# Supplementary: Data-driven Feature Tracking for Event Cameras

Nico Messikommer* Carter Fang* Mathias Gehrig Davide Scaramuzza

Robotics and Perception Group, University of Zurich

## 1. Future Work & Limitations

Since the EC and EDS datasets were recorded to benchmark pose estimation algorithms, they only contain static scenes. Thus, we did not evaluate how our method, and especially our frame attention module performs in scenes with dynamic objects. Nevertheless, we believe that our frame attention module can be useful for other trackers using event or standard cameras. Finally, our method relies on the quality of the feature detection in grayscale images, which can suffer in challenging scenarios. However, our self-supervision strategy opens up the possibility of also fine-tuning feature detectors for event cameras to increase the robustness of feature detection.

## 2. Dataset Split

We use five sequences from the Event Camera dataset [9] (EC) and four sequences from the Event-aided Direct Sparse Odometry dataset [5] (EDS) as test sequences. For fine-tuning, our pose supervision strategy is performed on five sequences from the EC and one sequence from the EDS dataset since EDS does not contain many sequences with ground truth pose in well-lit conditions. The overview of the test and fine-tuning sequences is shown in Tab. 1.

## 3. Multiflow Dataset

To qualitatively show the gap between the simulated and the real data, we visualize in Fig. 1 some examples from the Multiflow dataset [3], including the ground truth tracks corresponding to the extracted Harris features [4]. This sim-to-real gap can be reduced with our augmentation strategies on the Multiflow dataset and with our proposed fine-tuning strategy on real data, see Sec. 3.3.

## 4. Network Architecture Details

Tab. 2 shows the architectural details of our proposed network, which consists of a feature network and our proposed frame attention module. In the first step, two patch encoders inside the feature network process the event and the grayscale patches, which have a patch size of 31 pixels. After the correlation and the concatenation of the feature maps from both patch networks, a joint encoder refines the correlation map and introduces temporal information sharing through a ConvLSTM layer. Finally, the frame attention module processes each feature in one frame using shared linear layers and one global multi-head attention over all features in a frame. We refer to Fig. 2 in the main paper for the network overview.

*equal contribution.

Table 1. Test and fine-tuning sequences for the EC and EDS dataset.

| | Dataset | Sequence Name | Frames |
|---|---|---|---|
| Test | EC | Shapes Translation | 8-88 |
| | | Shapes Rotation | 165-245 |
| | | Shapes 6DOF | 485-485 |
| | | Boxes Translation | 330-410 |
| | | Boxes Rotation | 198-278 |
| | EDS | Peanuts Light | 160-386 |
| | | Rocket Earth Light | 338-438 |
| | | Ziggy In The Arena | 1350-1650 |
| | | Peanuts Running | 2360-2460 |
| Fine-Tuning | EC | boxes_hdr | all |
| | | calibration | all |
| | | poster_6dof | all |
| | | poster_rotation | all |
| | | poster_translation | all |
| | EDS | all_characters | all |

## 5. Quantitative Results & Tracking Metrics

As done in previous works [1, 2], we directly compare feature tracking metrics for a feature tracking methodology instead of computing pose errors using a pose estimation module. While pose estimation is one application, it requires the tuning of many hyperparameters specifically for the tracker. Thus, it complicates evaluation and produces biased results.

Figure 1. Samples from the Multiflow dataset including the ground truth tracks corresponding to extracted Harris features.

As tracking metrics, we report for each test sequence from the EC and EDS dataset the *expected feature age* in Tab. 3, the *feature age* in Tab. 4, the *inlier ratio* in Tab. 5 and the *normalized tracking error* in Tab. 6. For the *normalized tracking error*, we terminate the track if the distance to the ground truth exceeds 5 pixels, as done in [1]. However, it is not obvious how to compute this metric if the tracking error is higher than 5 pixels directly after the initialization, as it occurred for the baseline methods in Tab. 6. Furthermore, this metric does not consider the duration of the predicted tracks, e.g., one feature can be tracked for a short time duration with a small tracking error, which would lead to a small normalized tracking error. In contrast, a feature tracked for a long time horizon but with a higher distance to the ground truth will be assigned a higher tracking error. This example shows that the *normalized tracking error* on its own is not necessarily a good metric to evaluate stable and long feature tracks. Thus, we decided to report the *expected feature age* as a metric since it considers the tracking duration and the number of tracked features. Moreover, the *expected feature age* is computed over a range of termination thresholds with respect to the ground truth, which effectively eliminates this hyperparameter for the metric computation. Specifically, the *expected feature age* represents the multiplication of the *normalized feature age* with the fraction of successfully predicted tracks over the number of given feature locations, defined as *inlier ratio*. A feature is defined to be tracked successfully if the predicted feature location at the second timestep after initialization is in the termination threshold to the ground truth location. The *normalized feature age* is computed for the successfully tracked features based on the division of the time duration until the predicted feature exceeds the termination threshold to the ground truth location by the duration of the ground truth tracks. Because of the range of termination thresholds and the consideration of the number of successfully tracked features, the *expected feature age* represents an expressive and objective metric for reporting the tracking performance. Compared to [7], we evaluate the tracking performance and thus use the same features for each method. Furthermore, our evaluation focuses on the introduced Expected Feature Age to account for the impact of outliers, which is typically ignored.

## 6. Input Event Representation

Similar to previous works [2], our method requires spatially and temporally aligned frames and events. This data can be recorded by cameras outputting directly events and images with one sensor (ATIS) or with beam splitter setups using two cameras aligned through a mirror setup. To provide the events in a patch as input to our network, we first convert them to a dense event representation. Specifically, we use a maximal timestamp version of SBT [10], named SBT-Max, which consists of five temporal bins for positive and negative polarity leading to 10 channels. Because of these design choices, the used event representation can be considered a combination between TimeSurface [8] and SBT [10]. In each temporal bin, we assign to each pixel coordinate the relative timestamp of the most recent event during the time interval of the temporal bin. For the EC and EDS dataset, we convert events inside a 10 ms and 5 ms window, respectively.

## 7. Additional Ablation Experiments

In addition to the ablation experiments reported in Tab. 2 in the main paper, we ablated the event input representation as well as the augmentation parameters used during training. Due to time reasons, we performed the following ablation experiments by training the *reference model*, which does not include the frame attention module, for 70000 steps instead of 140000.

Table 2. Network architecture. Each convolution layer is followed by LeakyReLU and BatchNorm layers whereas the linear layers are followed by LeakyReLu layers. For the upsampling layers (Up), we use bilinear interpolation. The three numbers after each convolution layer indicate the two kernel dimensions and the output channel dimension. In the case of the linear layer, the single number stands for the output channels.

| | Layer | Spatial Size |
|---|---|---|
| Feature Network (2×Patch Encoders + Joint Encoder) | 2× Conv2D 1×1×32 | 31×31 |
| | 2× Conv2D 5×5×64 | 23×23 |
| | 2× Conv2D 5×5×128 | 15×15 |
| | 2× Conv2D 3×3×256 | 5×5 |
| | 2× Conv2D 1×1×384 | 1×1 |
| | 2× Conv2D 1×1×384 | 1×1 |
| | Up + Conv2D 1×1×384 | 5×5 |
| | Conv2D 3×3×384 | 5×5 |
| | Up + Conv2D 1×1×384 | 15×15 |
| | Conv2D 3×3×384 | 15×15 |
| | Up + Conv2D 1×1×384 | 23×23 |
| | Conv2D 3×3×384 | 23×23 |
| | Up + Conv2D 1×1×384 | 31×31 |
| | Conv2D 3×3×384 | 31×31 |
| | 2× Conv2D 3×3×384 | 31×31 |
| | Correlation Layer | 31×31 |
| | 2× Conv2D 3×3×128 | 31×31 |
| | 2× Conv2D 3×3×64 | 15×15 |
| | 2× Conv2D 3×3×128 | 7×7 |
| | ConvLSTM 3×3×128 | 7×7 |
| | 2× Conv2D 3×3×256 | 3×3 |
| | Conv2D 3×3×256 | 1×1 |
| Frame Attention | Linear 256 | 1×1 |
| | Linear 256 | 1×1 |
| | MultiHead Attention | 1×1 |
| | LayerScale 256 | 1×1 |
| | Linear Gating 256 | 1×1 |
| | Linear 2 | 1×1 |

### 7.1. Input Representations

The input event representation to an event-based network is an important consideration. Ideally, we aim to preserve as much of the spatiotemporal information as possible while minimizing the computational overhead of representation generations. We train the reference network with different representations: voxel grids [12], Stacking Based on Time (SBT) [10], a non-normalized version of SBT (SBTNo Norm) and a maximal timestamp version of SBT we call SBT-Max where each pixel is assigned the timestamp of the most recent event. The results are shown in Tab. 7. While many event-based networks have demonstrated promising results with voxel grids, their interpolation-based construction is computationally expensive. In contrast, SBT is a simpler, synchronous event representation that is more efficient. Each pixel simply accumulates or "stacks" incoming events. We find that SBT achieves competitive *Expected FA* compared to voxel grids on nearly all sequences. However, the performance of SBT degrades significantly without normalizing based on the number of events in the frame. In contrast to normalizing by the number of events, SBT-Max is normalized using the duration of the time window. In practice, the statistic-free normalization procedure of SBT-Max means that events outside the neighborhoods of tracked features can be ignored. Because of this deployment advantage and the competitive performance despite its more simplistic normalization, we select SBT-Max as event representation.

### 7.2. Augmentation Parameters

To validate the utility of our augmentation strategy, we train the reference network with different augmentation parameters. In Tab. 8, we present the experimental results for using rotations (R) of up to $\pm 30°$, scaling (S) of up to $\pm 10\%$, and translations (T) of up to $\pm 5px$. The default training settings use rotations of up to $\pm 15°$, scaling of up to $\pm 10\%$, and translations of up to $\pm 3px$. Without augmentation, we observe significant degradation on both datasets. The benefit of additional translation augmentation is inconclusive, given the degradation on EC and improvement on EDS. Lastly, with increased rotation augmentation, we observe that the performance improves on average for both datasets.

Table 3. The performance of our proposed and the baseline trackers on the EDS and EC dataset in terms of *Expected Feature Age*.

| | Expected FA ↑ | | | | |
|---|---|---|---|---|---|
| Sequence | ICP [6] | EM-ICP [11] | HASTE [1] | EKLT [2] | **Ours** |
| Shapes Translation | 0.306 | 0.402 | 0.564 | 0.740 | 0.856 |
| Shapes Rotation | 0.339 | 0.320 | 0.582 | 0.806 | 0.793 |
| Shapes 6DOF | 0.129 | 0.242 | 0.043 | 0.696 | 0.882 |
| Boxes Translation | 0.261 | 0.354 | 0.368 | 0.644 | 0.869 |
| Boxes Rotation | 0.188 | 0.349 | 0.447 | 0.865 | 0.691 |
| EC Avg | 0.245 | 0.334 | 0.427 | 0.775 | 0.818 |
| Peanuts Light | 0.044 | 0.077 | 0.076 | 0.260 | 0.420 |
| Rocket Earth Light | 0.045 | 0.158 | 0.085 | 0.175 | 0.291 |
| Ziggy In The Arena | 0.039 | 0.149 | 0.057 | 0.231 | 0.746 |
| Peanuts Running | 0.028 | 0.095 | 0.033 | 0.153 | 0.428 |
| EDS Avg | 0.040 | 0.120 | 0.063 | 0.205 | 0.472 |

Table 4. The performance of our proposed and the baseline trackers on the EDS and EC dataset in terms of *Feature Age FA*.

| | Feature Age (FA) ↑ | | | | |
|---|---|---|---|---|---|
| Sequence | ICP [6] | EM-ICP [11] | HASTE [1] | EKLT [2] | **Ours** |
| Shapes Translation | 0.307 | 0.403 | 0.589 | 0.839 | 0.861 |
| Shapes Rotation | 0.341 | 0.320 | 0.613 | 0.833 | 0.797 |
| Shapes 6DOF | 0.169 | 0.248 | 0.133 | 0.817 | 0.899 |
| Boxes Translation | 0.268 | 0.355 | 0.382 | 0.682 | 0.872 |
| Boxes Rotation | 0.191 | 0.356 | 0.492 | 0.883 | 0.695 |
| EC Avg | 0.256 | 0.337 | 0.442 | 0.811 | 0.825 |
| Peanuts Light | 0.050 | 0.084 | 0.086 | 0.284 | 0.447 |
| Rocket Earth Light | 0.103 | 0.298 | 0.162 | 0.425 | 0.648 |
| Ziggy In The Arena | 0.043 | 0.153 | 0.082 | 0.419 | 0.748 |
| Peanuts Running | 0.043 | 0.108 | 0.054 | 0.171 | 0.460 |
| EDS Avg | 0.060 | 0.161 | 0.096 | 0.325 | 0.576 |

Table 5. The performance of our proposed and the baseline trackers on the EDS and EC dataset in terms of *Inlier Ratio*.

| | Inlier Ratio ↑ | | | | |
|---|---|---|---|---|---|
| Sequence | ICP [6] | EM-ICP [11] | HASTE [1] | EKLT [2] | **Ours** |
| Shapes Translation | 0.986 | 0.916 | 0.957 | 0.882 | 0.962 |
| Shapes Rotation | 0.962 | 0.955 | 0.950 | 0.968 | 0.950 |
| Shapes 6DOF | 0.696 | 0.755 | 0.325 | 0.852 | 0.946 |
| Boxes Translation | 0.937 | 0.937 | 0.963 | 0.945 | 0.980 |
| Boxes Rotation | 0.946 | 0.798 | 0.908 | 0.980 | 0.949 |
| EC Avg | 0.905 | 0.872 | 0.820 | 0.925 | 0.957 |
| Peanuts Light | 0.740 | 0.868 | 0.815 | 0.780 | 0.802 |
| Rocket Earth Light | 0.369 | 0.401 | 0.293 | 0.375 | 0.374 |
| Ziggy In The Arena | 0.421 | 0.884 | 0.609 | 0.469 | 0.927 |
| Peanuts Running | 0.502 | 0.578 | 0.531 | 0.700 | 0.750 |
| EDS Avg | 0.508 | 0.683 | 0.562 | 0.581 | 0.713 |

Table 6. The performance of our proposed and the baseline trackers on the EDS and EC dataset in terms of *Track Normalized Error*.

| Sequence | Track Normalized Error ↓ | | | | |
|---|---|---|---|---|---|
| | ICP [6] | EM-ICP [11] | HASTE [1] | EKLT [2] | **Ours** |
| Shapes Translation | 1.943 | 3.941 | 2.628 | 1.104 | 1.153 |
| Shapes Rotation | 1.870 | 2.614 | 2.536 | 1.723 | 1.981 |
| Shapes 6DOF | - | - | - | 1.833 | 1.702 |
| Boxes Translation | 2.289 | 2.613 | 2.109 | 1.227 | 1.166 |
| Boxes Rotation | 2.571 | 3.855 | 3.383 | 1.375 | 1.836 |
| EC Avg | 2.168 | 3.256 | 2.664 | 1.452 | 1.568 |
| Peanuts Light | 3.185 | 2.323 | 2.432 | 3.560 | 3.957 |
| Rocket Earth Light | - | 4.062 | - | 2.405 | 3.599 |
| Ziggy In The Arena | - | 3.407 | 2.672 | - | 2.673 |
| Peanuts Running | - | - | - | 3.812 | 3.444 |
| EDS Avg | 3.185 | 3.264 | 2.552 | 3.259 | 3.418 |

Table 7. The performance of the *reference model* when trained with different input event representations.

| Sequence | Expected FA ↑ | | | |
|---|---|---|---|---|
| | SBT-Max | SBT No Norm | SBT [10] | Voxel Grids [12] |
| Shapes Translation | 0.780 | 0.160 | 0.887 | 0.802 |
| Shapes Rotation | 0.747 | 0.057 | 0.823 | 0.799 |
| Shapes 6DOF | 0.881 | 0.006 | 0.882 | 0.882 |
| Boxes Translation | 0.849 | 0.160 | 0.831 | 0.769 |
| Boxes Rotation | 0.614 | 0.057 | 0.677 | 0.638 |
| EC Avg | 0.774 | 0.088 | 0.820 | 0.778 |
| Peanuts Light | 0.388 | 0.020 | 0.373 | 0.372 |
| Rocket Earth Light | 0.271 | 0.009 | 0.284 | 0.282 |
| Ziggy In The Arena | 0.686 | 0.040 | 0.708 | 0.694 |
| Peanuts Running | 0.059 | 0.024 | 0.073 | 0.150 |
| EDS Avg | 0.351 | 0.023 | 0.359 | 0.374 |

Table 8. The performance of the *reference model* when trained with different augmentation parameters.

| Sequence | Expected FA ↑ | | | |
|---|---|---|---|---|
| | R15 S10 T3 | R30 | T5 | No Aug |
| Shapes Translation | 0.691 | 0.861 | 0.720 | 0.723 |
| Shapes Rotation | 0.726 | 0.766 | 0.697 | 0.617 |
| Shapes 6DOF | 0.883 | 0.882 | 0.876 | 0.499 |
| Boxes Translation | 0.809 | 0.791 | 0.743 | 0.501 |
| Boxes Rotation | 0.616 | 0.703 | 0.448 | 0.337 |
| EC Avg | 0.745 | 0.801 | 0.697 | 0.535 |
| Peanuts Light | 0.361 | 0.384 | 0.337 | 0.311 |
| Rocket Earth Light | 0.284 | 0.275 | 0.274 | 0.094 |
| Ziggy In The Arena | 0.658 | 0.699 | 0.669 | 0.166 |
| Peanuts Running | 0.080 | 0.098 | 0.156 | 0.028 |
| EDS Avg | 0.346 | 0.364 | 0.359 | 0.150 |

## References


[1] Ignacio Alzugaray and Margarita Chli. HASTE: multi-Hypothesis Asynchronous Speeded-up Tracking of Events. *British Machine Vision Conference (BMVC). London, UK: Springer*, page 744, 2020. 1, 2, 4, 5

[2] 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. 1, 2, 4, 5

[3] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense Continuous-Time Optical Flow from Events and Frames, 2022. 1

[4] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conf.*, volume 15, pages 147–151, 1988. 1

[5] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. Event-aided Direct Sparse odometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[6] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pages 16–23, 2016. 4, 5

[7] Jacques Manderscheid, Amos Sironi, Nicolas Bourdis, Davide Migliore, and Vincent Lepetit. Speed invariant time surface for learning to detect corner points with event-based cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 2

[8] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. In *British Mach. Vis. Conf. (BMVC)*, 2017. 2

[9] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Research*, 36(2):142–149, 2017. 1

[10] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, and others. 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*, pages 10081–10090, 2019. 2, 3, 5

[11] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based feature tracking with probabilistic data association. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4465–4470, 2017. 4, 5

[12] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997, 2019. 3, 5