without Kalman filter to module with Kalman filter.** "Incr" denotes the performance increase from our module without a Kalman filter to those with a Kalman filter.

Expect FA↑						
Experiment	Method	EC	EC-occ	EDS	EDS-occ	Paras
Feature Pyramid (Ours Event)	On	**0.819**	**0.522**	**0.474**	**0.343**	33.1M
	Off	0.789	0.439	0.395	0.283	29.3M
LSTM Module (Ours Event)	Feature+Displace	**0.819**	**0.522**	**0.474**	**0.343**	33.1M
	Feature	0.794	0.473	0.428	0.337	32.7M
	Displacement	0.617	0.332	0.282	0.186	31.9M
	Off	0.568	0.338	0.294	0.225	31.6M
Correlation Vector (Ours Event)	Feature Map Center	**0.819**	**0.522**	**0.474**	**0.343**	33.1M
	U-Net Bottleneck	0.792	0.472	0.441	0.330	35.7M

Table 3. **Ablation experiments.** The ablated modules are labeled under the experiments. Settings used in our final model are underlined.

frame on EC[46] and EDS[29] using an NVIDIA RTX3090. The introduction of the Kalman filter adds less than 1 ms to the budget. The preprocessing of event data takes 3ms on EC (240×180 resolution, 10ms frame interval) and 4ms on EDS (640×480 resolution, 5ms frame interval). It is executed on the CPU and can run in parallel with the tracking model. Deep-Ev-Tracker[44] requires identical preprocessing with ours, while Deep-Ev-Tracker(E2VID)[45] processes each frame in 153ms[45], and FF-KDT[64] takes 13ms on EC and 79ms on EDS. As a result, our integration of the event module and Kalman filter enables precise predictions at frequencies exceeding 100 FPS, while Deep-Ev-Tracker[44] only achieves 50 FPS. Additionally, our image module can operate efficiently with the Kalman filter at frequencies above 50 FPS, making it efficient enough for low-frequency image relocalization.

If the image module and event module run sequentially, with the image module operating at intervals of 0.05s and producing 20 predictions per second, it accounts for 0.4s of the runtime. The event module then uses the remaining 0.6s to produce at least 60 predictions. Consequently, the entire model generates 80 predictions per second, while FF-KDT[64] achieves only 7 FPS.

5.2. Ablations

To evaluate the distinct contributions of each integrated network component on tracking accuracy, we conducted a series of ablation studies as outlined in Tab. 3.

Feature Pyramid. The feature pyramid is used to take both the perception field and accuracy into account. We conducted experiments replacing the pyramid feature encoder with a simple patch encoder[44], which resulted in reduced performance due to the limited information from the encoder. This outcome suggests that the Feature Pyramid enhances tracking accuracy and effectively handles occlusions.

LSTM Module. Our event module utilizes two LSTM modules, specifically designed to convey previous feature and displacement information. We evaluated variants with one LSTM module removed and another without any LSTM modules. The results indicate that removing either LSTM module results in a decline in performance, underscoring the contributions of both modules.

Correlation Vector. The Deep-EV-Tracker[44] uses the U-Net bottleneck vector as the reference vector for correlating with the target feature map. However, we propose that the reference vector should be derived from the reference feature map to maintain symmetry. Our experiment indicates that using the vector from the reference feature map yields better performance.

6. Conclusion

In this paper, we presented BlinkTrack, a novel framework for high-frame-rate feature tracking that leverages the strengths of both event data and images. By integrating a differentiable Kalman filter within a learning-based architecture, BlinkTrack effectively addresses the challenges of asynchronous data fusion and improves tracking performance under multimodal data with noise(e.g., occlusion). Our extensive experiments, supported by newly generated and augmented datasets, show that BlinkTrack significantly outperforms existing methods in terms of robustness and speed, achieving state-of-the-art performance while running at over 80 FPS with multi-modality data and 100 FPS with preprocessed event data. These results underscore the potential of our approach for advanced feature tracking applications, establishing a new benchmark for future research in this field. **Limitations.** One limitation of the current performance may be the network capacity and the receptive field. In future work, we aim to investigate strategies for scaling up the network parameters while maintaining efficiency(e.g., achieving over 100 FPS).

Acknowledgements: This work was partially supported by the NSF of China(No.62425209).

References

[1] Ignacio Alzugaray and Margarita Chli. Ace: An efficient asynchronous corner tracker for event cameras. In *International Conference on 3D Vision (3DV)*, pages 653–661, 2018. 2

[2] Ignacio Alzugaray and Margarita Chli. Haste: multi-hypothesis asynchronous speeded-up tracking of events. *British Machine Vision Conference*, 2020. 2, 6, 7

[3] Wenbo Bao, Xiaoyun Zhang, Li Chen, and Zhiyong Gao. Kalmanflow: Efficient kalman filtering for video optical flow. In *IEEE International Conference on Image Processing (ICIP)*, pages 3343–3347, 2018. 3

[4] Wenbo Bao, Xiaoyun Zhang, Li Chen, and Zhiyong Gao. Kalmanflow 2.0: Efficient video optical flow estimation via context-aware kalman filtering. *IEEE Transactions on Image Processing*, 28(9):4233–4246, 2019. 3

[5] Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yitong Dong, Yijin Li, and Hongsheng Li. Context-pips: persistent independent particles demands spatial context features. In *International Conference on Neural Information Processing Systems*, 2023. 2

[6] S.S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999. 3

[7] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 6

[8] Carlos Campos, Richard Elvira, Juan J. Gomez, Jose M. M. Montiel, and Juan D. Tardos. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 1

[9] Hu Cao, Zehua Zhang, Yan Xia, Xinyi Li, Jiahao Xia, Guang Chen, and Alois Knoll. Embracing events and frames with hierarchical feature refinement network for object detection. In *European Conference on Computer Vision (ECCV)*, page 161–177, 2024. 2

[10] T.M. Chin, W.C. Karl, and A.S. Willsky. Probabilistic and sequential computation of optical flow using temporal coherence. *IEEE Transactions on Image Processing*, 3(6):773–788, 1994. 3

[11] Jason Chui, Simon Klenk, and Daniel Cremers. Event-based feature tracking in continuous time with sliding window optimization. *CoRR*, abs/2107.04536, 2021. 2

[12] Laurent Dardelet, Ryad Benosman, and Sio-Hoi Ieng. An event-by-event feature detection and tracking invariant to motion direction and velocity, 2021. 2

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. *CoRR*, abs/1712.07629, 2017. 5

[14] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *CoRR*, abs/2211.03726, 2022. 2, 7

[15] Yitong Dong, Yijin Li, Zhaoyang Huang, Weikang Bian, Jingbo Liu, Hujun Bao, Zhaopeng Cui, Hongsheng Li, and Guofeng Zhang. A global depth-range-free multi-view stereo transformer network with pose embedding. *International Conference on Neural Information Processing Systems*, 2024. 1

[16] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas Barlow McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 6

[17] Bertil Ekstrand. Some aspects on filter design for target tracking. *Journal of Control Science and Engineering*, 2012. 3

[18] P. Elad and A. Feuer. Recursive optical flow estimation-adaptive filtering approach. In *Convention of Electrical and Electronics Engineers in Israel*, pages 387–390, 1996. 3

[19] Guillermo Gallego, Jon E.A. Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2402–2412, 2018. 1

[20] Guillermo Gallego, Tobi Delbrück, 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. 1

[21] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Eklt: Asynchronous photometric feature tracking using events and frames. *International Journal of Computer Vision*, 128:1–18, 2020. 1, 2, 3, 6, 7

[22] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters*, 6(2):2822–2829, 2021. 2

[23] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13884–13893, 2023. 4

[24] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. 6, 7

[25] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *International Conference on 3D Vision (3DV)*, 2021. 6

[26] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12, 2024. 2, 6

[27] Han Han, Wei Zhai, Yang Cao, Bin Li, and Zhengjun Zha. Event-based tracking any point with motion-augmented temporal consistency. *CoRR*, abs/2412.01300, 2024. 2

[28] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 4

[29] Javier Hidalgo-Carrio, Guillermo Gallego, and Davide Scaramuzza. Event-aided direct sparse odometry. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. 6, 7, 8

[30] Sumin Hu, Yeeun Kim, Hyungtae Lim, Alex Junho Lee, and Hyun Myung. ecdt: Event clustering for simultaneous feature detection and tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3808–3815, 2022. 2

[31] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 2

[32] Rudolf E. Kálmán and Richard S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83:95–108, 1961. 5

[33] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *CoRR*, abs/2307.07635, 2023. 2, 7

[34] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 7

[35] Yijin Li, Han Zhou, Bangbang Yang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[36] Yijin Li, Xinyang Liu, Wenqi Dong, Han Zhou, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. Deltar: Depth estimation from a light-weight tof sensor and rgb image. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[37] Yijin Li, Zhaoyang Huang, Shuo Chen, Xiaoyu Shi, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Blinkflow: A dataset to push the limits of event-based optical flow estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023. 1

[38] Yijin Li, Yichen Shen, Zhaoyang Huang, Shuo Chen, Weikang Bian, Xiaoyu Shi, Fu-Yun Wang, Keqiang Sun, Hujun Bao, Zhaopeng Cui, Guofeng Zhang, and Hongsheng Li. Blinkvision: A benchmark for optical flow, scene flow and point tracking estimation using rgb frames and events. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[39] Songnan Lin, Ye Ma, Zhen-Qiang Guo, and Bihan Wen. Dvs-voltmeter: Stochastic process-based event simulator for dynamic vision sensors. In *European Conference on Computer Vision (ECCV)*, 2022. 6

[40] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 6

[41] Xinyang Liu, Yijin Li, Yanbin Teng, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 7

[43] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *CoRR*, abs/1801.06397, 2018. 6

[44] Nico Messikommer, Carter Fang, Mathias Gehrig, and Davide Scaramuzza. Data-driven feature tracking for event cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5642–5651, 2023. 2, 3, 4, 5, 6, 7, 8

[45] Nico Messikommer, Carter Fang, Mathias Gehrig, Giovanni Cioffi, and Davide Scaramuzza. Data-driven feature tracking for event cameras with and without frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12, 2025. 2, 3, 6, 7, 8

[46] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbrück, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *CoRR*, abs/1610.08336, 2016. 6, 7, 8

[47] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[48] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[49] Junjie Ni, Yijin Li, Zhaoyang Huang, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pats: Patch area transportation with subdivision for local feature matching. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023. 1

[50] Roberto Pellerito, Marco Cannici, Daniel Gehrig, Joris Belhadj, Olivier Dubois-Matra, Massimo Casasco, and Davide Scaramuzza. Deep visual odometry with events and frames. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8966–8973, 2023. 2

[51] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[52] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[53] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *European Conference on Computer Vision (ECCV)*, 2022. 6

[54] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), 2016. 1

[55] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[56] Hochang Seok and Jongwoo Lim. Robust feature tracking in dvs event stream using bézier mapping. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1647–1656, 2020. 2

[57] Jianbo Shi and Tomasi. Good features to track. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994. 6, 7, 8

[58] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: a machine learning approach for precipitation nowcasting. In *International Conference on Neural Information Processing Systems*, page 802–810, Cambridge, MA, USA, 2015. 4

[59] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. *CoRR*, abs/2003.12039, 2020. 4

[60] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *CoRR*, abs/2208.04726, 2022. 2, 3

[61] Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *CoRR*, abs/2105.01601, 2021. 4

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 2

[63] Lin Wang, I.S. Mohammad Mostafavi, Yo-Sung Ho, and Kuk-Jin Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10073–10082, 2019. 6

[64] Xiangyuan Wang, Huai Yu, Lei Yu, Wen Yang, and Gui-Song Xia. Towards robust keypoint detection and tracking: A fusion approach with event-aligned image features. *IEEE Robotics and Automation Letters*, 2024. 3, 6, 7, 8

[65] Ziwei Wang, Yonhon Ng, Cedric Scheerlinck, and Robert Mahony. An asynchronous kalman filter for hybrid event cameras. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[66] Ziwei Wang, Timothy Molloy, Pieter van Goor, and Robert E. Mahony. Event blob tracking: An asynchronous real-time algorithm. *CoRR*, abs/2307.10593, 2023. 2

[67] Greg Welch. An introduction to the kalman filter. In *International Conference on Computer Graphics and Interactive Techniques*, 1995. 5

[68] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018. 1

[69] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017. 1

[70] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. SAMURAI: adapting segment anything model for zero-shot visual tracking with motion-aware memory. *CoRR*, abs/2411.11922, 2024. 3

[71] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *CoRR*, abs/1804.02505, 2018. 1

[72] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 6

[73] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. *CoRR*, abs/2109.09052, 2021. 1, 2

[74] Jiqing Zhang, Yuanchen Wang, Wenxi Liu, Meng Li, Jinpeng Bai, Baocai Yin, and Xin Yang. Frame-event alignment and fusion network for high frame rate tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9781–9790, 2023. 1, 2

[75] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *CoRR*, abs/2110.06864, 2021. 3

[76] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. 1

[77] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 7