TAPFormer: Robust Arbitrary Point Tracking via Transient Asynchronous Fusion of Frames and Events

Supplementary Material

1. Additional Method Details

Transformer-based Trajectory Refinement. In the main paper, we introduced the multimodal fusion network and the construction of the multi-scale fused feature pyramid. Here, we describe how these features are used in a transformer-based trajectory optimizer (based on Cotracker3[10]) to iteratively refine the point coordinates and visibility.

Let $\mathcal{P} = \{\mathbf{P}^{(l)} \mid l = 0, 1, 2\}$ denote the three-level fused feature pyramid extracted from our model. Given a temporal window of size $W = 16$, the initial point state is representing the coordinate sequence and visibility estimates:

$$\mathbf{x} = \{(x_t, y_t)\}_{t=1}^{W}, \qquad \mathbf{v} = \{v_t\}_{t=1}^{W}, \tag{1}$$

At each refinement iteration, we extract a local feature patch around the predicted coordinates from each pyramid level. For scale l, a $(2r+1) \times (2r+1)$ patch with $r = 3$ is sampled using an operator $S(\cdot)$:

$$\mathbf{f}_t^{(l)} = S\Big(\mathbf{P}_t^{(l)}, (x_t, y_t), r\Big) \in \mathbb{R}^{(2r+1)\times(2r+1)\times C_l}. \tag{2}$$

To measure temporal consistency across the window, we compute correlations between $\mathbf{f}_t^{(l)}$ and the corresponding sampled patches at all other frames, resulting in a correlation matrix of size $(2r+1)^2 \times (2r+1)^2$. Each matrix is flattened and encoded by an MLP, and embeddings from all three scales are concatenated to obtain the final correlation descriptor.

We further include a positional encoding of relative motion across frames. The correlation embeddings, visibility estimates, and motion encoding are concatenated and passed through a spatio-temporal transformer, which predicts residual updates:

$$\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}, \qquad \mathbf{v} \leftarrow \mathbf{v} + \Delta\mathbf{v}. \tag{3}$$

We perform three refinement iterations. Thanks to the discriminative and temporally consistent fused feature pyramid, our optimizer converges quickly, unlike prior approaches (e.g., CoTracker3[10]) that typically require six iterations. This allows us to maintain high accuracy while significantly reducing computational cost.

2. Additional Ablation Experiments

Event Representation. To leverage the spatio-temporal information in asynchronous event streams, the sparse events are typically converted into dense tensor-like representations. This conversion (i) enables direct use of standard convolutional architectures and (ii) improves robustness by aggregating events within short temporal intervals.

Table 1. Effect of event representation on tracking performance.

Representation	EDS		EC	
	FA↑	EFA↑	FA↑	EFA↑
Event image[16]	81.4	69.5	90.8	90.2
Voxel grid[24]	82.5	69.8	92.1	91.5
Time surface[12]	82.3	70.4	93.3	92.6

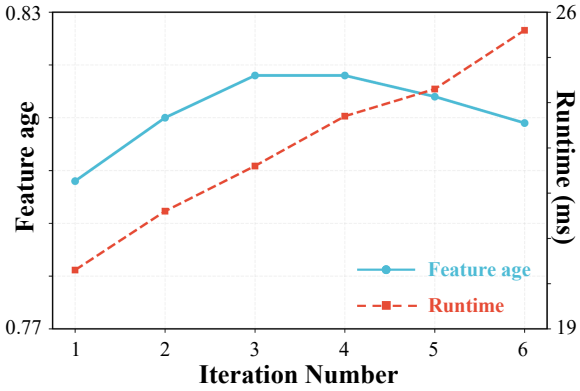
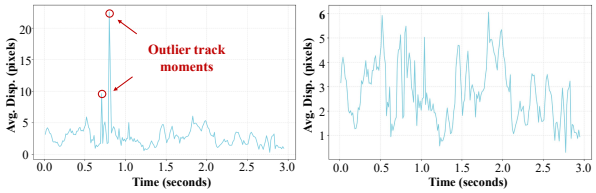


Figure 1. Performance and efficiency across tracking iterations. The left y-axis reports the feature age (higher is better), while the right y-axis shows the inference throughput measured in frames per millisecond (FPS/ms). The x-axis denotes the number of trajectory refinement iterations. This dual-axis plot illustrates how tracking accuracy and computational efficiency evolve as the number of iterations increases.

We evaluate three widely used representations: Event Image [16], Voxel Grid [24], and Time Surfaces [12]. Following the Stacking Based on Time (SBT) scheme [20], a fixed temporal window is divided into five sub-windows, and each representation is computed separately for every sub-window. Event Image accumulates event counts, Voxel Grid constructs a discretized spatio-temporal volume, and Time Surfaces encode the latest event timestamp in a decay-like form. All representations are evaluated on the EDS and EC datasets under identical training settings, and the quantitative results feature age (FA) and experted feature age (EFA), which quantify the duration until a track deviates beyond a threshold distance from the GT, are summarized in Tab. 1. Given its higher accuracy and lower computational cost, we adopt Time Surfaces as our event representation.



(a) Pre-correction trajectory (b) Post-correction trajectory

Figure 2. Ground-truth trajectory correction for the peanuts light sequence. (a) Pre-correction trajectory with two displacement outliers. (b) Post-correction trajectory after smoothing, producing a stable and reliable ground truth.

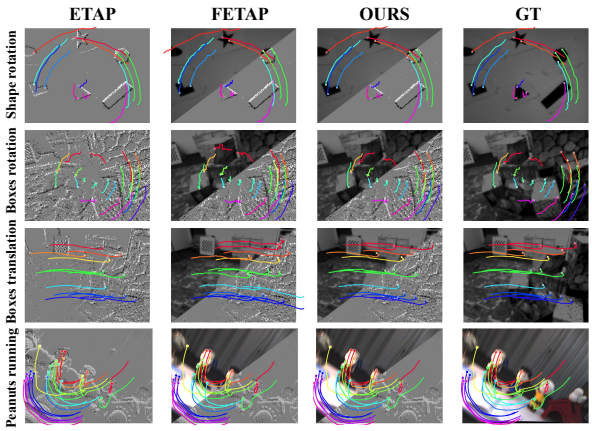


Figure 3. Additional visualizations on EC and EDS dataset.

Iteration Number. We analyze how the number of refinement iterations influences both accuracy and runtime. Increasing iterations improves performance but introduces an almost linear increase in computation. As shown in the two-axis performance–time curve (see Fig. 1), performance saturates around the third iteration: subsequent iterations yield only marginal gains while noticeably increasing cost. Thus, we adopt three iterations to balance accuracy and efficiency. This fast convergence reflects the stability and discriminative strength of our fused features, which provide reliable cues for trajectory refinement.

3. Extended Experimental Results

We provide additional quantitative and qualitative results to complement the main paper. Tab. 2 reports the detailed results on the InivTAP dataset. Tab. 5 and Tab. 4 list the full evaluation results on the EDS and EC datasets. Fig. 3 presents extra qualitative comparisons.

We additionally perform ground-truth trajectory correction for the peanuts light sequence in the EDS dataset. As shown in Fig. 2, the original ground-truth trajectory contains two displacement outlier points, which we smooth during correction. As reported in Tab. 5, the correction leads to performance improvements across all sequences and yields more distinguishable evaluation metrics.

Table 2. Performance comparison across different scenarios and modalities.

Scenario	Method	Modality	AJ↑	$\delta^{\text{vis}}_{\text{avg}}$ ↑	OA↑
Normal	ETAP	Event	19.7	33.6	88.8
	CoTracker3	Frame	61.3	77.1	92.1
	Ours	F + E	71.1	84.8	92.7
Fast	ETAP	Event	27.6	40.6	89.2
	CoTracker3	Frame	28.0	36.7	73.5
	Ours	F + E	39.5	52.3	95.4
Overexposure	ETAP	Event	11.8	22.3	77.8
	CoTracker3	Frame	30.2	43.9	70.6
	Ours	F + E	38.6	50.7	89.6
Static	ETAP	Event	10.6	22.3	71.8
	CoTracker3	Frame	53.4	72.2	88.6
	Ours	F + E	78.9	90.0	94.9

4. Application to SLAM

To demonstrate the practical value of our tracker, we integrate TAPFormer into a visual SLAM pipeline and evaluate it under challenging scenarios. The system includes feature detection, feature tracking, and backend pose optimization. We use SuperPoint [3] to extract well-distributed keypoints, followed by a lightweight management module to maintain stable feature coverage. TAPFormer then produces robust inter-frame correspondences. For the backend, we directly reuse the VINS-Mono [18] optimization module to estimate high-frequency camera poses.

Evaluation Metric and Baselines. To evaluate velocity and pose accuracy in a principled manner, we adopt the speed-weighted success metric commonly used in recent event-based SLAM work. Let RVE_i denote the relative velocity error at timestamp i, $\mathbf{v}_{\text{gt},i}$ the ground-truth velocity, and ξ an error threshold. The success score is defined as

$$S_\xi = \frac{\sum_{i=1}^{N} \|\mathbf{v}_{\text{gt},i}\| \cdot \delta(\text{RVE}_i < \xi)}{\sum_{i=1}^{N} \|\mathbf{v}_{\text{gt},i}\|}, \tag{4}$$

where $\delta(\cdot)$ is the Dirac indicator. This formulation emphasizes accurate estimates during high-speed motion by weighting each success according to the magnitude of the ground-truth velocity. By sweeping ξ over the interval $[0, 1]$, a success-rate curve is obtained, and the final AUC_v provides a unified, speed-aware scalar metric. We compare our system against two event-based SLAM baselines: 1) SuperEvent[2] + OKVIS2[13], which uses stereo events, stereo images, and IMU measurements; 2) SDEVO[22], which operates purely on stereo events.

Results. Fig. 4 presents qualitative tracking results by TAPFormer, and Tab. 3 summarizes the final AUC_v scores. Our approach achieves the best performance among all evaluated methods.

Table 3. Comparison with event-based SLAM methods. We compare our approach with two event-based SLAM baselines. After directly replacing the VINS-Mono frontend with ours, our method achieves the best performance.

Method	Modalities			AUC_v↑
	Imu Used	Events Used	Images Used	
SDEVO [22]	none	stereo	none	6.13
SuperEvent [2]	yes	stereo	stereo	6.27
Ours	yes	monocular	monocular	**6.29**

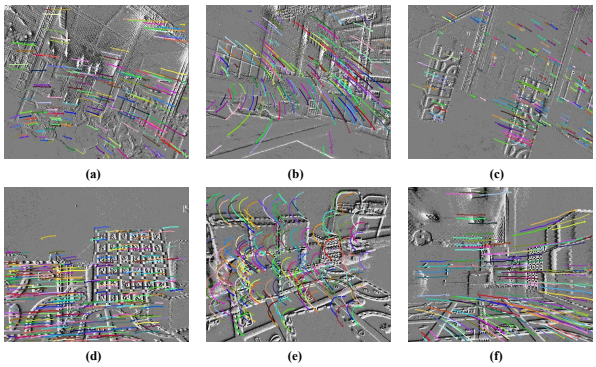


Figure 4. Visualization of our tracking results on SLAM sequences. (a–c) show results on high-speed UAV scenarios, while (d–e) present tracking results on ground-based high-speed sequences.

5. Dataset Details

We construct a large-scale synthetic dataset and provide two real, manually annotated frame-event TAP test sets. In the following, we describe the datasets in detail.

FE-FastKub Our FE-FastKub dataset differs significantly from existing TAP training sets. Specifically, it not only contains high-frame-rate photorealistic RGB images (with motion blur considered) but also provides corresponding synthetic event streams. To enhance the model's adaptability to fast-moving scenes, we further include explicit modeling of rapid motion scenarios. See Tab. 6

InivTAP and DrivTAP In addition to the synthetic training set, we introduce two real-world frame–event TAP test sets, both manually annotated with point trajectories and occlusion labels. The experimental data collection equipment is shown in Fig. 5, and the ground truth annotation tool is shown in Fig. 8. These datasets serve as reliable benchmarks for evaluating asynchronous frame–event tracking performance in real scenes, shown in Fig. 6.

Each dataset follows a three-stage annotation pipeline to ensure trajectory accuracy: (1) Initial motion trajectories are generated using RAFT [6] optical flow. (2) Human annotators refine each trajectory by referencing both the image

(a) Recording platform (b) Our devices (c) DAVIS346

Figure 5. Data collection setup. (b) Our custom synchronized capture device is mounted on the (a) vehicle platform to record DrivTAP, and (c) a DAVIS346 camera is used to collect the InivTAP dataset.

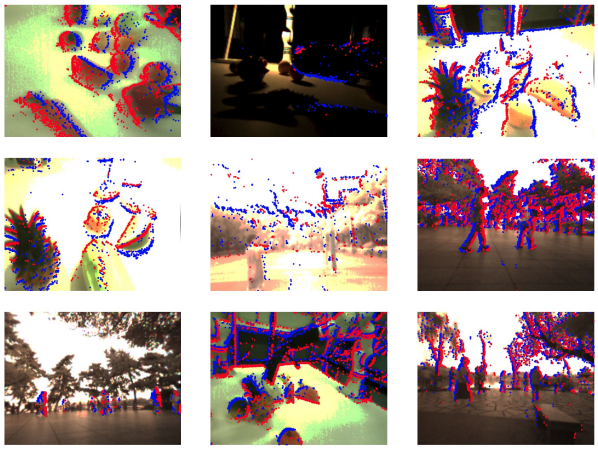


Figure 6. A few examples of InivTAP.

frames and the event-reconstructed frames (obtained using E2Vid [19]); the annotation is performed on the modality that offers clearer visual cues. (3) High-frequency noise is filtered to obtain the final smooth ground-truth trajectories.

For point selection, we prioritize targets that (i) remain visible for long durations, (ii) exhibit noticeable motion, and (iii) distributed across different moving objects, with no more than three points per object (five if one object moves).

Below, we summarize several key characteristics of our manually annotated test datasets. As shown in Fig. 7, most point trajectories span nearly the entire video, with over 60% of tracks present in more than 90% of the sequence duration (though some may be occluded). In total, 80.04% of points are never occluded, and most trajectories consist of a single continuous unoccluded segment.

Our datasets also exhibit substantial motion diversity. A notable portion of points undergo large displacements: 16.51% move more than 5% of the image height between consecutive frames, and 8.1% exceed 10%, indicating significant motion dynamics. Meanwhile, another portion of points remains nearly static, further highlighting the diversity of motion patterns covered by our dataset.

In addition, the test sequences include challenging conditions such as overexposure and low-light scenes, enabling a comprehensive evaluation of tracking performance across a broad range of real-world environments.

Table 4. Tracking performance comparison on the EC dataset. The first four rows list frame-only methods, followed by five event-only baselines, and the last four rows present approaches fusing frames and events. Methods with a gray background denote our models. All results are reported in percentages, and the best performance in each column is highlighted in **bold**.

Method	Average		shapes_trans		shapes_rot		shapes_6dof		boxes_trans		boxes_rot	
	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EAF↑	FA↑	EFA↑	FA↑	EFA↑
PIPs++ [21]	82.6	82.3	87.1	87.1	79.0	78.8	83.7	82.7	86.3	86.1	77.0	76.9
Cotracker3 [10]	92.5	91.9	94.4	94.3	87.3	86.8	93.3	91.6	**93.6**	**93.2**	94.3	93.9
Chrono [11]	82.9	82.3	85.1	84.5	81.5	81.1	75.6	74.1	83.3	83.0	88.8	88.6
TAPFormer-F	92.7	92.1	95.2	95.1	87.8	87.4	**92.8**	**91.1**	92.9	92.4	94.7	94.4
EM-ICP [23]	33.7	33.4	40.3	40.2	32.0	32.0	24.8	24.2	35.5	35.4	35.6	34.9
HASTE [1]	44.2	42.7	58.9	56.4	61.3	58.2	13.3	4.3	38.2	36.8	49.2	44.7
ETAP [8]	88.1	87.6	91.7	91.5	87.1	86.9	90.5	88.6	85.0	84.9	86.0	85.9
MATE [9]	88.5	87.5	-	-	-	-	-	-	-	-	-	-
TAPFormer-E	86.6	86.1	93.3	92.9	81.8	81.6	91.4	89.8	89.0	88.8	77.5	77.3
EKLT [5]	81.1	77.5	83.9	74.0	83.3	80.6	81.7	69.6	68.2	64.4	88.3	86.5
DeepEvT [17]	82.5	81.8	86.1	86.5	79.7	79.3	89.9	88.2	87.2	86.9	69.5	69.1
FE-TAP [15]	84.4	83.8	93.1	92.9	81.5	81.3	87.9	86.0	73.1	72.8	86.2	86.1
TAPFormer	**93.3**	**92.6**	**96.1**	**96.0**	**88.7**	**88.3**	**92.8**	**91.1**	93.2	92.6	**94.8**	**94.4**

Table 5. Tracking performance comparison in EDS dataset. The values in parentheses indicate the performance of the Peanuts Light sequence under ground-truth evaluation before trajectory correction. For the Average column, the value in parentheses indicates the mean score after replacing only the Peanuts Light result. Rocket Earth* indicating that the ground truth is not fully accurate. All results are reported in percentages, and the best performance in each column is highlighted in **bold**.

Method	Average		Peanuts Light		Rocket Earth*		Ziggy Arena		Peanuts Running	
	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑	FA↑	EFA↑
PIPs++ [21]	75.1(72.1)	63.0(60.3)	62.6(50.9)	58.8(47.9)	76.4	34.2	85.1	85.0	76.1	74.1
cotracker3 [10]	80.2(75.8)	68.8(64.7)	72.7(54.9)	68.4(51.9)	69.2	30.7	92.6	92.5	86.4	83.7
TAPFormer-F	80.9(76.4)	69.1(64.9)	74.7(56.7)	70.1(53.3)	71.5	31.5	91.8	91.8	85.6	82.8
EM-ICP [23]	16.1	12.0	8.4	7.7	29.8	15.8	15.3	14.9	10.8	9.5
HASTE [1]	9.6	6.3	8.6	7.6	16.2	8.5	8.2	5.7	5.4	3.3
ETAP [8]	74.5(70.1)	63.9(60.1)	70.2(53.7)	66.1(50.7)	67.6	33.3	83.8	83.7	76.2	72.6
MATE [9]	(71.3)	(62.6)	-	-	-	-	-	-	-	-
TAPFormer-E	76.8(72.4)	64.5(60.4)	67.7(50.2)	63.7(47.4)	**80.0**	**37.2**	81.3	81.1	78.1	75.9
EKLT [5]	32.5	20.5	28.4	26.0	42.5	17.5	41.9	23.1	17.1	15.3
DeepEvT [17]	61.3(57.6)	50.5(47.2)	59.4(44.7)	55.5(42.0)	64.8	29.1	74.8	74.6	46.0	42.8
FE-TAP [15]	72.2(67.6)	63.2(58.9)	73.1(54.9)	68.9(51.7)	53.8	24.6	84.9	84.4	76.9	74.9
TAPFormer	**82.3(77.6)**	**70.4(66.1)**	**76.5(57.9)**	**71.5(54.4)**	73.8	34.1	**92.7**	**92.7**	**86.1**	**83.3**

Table 6. Dataset comparison. Overview of publicly available synthetic event-based motion estimation datasets.

Dataset	Source	Events	Final image	Fast scene	#Samples	Resolution	fps[Hz]	sample duration[s]	Annotations			
									optical flow	TAP	depth	segmentations
TAP-Vid Kubric[4]	3D PBR	none	✓	×	≈ 10000	512 × 512	12	2	✓	✓	✓	✓
BlinkFlow[14]	3D PBR	synthetic	✓	×	3587	640 × 480	10	1	✓	×	✓	✓
MultiFlow[7]	2D warp	synthetic	×	×	12100	512 × 384	100	0.5	✓	×	×	×
EventKubric[8]	3D PBR	synthetic	×	×	10173	512 × 512	48	2	✓	✓	✓	✓
FE-FastKub(Ours)	3D PBR	synthetic	✓	✓	10953	512 × 512	48	2	✓	✓	✓	✓

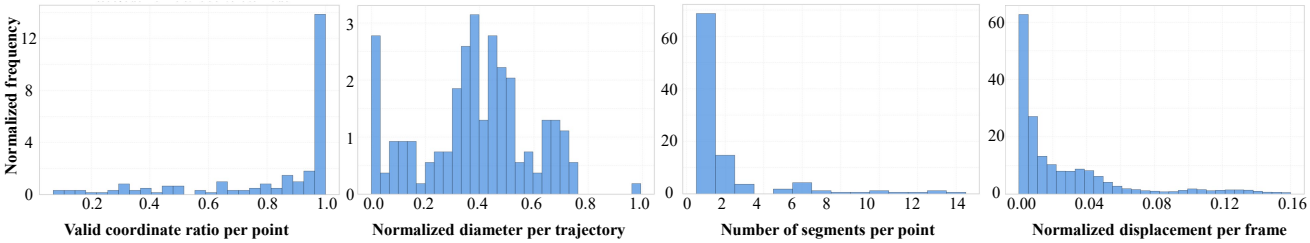


Figure 7. Statistics of ground-truth trajectories in InivTAP and DrivTAP. Valid coordinate ratio measures the percentage of time a queried point remains within the image boundaries. Diameter denotes the trajectory's motion range normalized by image height. Number of segments indicates how many continuous track fragments are formed due to occlusions. Displacement represents the per-frame motion magnitude normalized by image height.

Figure 8. Annotation interface for the TAP task. We select either RGB frames or event-reconstructed frames for manual ground-truth labeling, depending on which modality provides clearer visual cues. The tool consists of three main components: a visualization panel for displaying the sequence, an annotation information panel for managing point information, and a button panel providing operation interactive controls.

References

[1] I. Alzugaray and M. Chli. HASTE: multi-hypothesis asynchronous speeded-up tracking of events. In *31st British Machine Vision Conference (BMVC)*, page 744, 2020. 4

[2] Yannick Burkhardt, Simon Schaefer, and Stefan Leutenegger. Superevent: Cross-modal learning of event-based keypoint detection. *arXiv preprint arXiv:2504.00139*, 2025. 2, 3

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 2

[4] 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. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. 5

[5] 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. 4

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

[7] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4736–4746, 2024. 5

[8] Friedhelm Hamann, Daniel Gehrig, Filbert Febryanto, Kostas Daniilidis, and Guillermo Gallego. Etap: Event-based tracking of any point. In *Proceedings of the Computer Vision and Pattern Recognition Conference. (CVPR)*, pages 27186–27196, 2025. 4, 5

[9] Han Han, Wei Zhai, Yang Cao, Bin Li, and Zheng-jun Zha. Mate: Motion-augmented temporal consistency for event-based point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision. (ICCV)*, pages 8340–8349, 2025. 4

[10] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision. (ICCV)*, pages 6013–6022, 2025. 1, 4

[11] Inès Hyeonsu Kim, Seokju Cho, Jiahui Huang, Jung Yi, Joon-Young Lee, and Seungryong Kim. Exploring temporally-aware features for point tracking. In *Proceedings of the Computer Vision and Pattern Recognition Conference. (CVPR)*, pages 1962–1972, 2025. 4

[12] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359, 2016. 1

[13] Stefan Leutenegger. Okvis2: Realtime scalable visual-inertial slam with loop closure. *arXiv preprint arXiv:2202.09199*, 2022. 2

[14] 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 *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3881–3888. IEEE, 2023. 5

[15] 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. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 18834–18840, 2025. 4

[16] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5419–5427, 2018. 1

[17] Nico Messikommer, Carter Fang, Mathias Gehrig, and Davide Scaramuzza. Data-driven feature tracking for event cameras. In *IEEE Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages 5642–5651, Vancouver, BC, Canada, 2023. 4

[18] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4):1004–1020, 2018. 2

[19] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages 3857–3866, 2019. 3

[20] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (CVPR)*, pages 10081–10090, 2019. 1

[21] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision. (ICCV)*, pages 19855–19865, 2023. 4

[22] Sheng Zhong, Junkai Niu, and Yi Zhou. Deep visual odometry for stereo event cameras. *IEEE Robotics and Automation Letters*, 2025. 2, 3

[23] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based feature tracking with probabilistic data association. In *IEEE International Conference on Robotics and Automation. (ICRA)*, pages 4465–4470, 2017. 4

[24] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 989–997, 2019. 1