TimeTracker: Event-based Continuous Point Tracking for Video Frame Interpolation with Non-linear Motion

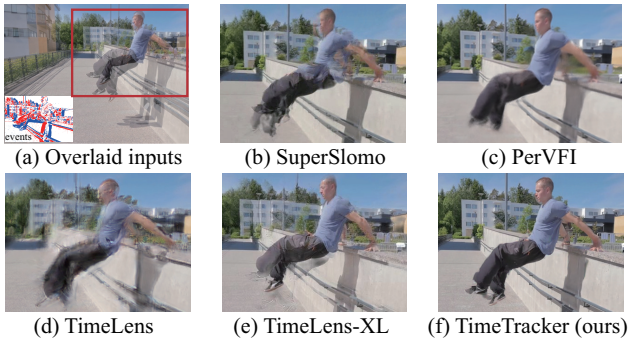
Haoyue Liu[1], Jinghan Xu[1], Yi Chang[1]*, Hanyu Zhou[1], Haozhi Zhao[1], Lin Wang[2], Lunxin Yan[1]
[1] National Key Lab of Multispectral Information Intelligent Processing Technology
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2] School of Electrical and Electronic Engineering, Nanyang Technological University
{liuhy, xujinghan, yichang}@hust.edu.cn

Abstract

*Video frame interpolation (VFI) that leverages the bio-inspired event cameras as guidance has recently shown better performance and memory efficiency than the frame-based methods, thanks to the event cameras' advantages, such as high temporal resolution. A hurdle for event-based VFI is how to effectively deal with non-linear motion, caused by the dynamic changes in motion direction and speed within the scene. Existing methods either use events to estimate sparse optical flow or fuse events with image features to estimate dense optical flow. Unfortunately, motion errors often degrade the VFI quality as the continuous motion cues from events do not align with the dense spatial information of images in the temporal dimension. In this paper, we find that object motion is continuous in space, tracking local regions over continuous time enables more accurate identification of spatiotemporal feature correlations. In light of this, we propose a novel continuous point tracking-based VFI framework, named **TimeTracker**. Specifically, we first design a Scene-Aware Region Segmentation (**SARS**) module to divide the scene into similar patches. Then, a Continuous Trajectory guided Motion Estimation (**CTME**) module is proposed to track the continuous motion trajectory of each patch through events. Finally, intermediate frames at any given time are generated through global motion optimization and frame refinement. Moreover, we collect a real-world dataset that features fast non-linear motion. Extensive experiments show that our method outperforms prior arts in both motion estimation and frame interpolation quality.*

1. Introduction

High-frame-rate imaging is invaluable in scientific research, industrial inspection, security surveillance, and other fields. However, high-speed cameras produce massive data volumes, necessitating high-performance computing equipment for processing and storage, which adds to operational complexity and maintenance. VFI offers a cost-effective alternative for high frame rate imaging by inferring intermediate frames from spatiotemporal cues in neighboring frames, enabling the temporal upsampling of video frames.

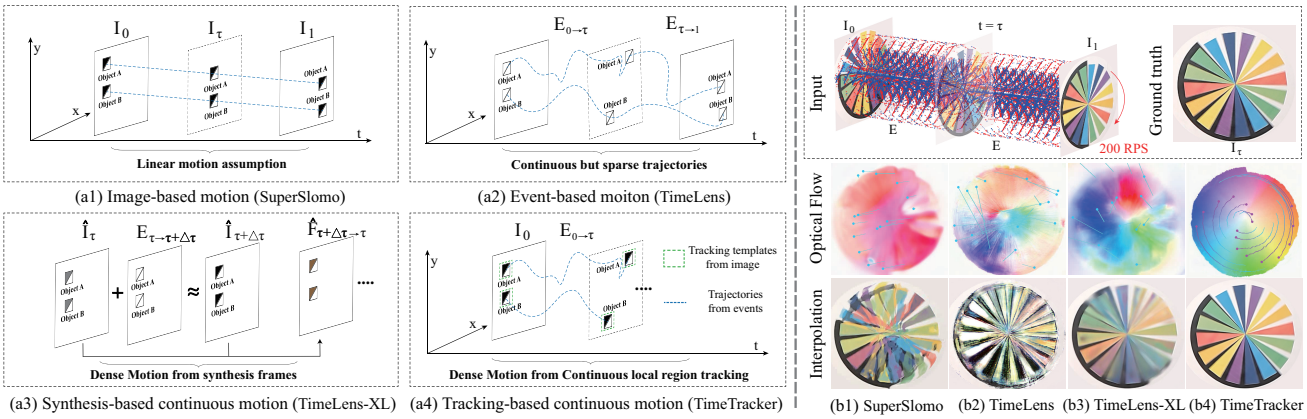


(a) Overlaid inputs (b) SuperSlomo (c) PerVFI
(d) TimeLens (e) TimeLens-XL (f) TimeTracker (ours)

Figure 1. Visual comparison of our method with other SOTA methods. (b) and (c) estimate optical flow from images, (d) estimates optical flow from events, and (e) fuses image and event information to estimate optical flow. Our method, based on continuous point tracking for optical flow estimation, achieves the best performance.

Optical flow estimation [1, 2] provides per-pixel displacement fields between frames, making it a widely used approach in VFI tasks [3–7]. However, the loss of inter-frame information makes it challenging to capture the motion of the scene accurately. Existing methods typically assume linear motion between frames, but this assumption is often inadequate for real-world scenes with complex motion. To improve motion estimation accuracy, [8–11] employs quadratic or cubic motion models, yet accurately capturing complex nonlinear motion between frames remains challenging. Inaccurate motion assumptions can lead to reconstruction artifacts, as shown in Fig. 2 (a1) and (b1).

Event cameras [14, 15] independently activate each pixel based on changes in brightness, capturing continuous motion edge information of objects. This characteristic offers several notable advantages, including high temporal reso-



(a1) Image-based motion (SuperSlomo) (a2) Event-based moiton (TimeLens)

(a3) Synthesis-based continuous motion (TimeLens-XL) (a4) Tracking-based continuous motion (TimeTracker)

(a) Illustration of four flow-based VFI paradigms

(b1) SuperSlomo (b2) TimeLens (b3) TimeLens-XL (b4) TimeTracker

(b) The optical flow estimation and frame interpolation results of different methods in challenging scenarios (200 revolutions per second turntable)

Figure 2. Illustration of (a) four flow-based VFI paradigms and (b) their comparison results. Image-based methods like SuperSlomo [3] rely on a linear motion assumption, which results in significant inaccuracies in nonlinear motion scenarios. Timelens [12] is a typical event-based VFI method that can only estimate sparse optical flow. TimelensXL [13] iteratively computes any-time optical flow by synthesizing intermediate frames, but errors in synthesized frames directly impact the accuracy of optical flow. We achieve dense any-time optical flow through image region segmentation and event-based point tracking, improving VFI performance in nonlinear scenarios.

lution and lower bandwidth, making them a cost-effective data source for VFI. Consequently, event-based VFI presents an attractive approach. Synthesis-based methods [16–18] directly fuse event and image features, with intermediate frames learned through a neural network. Flow-based methods [12, 13, 19–23] estimate inter-frame optical flow directly from events [12] or combine image-based and event-based optical flows [19, 23] to determine the inter-frame optical flow. Compared to synthesis-based methods, flow-based approaches offer stronger physical constraints and greater robustness.

While event cameras can capture high temporal resolution motion information, their asynchronous triggering nature results in a highly sparse spatial distribution. Although accumulating events over longer periods can alleviate this sparsity, it inevitably reduces the temporal resolution of the event slices. Accumulating events for extended durations in fast and nonlinear motion scenes can introduce greater errors. Additionally, when the brightness change falls below the event trigger threshold (as in low-texture areas), no events are generated, further exacerbating spatial sparsity and making it insufficient for dense prediction tasks like VFI, as shown in Fig. 2 (a2) and (b2). Moreover, estimating high temporal resolution optical flow from reconstructed images creates a chicken-and-egg problem, where errors in the reconstructed images can accumulate in the optical flow estimation, ultimately reducing the accuracy of the final interpolation results, as shown in Fig. 2 (a3) and (b3). The spatial sparsity of events poses a tricky challenge for event-based VFI: *how can we obtain an accurate dense any-time optical flow from spatially dense but temporally discrete frames and spatially sparse but temporally continuous events?*

To address the above issues, we propose a novel VFI framework based on point tracking, TimeTracker. The core insight of the proposed method is to transform the any-time optical flow estimation problem into a local feature tracking problem. Events, which naturally capture high time resolution motion trajectories of fast-moving objects, help mitigate correlation-matching errors that typically arise during optical flow computation due to the sparsity of event data. This approach leverages the complementary nature of event and image modalities in the temporal and spatial dimensions, improving interpolation accuracy for high-speed and complex trajectory motion scenarios.

Specifically, we first perform clustering and segmentation in the spatial domain based on the rich appearance features of the image. Locally similar regions in appearance tend to exhibit similar motion, especially for the rigid object. Next, we generate a motion region mask using the event trigger positions to distinguish dynamic regions from static ones. After sparsifying the image, we integrate the high temporal resolution motion trajectories provided by events for local feature tracking, resulting in a coarse dense any-time optical flow. Since feature tracking is only performed within a limited spatial area, it avoids the errors caused by global feature matching in conventional optical flow estimation. Subsequently, a global attention module is used to optimize the optical flow iteratively, and the interpolation results for any moment are computed based on the refined optical flow. Finally, we use a frame optimization module to repair the regions with local optical flow estimation errors. The results of TimeTracker are shown in Fig. 1, Fig. 2 (a4) and (b4).

In addition, we build a coaxial imaging system and collect a challenging real-world paired Dataset of images and events featuring Complex, High-speed Motion (CHMD), which serves as an evaluation benchmark. Compare to ex-

isting datasets BS-ERGB [19], ERF-X170FPS [23] and HQ-EVFI [13], CHMD features faster and more complex motion scenarios. It is carefully designed with controlled lighting conditions and optimized exposure times to minimize noise and motion blur. Overall, our main contributions can be summarized as follows:

- We propose the TimeTracker framework, which achieves any-time optical flow estimation through event-based point tracking. This method addresses the challenge of accurately estimating optical flow in high-speed and non-linear motion scenes for VFI tasks, while also enabling multi-frame interpolation.
- We propose an any-time dense optical flow estimation strategy that fully leverages the advantages of both modalities in the temporal and spatial dimensions. By utilizing the rich appearance information from images to sparsify the scene in the spatial domain, we transform the any-time optical flow estimation problem into a local feature tracking problem, effectively avoiding motion estimation errors caused by the spatial sparsity of events.
- We introduce a challenging real-world paired dataset featuring complex, high-speed motion as an evaluation benchmark. Extensive experiments demonstrate that the proposed method achieves state-of-the-art performance across multiple datasets with complex motion.

2. Related Work

Frame-based Video Frame Interpolation. Frame-based VFI has been widely studied and can generally be categorized into two main approaches: flow-based, and kernel-based methods. Flow-based methods [3–7] explicitly estimate the motion between frames to generate intermediate latent frames. Kernel-based methods [24–26] synthesize intermediate frames directly by applying convolution kernels within a network. Flow estimation [1, 2] is widely used in VFI because it provides clear physical meaning and motion description. Although existing methods have shown promising results, challenges remain in complex motion scenarios due to missing inter-frame information. To improve motion estimation accuracy, some studies [8–11] have introduced quadratic or cubic motion models , yet these approaches still struggle to model the intricate motion between frames.

Event-based Video Frame Interpolation. Event cameras [14, 15] provide high temporal resolution inter-frame visual information at a lower data rate, making event-based VFI a topic of growing interest in recent years [12, 13, 16–23, 27, 28]. Works like [16–18] fuse high temporal resolution events and images directly to generate intermediate frames, however, the sparse nature of events can lead to artifacts in synthesized results. Timelens [12] introduced a hybrid VFI framework that combines flow- and synthesis-based methods by estimating optical flow solely from events. Timelens++ [19], A^2OF [22] and CBM-Net [23] estimates optical flow from both event and image features, however, the significant differences between the two modalities may lead to errors in the feature correlation calculation. The approach most similar to ours is TimeLens-XL [13], which synthesizes intermediate frames using images and short-duration events, iteratively optimizing any-time optical flow and interpolation frames. However, its flow estimation depends on accurate frame synthesis, and errors in the synthesized frames can propagate as inaccurate optical flow over time.

In this work, we segment the image into similar local patches and track the motion trajectories of these patches, resulting in a non-linear and dense any-time optical flow.

Continuous Motion Estimation. In recent years, frame-based point-tracking methods [29–34] have made notable progress, allowing for the tracking of arbitrary points in images over extended time sequences. However, due to the low frame rate of images, these methods can only estimate optical flow over broader time scales, making them ineffective for capturing motion between frames. Event-based point-tracking methods [35–37] offer high temporal resolution but produce only sparse tracking trajectories, which can be insufficient for dense motion estimation tasks.

B-Flow [38] and MotionPriorCMax [39] attempt to densify event data by representing events as voxels, yet they struggle to produce accurate optical flow in sparse event regions. In contrast, Our approach fully leverages the rich appearance information from images to segment the scene into regions, initializes region-specific optical flow using point-tracking techniques, and refines this into dense, any-time optical flow using a global attention mechanism. This strategy effectively utilizes the spatial detail in images while avoiding the correlation errors that arise from attempting to estimate global motion from sparse event data alone.

3. Event-based Dense Any-time Flow for VFI

3.1. Framework Overview

The key to achieving VFI in nonlinear motion scenes is accurately estimating dense, any-time optical flow between frames. Nonlinear and large displacement motion is decomposed into local linear and small displacement motion at high temporal resolution scales. To achieve this, we utilize the event modality to capture fine-grained temporal details. To address the sparsity of events in the spatial domain, we explore an optical flow densification strategy that combines the strengths of both image and event modalities. Specifically, we segment the image into regions with similar appearance and employ events for template-based region tracking. As a result, the dense optical flow estimation problem is transformed into a local feature tracking problem. Thanks to the inherent temporal continuity of motion, this approach is more robust compared to directly fusing the bimodal fea-

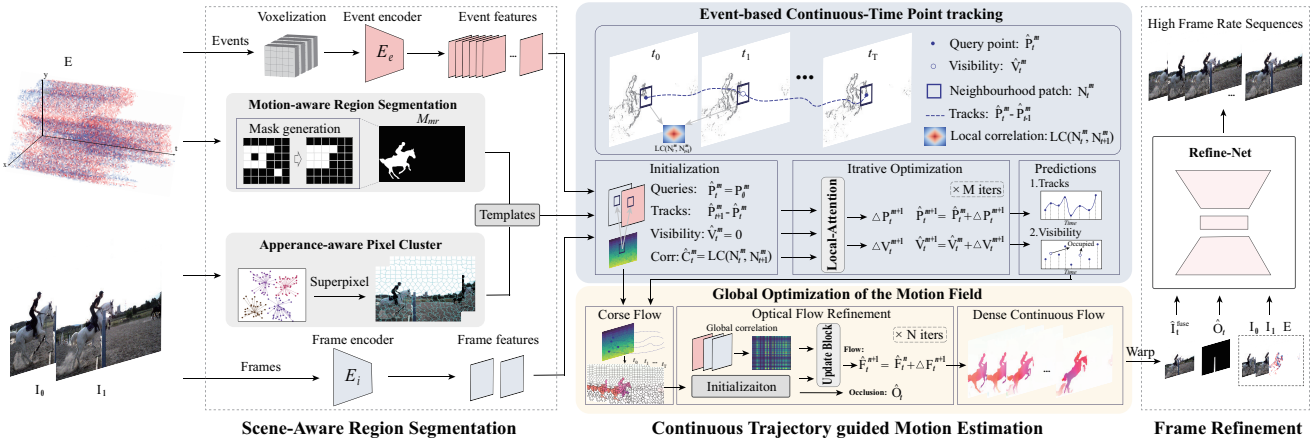


Figure 3. The overall architecture of the TimeTracker includes a Scene-Aware Region Segmentation (SARS) module, a Continuous Trajectory guided Motion Estimation (CTME) module, and a Frame Refinement (FR) module. The SARS segments the scene into multiple similar regions as tracking templates based on motion and appearance information. The CTME tracks the motion trajectories of each region and forms a dense any-time optical flow. Finally, the FR refines the warped images to obtain the interpolation results.

tures. Finally, a frame optimization module is used to refine regions with local optical flow estimation errors, further enhancing the interpolation accuracy. Fig. 3 illustrates the overall framework of TimeTracker.

3.2. Scene-Aware Region Segmentation

Motion-aware Region Segmentation. The event generation process [15] can be formulated as

$$\Delta L = logI(x, y, t) - logI(x, y, t - \Delta t) = pC, \quad (1)$$

where $logI(x, y, t)$ is the logarithmic illumination at pixel (x, y) and time t, Δt is the time interval between consecutive events, $p \in [-1, 1]$ is the polarity of events, and C is the contrast threshold of the event camera. This process indicates that an event is triggered once the logarithmic illumination change at a particular pixel exceeds the threshold C. Assuming constant illumination [15], for small Δt, equation (1) can be formulated as

$$\Delta L \approx -\nabla L \cdot v \Delta t, \quad (2)$$

where ∇L is the brightness gradient, v is the optical flow. It can be observed that events are generated by the edges of object motion, while stationary regions with constant brightness generate no events. However, the differential imaging process is prone to generating isolated noise. To initialize a better tracking template, it is necessary to filter out regions with no events and those containing isolated noise from the region segmentation results. First, we represent all events $E = \{e_i\}_{i=0}^{N-1}$ between boundary images $\{I_0, I_1\}$ as an event frame, ignoring the polarity of the events. Then, we apply morphological closing to connect broken regions in the event frame and remove isolated noise points, resulting in the motion region mask M_{mr}.

Appearance-aware Pixel Cluster. Directly inferring a dense optical flow from sparse events is an ill-posed problem. We aim to make full use of the rich appearance information in images by segmenting the scene into several small regions and then tracking motion trajectories within each region using the corresponding events. This approach enables us to initialize a coarse dense optical flow.

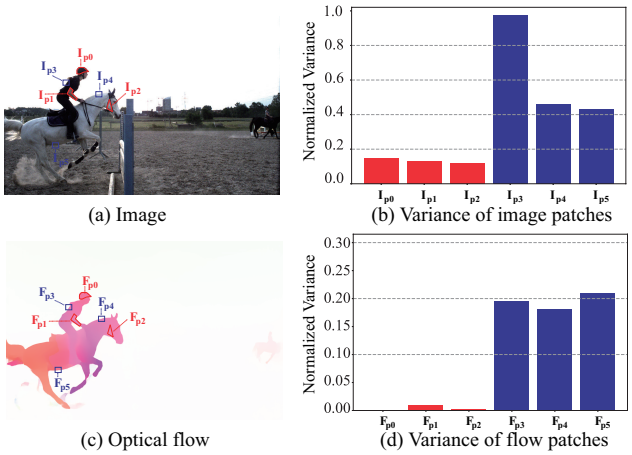
Rigid motion often exhibits spatial consistency, leading some methods [40–42] to leverage semantic information to enhance optical flow estimation by using distinctive object edges to maintain motion boundary accuracy. However, this assumption may break down in cases of dynamic textures, such as a waving arm, and directly using semantic segmentation models can incur high computational costs. Therefore, we adopt a simple yet effective method, SLIC [43], to cluster pixels based on their intensity values and spatial positions, creating smaller segmented regions where motion consistency is easier to maintain at a finer spatial scale.

Fig. 4 illustrates this concept: in the red box in Fig. 4 (a), the moving foreground shares similar pixel values, while the blue box includes both foreground and background elements with more varied pixel values. Fig. 4 (b) displays normalized pixel variance within each region, and Fig. 4 (c) and Fig. 4 (d) show the corresponding optical flow and normalized variance. It is evident that small regions with closely similar pixel values and spatial proximity exhibit consistent optical flow, and vice versa. Finally, we use the motion mask M to filter out invalid regions. Details and visual results of the SLIC can be found in the supplementary materials.

3.3. Continuous Trajectory Motion Estimation

Event Representation. The event stream needs be converted into tensor form for network input. We use voxels [44] to represent the events. The voxelization method transform events $E_k = \{e_i\}_{i=0}^{N-1}$ into a tensor $V \in \mathbb{R}^{B \times H \times W}$ with B bins, which can be formulated as

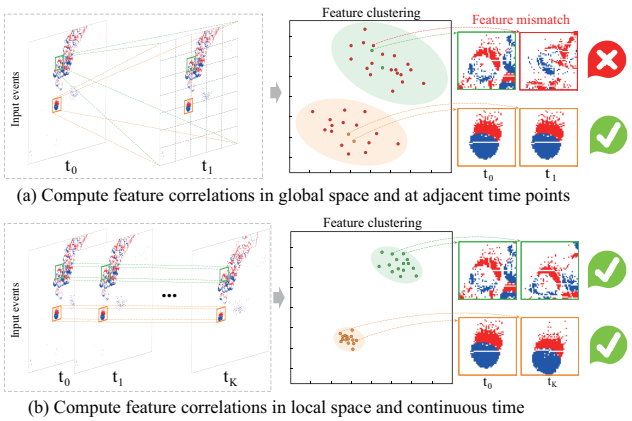
$$V(k) = \sum_i p_i \max(0, 1 - \left| k - \frac{t_i - t_0}{t_N - t_0}(B-1) \right|), \quad (3)$$



(a) Image (b) Variance of image patches

(c) Optical flow (d) Variance of flow patches

Figure 4. The correlation between appearance and motion. We select three regions with similar appearances, p0∼p2 , in the image (a) and optical flow (b), along with three regions containing both foreground and background, p3∼p5. (c) and (d) illustrate that small regions with minimal pixel value differences and spatial proximity in the image also exhibit consistent optical flow, and vice versa.



(a) Compute feature correlations in global space and at adjacent time points

(b) Compute feature correlations in local space and continuous time

Figure 5. Two continuous motion estimation paradigms. (a) Traditional motion estimation methods calculate the cost volume between temporally adjacent features. Due to the low spatial distinguishability and sparsity of events, mismatches are likely to occur. (b) Since events are temporally continuous, performing continuous tracking within a local region is more robust.

where N is the number of events, p_i and t_i represent the polarity and timestamp of the i-th event respectively, and the range of k is in $[0, B-1]$.

Event-based Continuous-Time Point Tracking. Compared to popular event-based optical flow methods like E-RAFT [45], which calculate the global cost volume between features at adjacent time points to obtain correlations, we focus on tracking point trajectories over continuous time within a local region, and learn the displacement of the point over continuous time. Our motivation stems from the inherent property of events being triggered at motion edges, enabling them to capture continuous trajectories with high temporal resolution. However, events exhibit high spatial sparsity and distinguishability, which can lead to erroneous matches when computing the global cost volume of event features, as illustrated in Fig. 5, the similarity of event features is higher in local space and continuous time (feature clustering is denser). Therefore, we select points from the regions filtered in Sec. 3.2 for inter-frame tracking, which represent the motion trajectory of the region over continuous time.

Specifically, we begin by using SIFT [46] to identify easily trackable points on the boundary frames $\{I_0, I_1\}$. These points are then used to initialize a query point for each superpixel segment. If no event occurs at that location, we select the nearest event trigger location within the region as the query point. This process results in a set of query points.

Next, we use two separate feature extraction networks to extract multi-scale features from events and images, respectively. These features are generated across four downsampling scales: $s \in \{4, 8, 16, 32\}$, with down-sampled features obtained via average pooling. Centered on each sample point, a square neighborhood N_t is sampled at each scale using bilinear interpolation. In this region, the 4D correlation [2, 34] $C_t = LC(N_t, N_{t+1})$ between features at adjacent time points is calculated to track the position of the query point over time. $LC(\cdot)$ is obtained by stacking the inner products of features across multiple scales.

Similar to [33, 34], we use a transformer [47] block to establish attention over local space in continuous time, enabling motion trajectory tracking and optimization. First, for i-th query point at each time step t, we define the input tokens as follows:

$$G_t^i = \left(P_{t+1}^i - \hat{P}_t^i, N_t^i, \hat{V}_t^i, C_t^i, \eta(\hat{P}_t^i - \hat{P}_1^i)\right), \quad (4)$$

where $\hat{P}_{t+1}^i - \hat{P}_t^i$ is the predicted pixel displacement, N_t^i is the content feature patch around query point, $\hat{V}_t^i$ is the predicted visibility, C_t^i is the correlation volume, and $\eta(\cdot)$ denotes a sinusoidal positional encoding function. Then, we apply a sliding window of length L between the boundary frames, moving forward by $L/2$ at each step. This can simultaneously model both the temporal correlation of a point trajectory over continuous time and the spatial correlation between different points. Each window undergoes M optimization steps to yield the final tracks and visibility.

Global Optimization of the Optical Flow. Through point tracking, we can obtain the trajectory of each superpixel region over continuous time, resulting in a coarse, dense any-time optical flow $\hat{F}_{0\to1}^{coa}$ and a corresponding visibility mask $\hat{O}_{0\to1}$. Since potential errors may exist in point tracking, we further perform global optimization on the optical flow results. Our approach is based on the idea that optical flow and frame interpolation tasks should mutually reinforce each other, with interpolation results providing backward constraints to improve the accuracy of optical flow estimation. Specifically, at high frame rates, large displacements and nonlinear motion can be approximated as locally linear

motion. If the interpolation results are accurate, existing optical flow estimation methods [2, 48] should be able to produce relatively accurate optical flow. After excluding occluded regions, this optical flow should remain consistent with $\hat{F}^{coa}_{0\to1}$. Consequently, we construct a self-supervised consistency loss between them during training and update the parameters using a global optical flow optimization module, such as RAFT [2], and iteratively optimize N times during training. The global optimization module can simultaneously optimize both the optical flow and the occlusion. The loss function is defined as

$$\mathcal{L}_{flow} = \left\| (\hat{F}^{coa}_{t-1\to t} - v(\hat{I}_{t-1}, \hat{I}_t)) \odot \hat{O}_t \right\|_1, \quad (5)$$

where $v(\cdot)$ denotes a frame-based optical flow estimation network, $\hat{I}_{t-1}$ and $\hat{I}_t$ are the VFI results, $\hat{O}_t$ is occlusion mask, and $\odot$ denotes element-wise multiplication. After global optimization, we obtain the refined dense any-time optical flow $\hat{F}^{refine}_{0\to1}$. In addition, we reverse the event stream following the method in [12] and compute the backward optical flow $\hat{F}^{refine}_{1\to0}$ in the same way, thereby obtaining bidirectional any-time optical flow.

3.4. Frame Interpolation and Refinement

After obtaining the bidirectional optical flow, we warp the boundary images $\{I_0, I_1\}$ and fuse them using the method described in [49, 50] to generate the intermediate frame:

$$\hat{I}^{fuse}_t = \frac{C_{t,0}}{C_{t,0}+C_{t,1}} w_b(I_0, \hat{F}^{refine}_{t\to0}) + \frac{C_{t,1}}{C_{t,0}+C_{t,1}} w_b(I_1, \hat{F}^{refine}_{t\to1}), \quad (6)$$

where $w_b(\cdot)$ denotes backward warping, $\hat{F}^{refine}_{t\to0}$ and $\hat{F}^{refine}_{t\to1}$ are the intermediate flows sampled from the bidirectional any-time optical flow based on t, and $C_{0,1}$ is the confidence map, defined as follows:

$$C_{0,1} = exp\left(-\frac{\left|\hat{F}_{0\to1}(x) + \hat{F}_{1\to0}(x + \hat{F}_{0\to1}(x))\right|^2}{\gamma_1(\left|\hat{F}_{0\to1}\right|^2 + \left|\hat{F}_{1\to0}(x+\hat{F}_{0\to1})\right|^2) + \gamma_2} \right), \quad (7)$$

where $\gamma_1 = 0.01$ and $\gamma_2 = 0.5$ from [49]. Due to the challenge of accurately estimating optical flow in occluded regions, we generate this part using a synthetic approach, which has been shown to be effective in previous work [12, 18, 19, 23]. We first compute the occlusion regions $\hat{O}_t = \hat{O}^{forward}_t \cap \hat{O}^{backward}_t$, then input the fused image $\hat{I}^{fuse}_t$, boundary images $\{I_0, I_1\}$, events E, and occlusion mask $\hat{O}_t$ into the U-shape refine network. This network separately encodes the occlusion mask to provide attention for the refinement process. It then extracts information from events and boundary images to correct the erroneous regions in the fused image. Note that, after obtaining the any-time optical flow, we do not need to recursively interpolate frames at all time steps; instead, we perform selective interpolation based on the specified timestamp.

3.5. Training Details

Loss Function. The model loss function includes tracking loss, occlusion loss, reconstruction loss, and optical flow loss. The tracking loss and the occlusion loss are defined as:

$$\mathcal{L}_{track} = \sum_{m=1}^{M} \left\| \hat{P}^m_t - P^{GT}_t \right\|_1, \quad (8)$$

$$\mathcal{L}_{occ} = \sum_{m=1}^{M} BCE(\hat{V}^m_t, V^{GT}_t), \quad (9)$$

where $BCE(\cdot)$ denotes binary cross entropy loss, m is the number of iterations, $\hat{P}^m_t$ and $\hat{V}^m_t$ are the predicted point positions and occlusions, respectively.

The reconstruction loss is defined as:

$$\mathcal{L}_{rec} = \left\| \hat{I}_t - I^{GT}_t \right\|_1, \quad (10)$$

where $\hat{I}_t$ is the predicted frame. The total loss during the training phase of the tracking model is:

$$\mathcal{L}_{total\_track} = \mathcal{L}_{track} + \lambda_1 \mathcal{L}_{occ}, \quad (11)$$

the total loss during the training of the VFI model is:

$$\mathcal{L}_{total\_rec} = \lambda_2 \mathcal{L}_{rec} + \lambda_3 \mathcal{L}_{flow}, \quad (12)$$

we set $\lambda_1 = 1$, $\lambda_2 = 1$, and $\lambda_3 = 0.8$ respectively.

Implementation. We train the model in two steps. In the first step, we convert the Tap-Vid dataset [29] to events using ESIM [51] and train the point tracking model on this dataset for 200K iterations. We then fine-tune on the Multiflow dataset [38] for 15K iterations, using the ADAM [52] optimizer with a learning rate of 0.0005, the sliding window length is set to $L = 10$, the iteration number is set to $M = 5$. In the second step, we convert the GoPro dataset to events using ESIM [51], freeze the tracking model, and train for 200K iterations with the ADAM [52] optimizer, starting with a learning rate of 10^{-4} and applying cosine decay down to 10^{-6}. Training samples are cropped to 256×256, the iteration number is set to $N = 10$. All training is conducted using the PyTorch [53] on an NVIDIA A100 GPU.

4. Experiments

4.1. Datasets and Experimental Settings

Datasets. Following the setup in previous works [12, 13, 18, 23], we evaluate the model on both synthetic and real event datasets using the same approach. For synthetic evaluation, we select GoPro [54], and SNU-FLIM [55], generating events with ESIM [51]. For real-world data evaluation, we chose the BS-ERGB [19] and CHRD datasets, with CHRD including more challenging scenarios involving fast, nonlinear motion. Further details on CHRD are provided in the supplementary materials.

Comparison Methods. We compare our model with three frame-based SOTA methods: SuperSlomo [3], PerVFI [56], and VFIT-B [57], and four event-based SOTA methods: Timelens [12], CBMNet [23], SuperFast [17], and TimelensXL [13]. Among these, SuperSlomo [3] and PerVFI [56] are frame optical flow based methods, Timelens [12] is an event optical flow based method, CBMNet [23] and TimelensXL [13] are event and image fusion optical flow methods, VFIT-B [57] is an frame synthesis-based method,

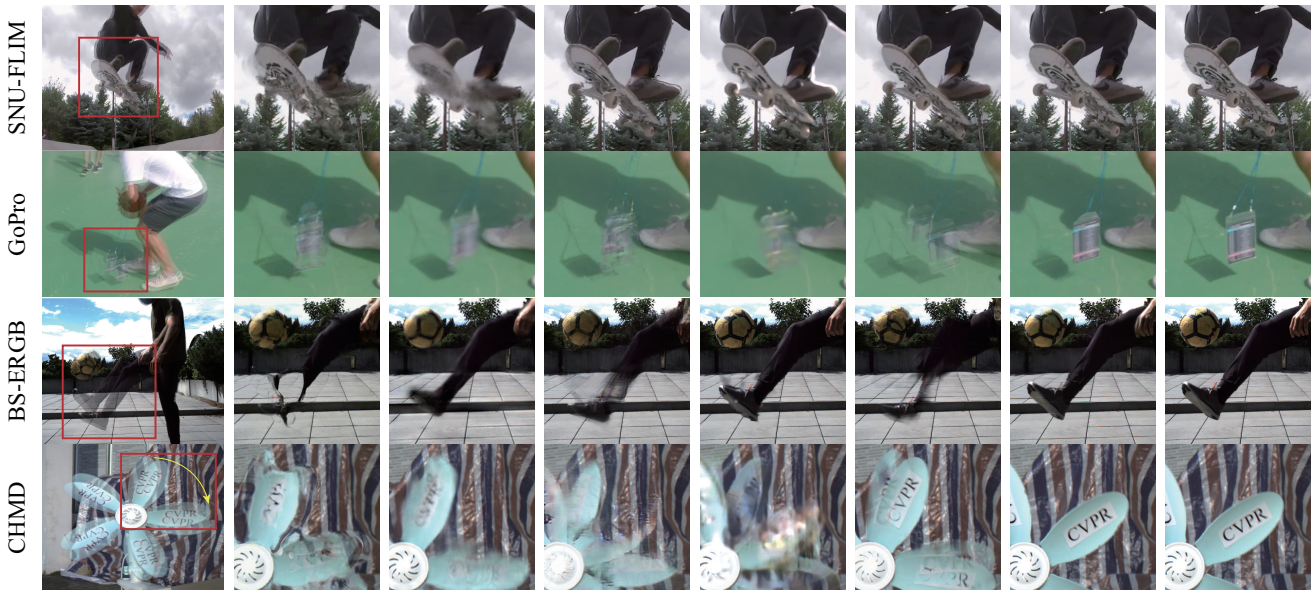


Figure 6. Visual comparison of the proposed method and other SOTA methods across different datasets.

dataset	Gopro				SNU-FILM			
	7skips		15skips		Hard		Extreme	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SuperSlomo[3]	28.28	0.902	23.31	0.776	24.71	0.846	21.73	0.794
VFIT-B[57]	30.80	0.912	26.10	0.836	26.34	0.883	25.49	0.852
PerFVI[56]	31.86	0.933	27.46	0.845	29.77	0.913	27.84	0.891
Timelens[12]	34.42	0.948	33.31	0.928	31.45	0.928	28.73	0.897
SuperFast[17]	33.76	0.943	32.97	0.927	28.74	0.903	26.37	0.863
CBMNet[23]	36.86	0.955	35.32	0.947	30.87	0.918	27.56	0.884
Timelens-XL[13]	37.02	0.959	36.19	0.949	30.95	0.920	27.93	0.894
TimeTracker (ours)	**37.13**	**0.962**	**36.54**	**0.958**	**32.86**	**0.935**	**29.27**	**0.915**

Table 1. Quantitative results on synthetic datasets.

dataset	BS-ERGB				Ours			
	1skip		3skips		7skips		15skips	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SuperSlomo[3]	23.33	0.734	22.43	0.716	21.17	0.705	20.26	0.674
VFIT-B[57]	24.44	0.741	24.31	0.725	22.04	0.743	21.70	0.682
PerFVI[56]	27.72	0.761	26.07	0.763	24.82	0.768	21.65	0.702
Timelens[12]	28.13	0.787	26.82	0.769	25.86	0.771	24.12	0.748
SuperFast[17]	27.87	0.768	26.77	0.758	22.79	0.762	20.59	0.722
CBMNet[23]	29.03	0.807	28.10	0.794	26.27	0.792	25.38	0.766
Timelens-XL[13]	29.35	0.813	28.69	0.802	26.13	0.785	24.77	0.732
TimeTracker (ours)	**29.85**	**0.823**	**29.14**	**0.807**	**28.45**	**0.814**	**27.69**	**0.805**

Q

Table 2. Quantitative results on real-world datasets.

and SuperFast [17] is a synthesis-based method that directly fuses images and events. Additionally, since Timelens [12] and SuperFast [17] were trained on different datasets, we fine-tune them on the GoPro dataset [54] after loading their pretrained weights to ensure a fair comparison.

4.2. Comparison Experiments

Comparison on Synthetic Datasets. The quantitative and qualitative results on the synthetic datasets are reported in Tab. 1 and the first two rows of Fig. 6. The proposed method outperforms existing methods in both PSNR and SSIM metrics. Frame-based methods, limited by the loss of inter-frame information, exhibit noticeable artifacts in areas with complex motion, and the reconstruction quality degrades significantly as more frames are skipped. Event-based methods, on the other hand, tend to show texture loss in dense-texture areas. In comparison, our method demonstrates a clear advantage in both visual quality and quantitative metrics. Benefiting from continuous trajectory tracking, TimeTracker achieves more stable results when skipping more frames.

Comparison on Real Datasets. Tab. 2 and the last two rows of Fig. 6 present the quantitative and qualitative results on real-world datasets. Event-based methods demonstrate better PSNR and SSIM metrics. On the CHRD dataset, which includes fast, nonlinear motion, frame-based method, frame-based methods incorrectly estimate the position of the fan blades, while other event-based methods exhibit severe artifacts. Our method demonstrates a significant advantage due to its accurate dense any-time optical flow.

4.3. Ablation Study and Discussion

What role does each component of TimeTracker play? We study the roles of the main components in TimeTracker by removing them individually. Fig. 7 visually demonstrates the results of motion estimation and frame interpolation. We first replace the superpixel module with uniform image segmentation. The template obtained through uniform segmentation has pixels that do not exhibit motion consistency, and the incorrect initialization leads to poor motion global

w/o superpixel **w/o** point tracking **w/o** global optimization **w/o** frame refinement

Figure 7. Ablation studies of TimeTracker. The first row and the second row show the optical flow and VFI results, respectively.

Superpixel	Point tracking	Global optimization	Frame refinement	PSNR ↑	SSIM ↑
	✓	✓	✓	30.62	0.925
✓		✓	✓	27.15	0.885
✓	✓		✓	32.43	0.933
✓	✓	✓		35.29	0.942
✓	✓	✓	✓	37.47	0.965

Table 3. Ablation studies on main components of TimeTracker.

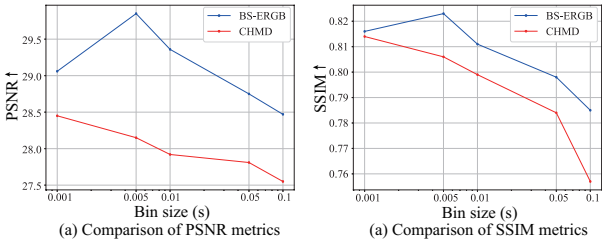
optimization results. Removing the point tracking module results in zero-initialized optical flow, which fails to estimate the flow correctly. Removing the global optimization module causes artifacts from residual region segmentation in the optical flow. We further perform a quantitative analysis using the GoPro [54] dataset, as shown in Tab. 3. The superpixel and point tracking modules are critical, while global optical flow optimization further improves reconstruction quality, with image refinement having a relatively weak effect.

Comparison of Motion Estimation Strategies. We further compare the effectiveness of the motion estimation modules in different VFI methods [12, 13, 23, 56], as shown in Tab. 4. Specifically, we use the optical flow from the above methods to perform backward warping on the boundary frames and calculate the PSNR and SSIM metrics of the warped images. The tests are conducted on the GoPro dataset. PerVFI [56] exhibits optical flow errors due to its inaccurate motion prior assumptions. Event-based methods like TimeLens [12] and CBMNet [23] face limitations in motion estimation accuracy due to the inherent sparsity of events. While TimeLens-XL [13] attempts to address this by estimating optical flow through synthesized frames, which introduces additional instability. In contrast, TimeTracker achieves better results by utilizing point tracking-guided motion estimation.

Temporal Resolution of Voxel Grid. Events need to be voxelized before they can be converted into tensors that can be processed by the network. The shorter the time interval for generating the voxels, the smaller the displacement of objects within a unit voxel, leading to higher correlation between voxels at adjacent time steps, and vice versa. However, shorter time intervals also increase the computational load for point tracking within a fixed time period, and excessively short intervals may result in incomplete event features within a single voxel bin. In Fig 8, we analyze the impact of different voxel bin sizes on VFI results in the BS-ERGB [19] and CHMD, where the objects in CHMD move relatively faster.

Methods	PerVFI	TimeLens	CBMNet	TimeLens-XL	TimeTraker
Data source	Image (I)	Event (E)	I+E	I+E	I+E
PSNR ↑	30.25	30.84	32.19	33.46	**35.29**
SSIM ↑	0.908	0.916	0.925	0.939	**0.942**

Table 4. Comparison of optical flow estimation performance.



(a) Comparison of PSNR metrics (a) Comparison of SSIM metrics

Figure 8. The impact of bin size settings on reconstruction quality.

It can be seen that in the BS-ERGB, the reconstruction quality is optimal when the voxel bin size is set to 0.005s, but in the CHMD, the reconstruction quality decreases as the voxel bin size increases. The reason is that objects with fast motion in CHMD have a higher event density, and high temporal resolution voxels help improve tracking performance.

Limitation and Future Work. Tracking-based motion estimation methods face limitations in dynamic texture scenes, such as fluids. The main challenge is that fluid features are temporally discontinuous (e.g., the shape of splashing water changes continuously, potentially appearing and disappearing rapidly). As a result, TimeTracker adopts an approach similar to existing methods [12, 19, 23], directly synthesizing the relevant regions, which may lead to suboptimal results. However, event cameras inherently possess the ability to measure fluid motion. For instance, EBOS [58] uses background-oriented schlieren method to measure gas flow fields, and EBIV [59] employs events to measure fluid particle velocities. In future work, we will further explore high frame rate imaging technology in dynamic textures scenes.

5. Conclusion

In this work, we propose TimeTracker, a novel point-tracking-based VFI framework that effectively adapts to complex nonlinear motion scenarios. To the best of our knowledge, this is the first study to address the VFI problem through continuous point tracking. We segment the scene into locally similar regions using the rich appearance features from the image, then track the continuous trajectories of these local regions using events, resulting in dense and any-time optical flow. Intermediate frames at any given time are generated through global motion optimization and frame refinement. Additionally, we introduce a dataset featuring fast nonlinear motion as a evaluation benchmark. The proposed method significantly outperforms state-of-the-art approaches, and we believe that our work can bring new perspectives to the community.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant U24B20139 and 62371203, the Hubei Province Science Foundation of Distinguished Young Scholars under Grant JCZRJQ202500097, and the Start Up Grant at Nanyang Technological University under Grant 03INS002165C140. The computation is completed in the HPC Platform of Huazhong University of Science and Technology.

References

[1] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8934–8943, 2018. 1, 3

[2] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, pages 402–419, 2020. 1, 3, 5, 6

[3] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9000–9008, 2018. 1, 2, 3, 6, 7

[4] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3703–3712, 2019.

[5] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5437–5446, 2020.

[6] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Eur. Conf. Comput. Vis.*, pages 109–125, 2020.

[7] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Eur. Conf. Comput. Vis.*, pages 624–642, 2022. 1, 3

[8] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 1, 3

[9] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *Eur. Conf. Comput. Vis.*, pages 107–123, 2020.

[10] Youjian Zhang, Chaoyue Wang, and Dacheng Tao. Video frame interpolation without temporal priors. *Adv. Neural Inform. Process. Syst.*, pages 13308–13318, 2020.

[11] Mengshun Hu, Kui Jiang, Zhihang Zhong, Zheng Wang, and Yinqiang Zheng. Iq-vfi: Implicit quadratic motion estimation for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6410–6419, 2024. 1, 3

[12] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16155–16164, 2021. 2, 3, 6, 7, 8

[13] Yongrui Ma, Shi Guo, Yutian Chen, Tianfan Xue, and Jinwei Gu. Timelens-xl: Real-time event-based video frame interpolation with large motion. In *Eur. Conf. Comput. Vis.*, pages 178–194, 2025. 2, 3, 6, 7, 8

[14] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 $\times$128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits*, 43(2):566–576, 2008. 1, 3

[15] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1):154–180, 2020. 1, 3, 4

[16] Zeyu Xiao, Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Eva2: Event-assisted video frame interpolation via cross-modal alignment and aggregation. *IEEE Trans. Comput. Imaging*, 8:1145–1158, 2022. 2, 3

[17] Yue Gao, Siqi Li, Yipeng Li, Yandong Guo, and Qionghai Dai. Superfast: 200$\times$ video frame interpolation via event camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7764–7780, 2022. 6, 7

[18] Yuhan Liu, Yongjian Deng, Hao Chen, and Zhen Yang. Video frame interpolation via direct synthesis with the event-based reference. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8477–8487, 2024. 2, 3, 6

[19] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17755–17764, 2022. 2, 3, 6, 8

[20] Zhiyang Yu, Yu Zhang, Deyuan Liu, Dongqing Zou, Xijun Chen, Yebin Liu, and Jimmy S Ren. Training weakly supervised video frame interpolation with events. In *Int. Conf. Comput. Vis.*, pages 14589–14598, 2021.

[21] Weihua He, Kaichao You, Zhendong Qiao, Xu Jia, Ziyang Zhang, Wenhui Wang, Huchuan Lu, Yaoyuan Wang, and Jianxing Liao. Timereplayer: Unlocking the potential of event cameras for video interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17804–17813, 2022.

[22] Song Wu, Kaichao You, Weihua He, Chen Yang, Yang Tian, Yaoyuan Wang, Ziyang Zhang, and Jianxing Liao. Video interpolation by event-driven anisotropic adjustment of optical flow. In *Eur. Conf. Comput. Vis.*, pages 267–283, 2022.

[23] Taewoo Kim, Yujeong Chae, Hyun-Kurl Jang, and Kuk-Jin Yoon. Event-based video frame interpolation with cross-modal asymmetric bidirectional motion fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 18032–18042, 2023. 2, 3, 6, 7, 8

[24] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 670–679, 2017. 3

[25] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE international conference on computer vision*, pages 261–270, 2017.

[26] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive

collaboration of flows for video frame interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5316–5325, 2020. 3

[27] Hoonhee Cho, Taewoo Kim, Yuhwan Jeong, and Kuk-Jin Yoon. Tta-evf: Test-time adaptation for event-based video frame interpolation via reliable pixel and sample estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 25701–25711, 2024. 3

[28] Yunfan Lu, Guoqiang Liang, Yusheng Wang, Lin Wang, and Hui Xiong. Uniinr: Event-guided unified rolling shutter correction, deblurring, and interpolation. In *European Conference on Computer Vision*, pages 1–20. Springer, 2025. 3

[29] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Adv. Neural Inform. Process. Syst.*, pages 13610–13626, 2022. 3, 6

[30] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Int. Conf. Comput. Vis.*, pages 10061–10072, 2023.

[31] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *Int. Conf. Comput. Vis.*, pages 19795–19806, 2023.

[32] 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 *Int. Conf. Comput. Vis.*, pages 19855–19865, 2023.

[33] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. *arXiv preprint arXiv:2407.15420*, 2024. 5

[34] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 3, 5

[35] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Eklt: Asynchronous photometric feature tracking using events and frames. *Int. J. Comput. Vis.*, 128(3):601–618, 2020. 3

[36] Nico Messikommer, Carter Fang, Mathias Gehrig, and Davide Scaramuzza. Data-driven feature tracking for event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5642–5651, 2023.

[37] Jiaxiong Liu, Bo Wang, Zhen Tan, Jinpu Zhang, Hui Shen, and Dewen Hu. Tracking any point with frame-event fusion network at high frame rate. *arXiv preprint arXiv:2409.11953*, 2024. 3

[38] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024. 3, 6

[39] Friedhelm Hamann, Ziyun Wang, Ioannis Asmanis, Kenneth Chaney, Guillermo Gallego, and Kostas Daniilidis. Motion-prior contrast maximization for dense continuous-time motion estimation. *arXiv preprint arXiv:2407.10802*, 2024. 3

[40] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J Black. Optical flow with semantic segmentation and localized layers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3889–3898, 2016. 4

[41] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *AAAI Conf. Artif. Intell.*, pages 10713–10720, 2020.

[42] Shuai Yuan, Shuzhi Yu, Hannah Kim, and Carlo Tomasi. Semarflow: Injecting semantics into unsupervised optical flow estimation for autonomous driving. In *Int. Conf. Comput. Vis.*, pages 9566–9577, 2023. 4

[43] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2274–2282, 2012. 4

[44] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 989–997, 2019. 4

[45] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *International Conference on 3D Vision*, pages 197–206, 2021. 5

[46] David G Lowe. Object recognition from local scale-invariant features. In *Int. Conf. Comput. Vis.*, pages 1150–1157, 1999. 5

[47] A Vaswani. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 2017. 5

[48] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *Eur. Conf. Comput. Vis.*, pages 668–685, 2022. 6

[49] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI Conf. Artif. Intell.*, 2018. 6

[50] Jisoo Jeong, Hong Cai, Risheek Garrepalli, Jamie Menjay Lin, Munawar Hayat, and Fatih Porikli. Ocai: Improving optical flow estimation by occlusion and consistency aware interpolation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 19352–19362, 2024. 6

[51] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3586–3595, 2020. 6

[52] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[53] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 6

[54] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3883–3891, 2017. 6, 7, 8

[55] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10663–10671, 2020. 6

[56] Guangyang Wu, Xin Tao, Changlin Li, Wenyi Wang, Xiaohong Liu, and Qingqing Zheng. Perception-oriented video frame interpolation via asymmetric blending. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2753–2762, 2024. 6, 7, 8

[57] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17482–17491, 2022. 6, 7

[58] Shintaro Shiba, Friedhelm Hamann, Yoshimitsu Aoki, and Guillermo Gallego. Event-based background-oriented schlieren. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023. 8

[59] Christian E Willert and Joachim Klinner. Event-based imaging velocimetry: an assessment of event-based cameras for the measurement of fluid flows. *Exp. Fluids*, 63(6):101, 2022. 8