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

Supplementary Material

Summary

The supplementary material is organized as follows.

- Section 1 introduces the implementation details of the proposed method and dataset.
- Section 2 discusses more ablation studies of TimeTracker.
- Section 3 shows more visual results on different datasets.

1. Implementation Details

1.1. Appearance-aware Pixel Cluster

We first present the algorithmic process of Simple Linear Iterative Clustering (SLIC) [1], as shown in Algorithm 1. SLIC segments the input image I into N appearance-similar regions based on the preset number of cluster centers N. To adapt to different image sizes, we set the number of cluster centers to $N = \frac{H}{30} \times \frac{W}{30}$, meaning each segmented region is initialized to 30×30 pixels. In Sec. 2.1, we conduct an ablation study on the number of cluster centers.

Algorithm 1 SLIC

Input: Image I and number of clustering centers N
Output: Region index $R_i \in \{1, N\}$ of each pixel i

Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S
Move cluster centers to the lowest gradient position in a 3×3 neighborhood
Set label $l(i) = -1$ for each pixel i
Set distance $d(i) = \infty$ for each pixel i
while $E > threshold$ **do**
  for each cluster centers C_k **do**
    for each pixel i in a $2S \times 2S$ region around C_k **do**
      if $D < d(i)$ **then**
        set $d(i) = D$
        set $l(i) = k$
      end if
    end for
  end for
  update cluster centers C_k
  Compute residual error E
end while

1.2. Method details

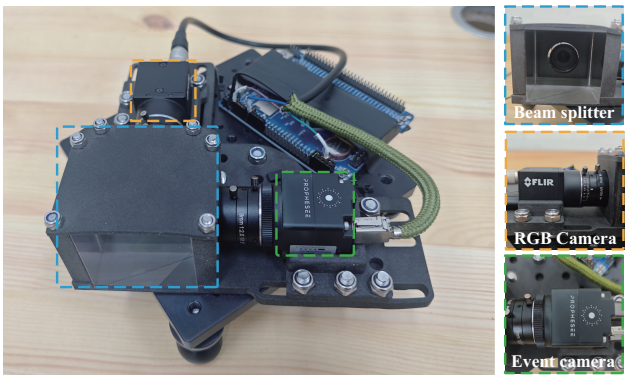
Events chunking method. The bin size of event voxel is set to 5ms during training. During testing, bin sizes are adjusted based on pixel displacement speed. Motion trajectories are obtained by linearly connecting adjacent bins. For high-speed motion, accuracy improves by reducing the bin size.
Motion-aware Region Segmentation. The motion mask is generated using a 7×7 kernel closing operation, and super-pixel regions intersecting with it are selected as templates. This design ensures the model prioritizes interpolation in motion areas of the scene.
Refine-Net. Refine-Net is a synthesis-based branch with a 2D U-shape structure consisting of a three-layer encoder-decoder. Superpixel-based point tracking effectively reduces the complexity of optical flow while better preserving the spatial details required for the VFI task. However, in regions with dynamic textures such as fluids, ensuring accurate motion estimation is particularly challenging. To address this, the frame refinement module in our TimeTracker framework is specifically designed to correct areas with optical flow estimation errors, enhancing the overall interpolation quality.

1.3. Datasets Details

It is essential to train and test event-based frame interpolation methods on real-world datasets. Several public datasets have been proposed, including HS-ERGB [18], BS-ERGB [19], ERF-X170FPS [12], and HQ-EVFI [14], as shown in Tab. 1. To ensure the alignment between the RGB camera and the events, HS-ERGB [18] requires the system to be static and objects of interest should only move in a fronto-parallel plane at a predetermined depth. BS-ERGB [19] addresses these issues using a co-axial structure, capturing a variety of challenging scenes including linear and non-linear movements. However, the camera frame rate remains low, resulting in a dataset limited to 28fps, which can lead to significant occlusions in the scenes. To mitigate this, ERF-X170FPS [12] employs a higher-speed camera, achieving a dataset with 170fps, but it faces issues with the misalignment of RGB and events. HQ-EVFI [14] aims to reduce motion blur and noise by using short-exposure shots with supplementary lighting, resulting in non-uniform illumination in some scenes. To address these issues and focus more

Dataset	RGB Camera Sensor	FPS	Characteristics
HS-ERGB[18]	FLIR BFS-U3-16S2C-CS 1440×1080	226	Static cameras; Predetermined depth
BS-ERGB[19]	FLIR BFS-U3-89S6C-C 4096×2196	42	Challenging scenarios; Low frame rate
ERF-X170FPS[12]	FLIR BFS-U3-16S2C-CS 1440×1080	226	High diversity of scenarios; Misalignment of RGB and events
HQ-EVFI[14]	MER2-301-125U3C 2048×1536	142	Low noise and low motion blur; Uneven lighting in indoor scenarios
CHMD	FLIR BFS-U3-04S2C-CS 720×540	522	High frame rate; High-Speed Nonlinear Motion scenarios

Table 1. Quantitative results on DSEC benchmark.



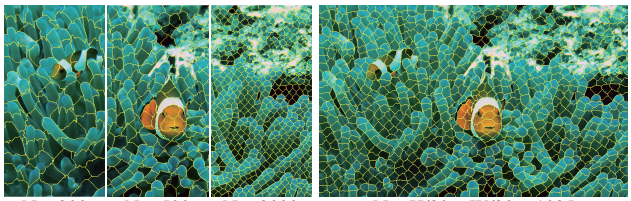
(a) Coaxial imaging system

(b) Fast nonlinear motion scenarios

Figure 1. Features of the proposed CHMD. (a) The implementation of our coaxial imaging system. (b) CHMD includes high-speed nonlinear scenarios andcollects events that are aligned with high-frame-rate images at the pixel level.

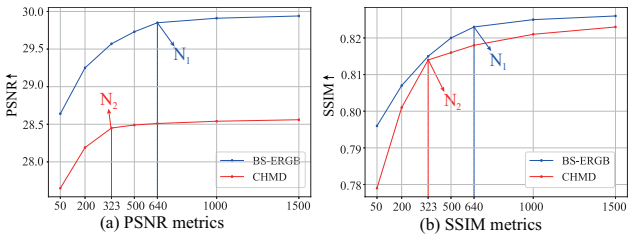
on extremely high-speed moving targets, we propose a new evaluation benchmark called CHMD.

First, we constructed a co-optical axis imaging system comprising an event camera (Prophesee EVK4, 1280×720), a high-speed camera (FLIR BFS-U3-04S2C-CS, 720×540), and a beam splitter (Thorlabs BSW26R), as illustrated in Fig. 1 (a). The beam splitter divides the incoming light into two equal parts and directs them to the event camera and the high-speed camera respectively. Additionally, we provided external trigger signals to the cameras through a programmable synchronous circuit, enabling precise synchronization of the timestamps of both cameras. Finally, we achieved pixel alignment between the two cameras through the stereo rectification process. We collected data at three frame rates: 100, 300, and 500fps, with a particular emphasis



(a) Fixed number of cluster centers (b) Dynamic number of cluster centers

Figure 2. Visual comparison of (a) fixed cluster center numbers and (b) dynamic cluster center numbers.



(a) PSNR metrics (b) SSIM metrics

Figure 3. Comparison of different settings for the number of cluster centers N. N_1 and N_2 represent the number of cluster centers calculated based on Sec. 1.1 on the BS-ERGB [19] and CHMD datasets, respectively, while $[50, 200, 500, 1000, 1500]$ are manually set fixed cluster center numbers.

on high-speed non-linear moving targets.

The CHMD contains 90 sequences, which includes static platform nonlinear moving targets (e.g., swinging a stick, fan rotation, spinning top) and dynamic scenes captured by rapidly shaking the camera, as shown in Fig. 1 (b).

2. Ablation Study and Discussion

2.1. Number of SLIC Segmented Regions

In SLIC, each cluster center corresponds to an image segmentation region. Fewer cluster centers lead to fewer image blocks for tracking, reducing computational cost, whereas more cluster centers increase the number of blocks and computational complexity.

Fig. 2 demonstrates the visual effects of different numbers of cluster centers, while Fig. 3 quantitatively compares their impact on the BS-ERGB [19] and CHMD datasets. We set $N \in [50, 200, 500, 1000, 1500]$ for fixed cluster center numbers and calculated dynamic cluster center numbers $N_1 = 640$ (BS-ERGB: 970×625) and $N_2 = 323$ (CHMD: 592×536) based on the resolutions of the datasets. The results show that as the number of cluster centers increases, both PSNR and SSIM metrics improve; however, the rate of improvement diminishes while the computational cost rises significantly. The dynamically calculated cluster center numbers, N_1 and N_2, are near the inflection point of the curve, striking an effective balance between computational cost and accuracy.

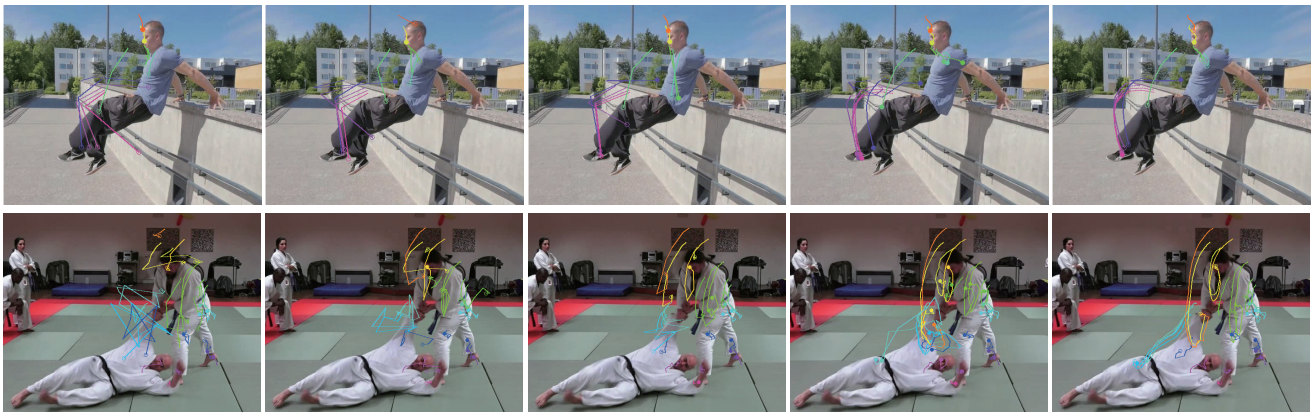


Figure 4. Visual comparison of the point tracking module of TimeTracker and other SOTA methods on the TAP-Vid-DAVIS benchmark.

2.2. Comparison of Point Tracking Results

Comparison Methods. To validate the effectiveness of the point tracking module in TimeTracker, we compare it with five SOTA point tracking methods, including three frame-based methods: TAP-Net [3], TAPIR [4], and Cotracker [11], as well as two event-based methods: EV_Tracker [15] and MotionPriorCM [9]. All methods are evaluated on the TAP-Vid-DAVIS [3] benchmark, which includes 30 real-world sequences and provides accurate point tracking annotations. For event-based methods, we use ESIM [5] to convert images into events as input. All methods are fine-tuned on TAP-Vid-Kubric [3] data. The image sequence is provided with a one-frame interval to simulate high-speed nonlinear scenarios.

Evaluation Metrics. We follow the evaluation metrics adopted by existing methods [3, 4, 11]: (1) Occlusion Accuracy (OA, average position accuracy of visible points and binary occlusion accuracy), (2) δ^x (The positional accuracy is calculated only for frames where the points are visible). δ^x_{avg} averages across 5 thresholds: 1,2,4,8, and 16 pixels, and (3) Average Jaccard (AJ, measuring jointly geometric and occlusion prediction accuracy). The images are resized to 256×256 before calculating δ^x_{avg} and AJ.

Qualitative and Quantitative Results. Fig. 4 and Tab. 2 show the quantitative and qualitative results on the TAP-Vid-DAVIS benchmark. Please note that since object motion in the TAP-Vid-DAVIS dataset is relatively slow, each input frame skips the next two frames to simulate high-speed motion. Frame-based methods TAP-Net [3], TAPIR [4], and Cotracker [11], even when capable of correctly tracking the positions of points at different times, produce tracking trajectories resembling polylines due to the lack of inter-frame information. The event-based methods EV_Tracker [15] and MotionPriorCM [9] tends to suffer from mismatches between points. In contrast, our method accurately tracks the motion trajectories of points.

Methods	Input	AJ ↑	δ^x_{avg} ↑	OA ↑
TAP-Net [3]	Frame	31.5	46.7	76.2
TAPIR [4]	Frame	53.4	64.9	84.6
Cotracker [11]	Frame	55.2	65.1	85.9
EV_Tracker [15]	Events	—	61.5	—
MotionPriorCM [9]	Events	—	68.2	—
Ours	Events	**56.7**	**71.3**	**86.2**

Table 2. Quantitative results on TAP-Vid-DAVIS benchmark.

Methods	Input	7skip		15skip	
		PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
TAP-Net [3]	Frame	32.72	0.905	30.43	0.902
TAPIR [4]	Frame	34.15	0.917	32.04	0.908
Cotracker [11]	Frame	34.72	0.935	32.96	0.924
EV_Tracker [15]	Events	34.86	0.938	33.94	0.933
MotionPriorCM [9]	Events	35.84	0.950	34.47	0.942
Ours	Events	**37.05**	**0.957**	**36.41**	**0.955**

Table 3. Comparison of point tracking methods in the VFI task.

Additionally, we test the results of replacing the point tracking module in the TimeTracker framework with other methods. All methods are fine-tuned (50k iterations) and tested on the GoPro dataset [16]. As shown in Tab. 3, the proposed tracking method delivers the best performance. Note that, point tracking in our framework is a pluggable module, allowing more advanced methods to be integrated in the future to further enhance the performance of our framework.

2.3. Comparison of Optical Flow Results

Comparison Methods. To validate the effectiveness of the optical flow module in TimeTracker, we compare it with five SOTA optical flow estimation methods, including three frame-based methods: supervised RAFT [17] and Flowformer [10], unsupervised ARFlow [13], as well as two event-based methods: E-RAFT [7] (Event-only) and BFLow [8] (Event-Frame hybrid). All methods are evaluated on

Figure 5. Visual comparison of the optical flow module of TimeTracker and other SOTA methods on the DSEC flow benchmark.

Methods	Input	EPE ↓	AE ↓
RAFT [17]	Frame	0.79	3.46
Flowformer [10]	Frame	0.72	2.73
ARFlow [13]	Frame	0.83	3.01
E-RAFT [7]	Events	0.90	3.12
BFLow [8]	Frame+Events	0.85	2.97
Ours	Frame+Events	**0.70**	**2.68**

Table 4. Quantitative results on DSEC benchmark.

the DSEC [6] benchmark, which provides both events and frames along with sparse optical flow ground truth. All methods are fine-tuned on DSEC [6] data.

Evaluation Metrics. We follow recommendations of DSEC [6], utilizing EPE (Endpoint Error, the average of the L2-Norm of the optical flow error) and AE (Angular Error) as evaluation metrics.

Qualitative and Quantitative Results. Fig. 5 and Tab. 4 show the quantitative and qualitative results on the DSEC benchmark. It is evident that event-based optical flow estimation results are worse than frame-based methods due to the sparsity of events, which makes it challenging to establish accurate pixel correspondences between adjacent frames. Our method circumvents this ill-posed problem by segmenting the image into small blocks and leveraging continuous-time tracking, thereby enabling precise dense and continuous time optical flow estimation.

2.4. Ablation study of the loss function

In Tab. 5, we evaluate the impact of various loss functions on the final metrics using the GoPro dataset [16]. The baseline model employs only the reconstruction loss $\mathcal{L}_{rec}$. The occlusion loss $\mathcal{L}_{occ}$ has a negligible effect when used alone. The tracking loss $\mathcal{L}_{track}$, representing the pretraining of the point tracking module, and the global optical flow optimization loss $\mathcal{L}_{flow}$ both significantly contribute to reconstruction quality. The best performance is achieved when all losses are combined.

$\mathcal{L}_{track}$	$\mathcal{L}_{occ}$	$\mathcal{L}_{flow}$	$\mathcal{L}_{rec}$	PSNR ↑	SSIM ↑
			✓	31.57	0.928
✓			✓	34.91	0.951
	✓		✓	31.59	0.929
		✓	✓	33.87	0.946
✓	✓	✓	✓	37.13	0.962

Table 5. Ablation study of the loss functions.

3. Additional Results

Additional qualitative results on four different datasets (SNU-FLIM [2], GoPro [16], BS-ERGB [19] and CHMD) are shown in Fig. 6. Frame-based methods perform well in slow-motion scenes, but they tend to produce reconstruction artifacts and noticeable object errors in fast and complex motion scenarios due to inaccurate motion estimation. On the other hand, event-based methods suffer from reconstruction errors due to the sparsity of events. In comparison, TimeTracker outperforms state-of-the-art methods in visual quality on both simulated and real-world datasets.

References

[1] 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. 1

[2] 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*, pages 10663–10671, 2020. 4

[3] 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

[4] 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. 3

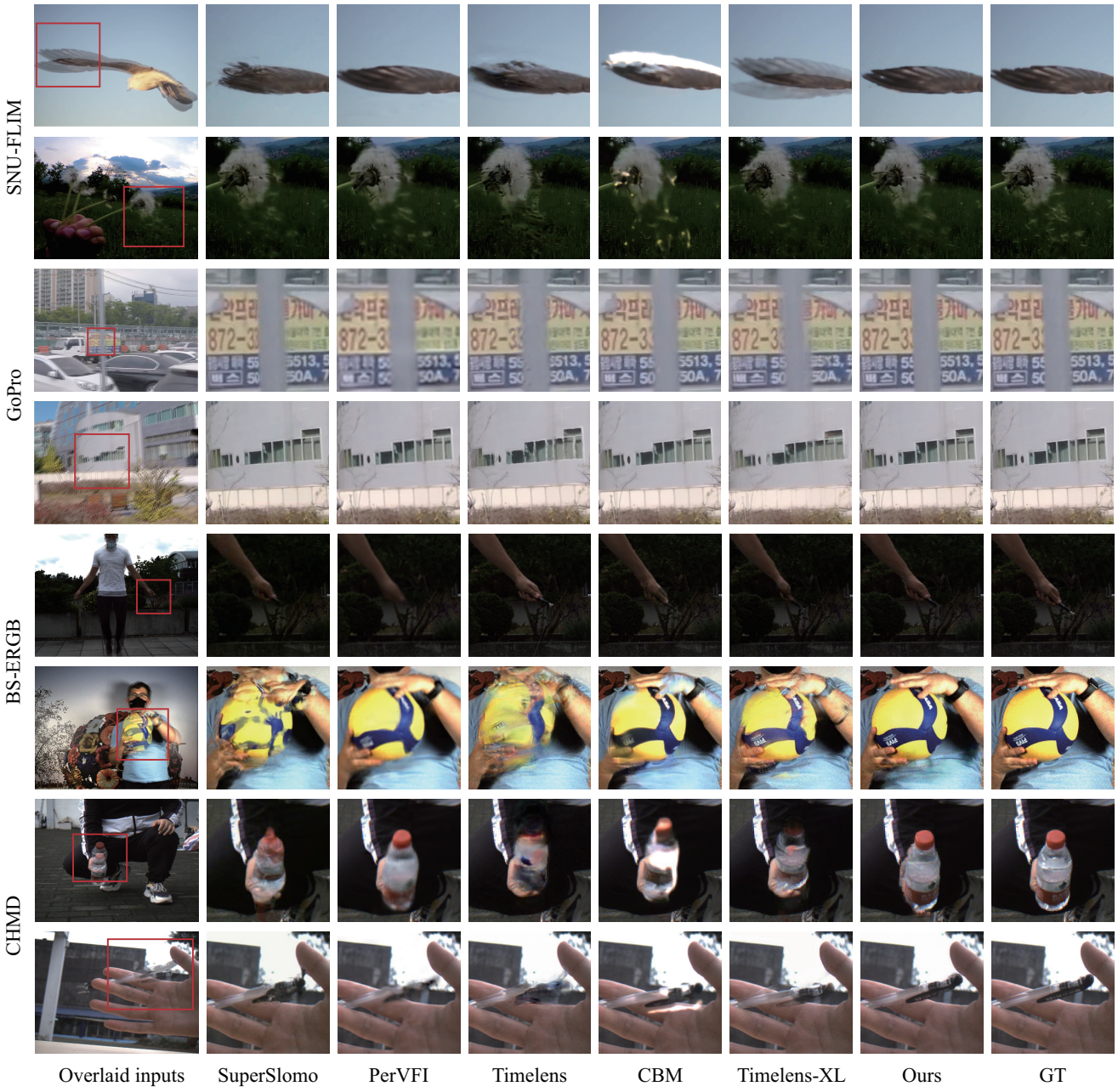


Figure 6. Visual comparison of the proposed method and other SOTA methods across different datasets

[5] 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. 3

[6] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robot. Autom. Lett.*, 6(3):4947–4954, 2021. 4

[7] 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. 3, 4

[8] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024. 3, 4

[9] 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

[10] 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. 3, 4

[11] 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

[12] 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. 1

[13] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6489–6498. 3, 4

[14] 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. 1

[15] 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. 3

[16] 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. 3, 4

[17] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, pages 402–419, 2020. 3, 4

[18] 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. 1

[19] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric nonlinear flow and multi-scale fusion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 17755–17764, 2022. 1, 2, 4