3D Feature Tracking via Event Camera Supplementary Material

Siqi Li[1] Zhikuan Zhou[1] Zhou Xue[2] Yipeng Li[3] Shaoyi Du[4] Yue Gao[1]*

[1]{BNRist, THUIBCS, School of Software}, Tsinghua University [2]Li Auto [3]Department of Automation, Tsinghua University

[4]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

{lsq19, zzk22}@mails.tsinghua.edu.cn, xuezhou08@gmail.com, dushaoyi@xjtu.edu.cn, {liep, gaoyue}@tsinghua.edu.cn

1. Supplementary Videos

In our supplementary material, we also provide videos of the 3D feature tracking results of our proposed method and the second-best method DeepEvT [3] + SDE [4] for a better comparison, as shown in the folder "**Comparison Results Videos**". The predicted feature trajectories are in blue, and the ground truth feature trajectories are in red. The comparison result videos are $10\times$ slower.

2. Method Comparisons

Our proposed method and dataset mainly focus on the 3D feature tracking task under high-speed scenarios, *i.e.*, tracking long-term feature trajectories with high-speed motions. Here we explain why we claim that there are no existing methods that could achieve this task. There are several approaches that may achieve similar motion estimation, but all suffer from fatal weaknesses.

Feature tracking methods. Existing feature tracking methods could predict long-term feature trajectories. With the aid of event cameras, event-based feature tracking methods could achieve high-speed feature tracking. However, as mentioned in the paper, existing event-based feature tracking methods could only estimate the 2D feature trajectories in the image plane.

Optical flow estimation methods. Optical flow estimation methods aim to predict the pixel-wise motion field between adjacent timestamps instead of long-term motion estimation. Therefore, fatal errors will occur when achieving long-term motion prediction, as shown in our experiments. On the other hand, similar to feature tracking, there are some existing event-based optical flow estimation methods that could predict the high-speed short-term motion field. However, there is also a lack of 3D event-based optical flow estimation methods.

Scene flow estimation methods. There are some existing scene flow estimation methods that could estimate 3D motion field between adjacent timestamps. However, these methods also could not achieve 3D feature tracking under high-speed scenarios. The main reasons lie in two folds. On the one hand, similar to optical flow estimation methods, these scene flow estimation methods also focus on short-term motion prediction and lack of long-term consistency. On the other hand, existing scene flow estimation methods are mainly based on videos or time-series point clouds. Therefore, as mentioned in the paper, these methods could not achieve motion estimation under high-speed scenarios limited by the frame rate of frame-based cameras and 3D vision sensors.

To sum up, existing methods either could not handle high-speed motion or could not achieve long-term consistent feature tracking. Therefore, to the best of our knowledge, we believe that there are no existing methods that could achieve satisfied 3D feature tracking under high-speed scenarios.

3. Dataset Processing

This section describes the system setup of our hybrid vision system and the processing details of our dataset. As mentioned in our paper, is it difficult to directly obtain high-speed 3D feature trajectories due to the limitation of existing 3D vision sensors. To tackle this challenge, we combine stereo DAVIS346 event cameras, the Optitrack motion capture system, and the FARO Quantum ScanArm to obtain feature-level high-speed 3D ground truth trajectories.

To achieve temporal synchronization, we use a sync cable to connect both DAVIS346 event cameras. Then, the timestamps of the event streams from both cameras could be automatically synchronized. Then, we use eSync 2 to output the trigger signal of the Optitrack (square wave, raising edge corresponding to the start exposure time). The trigger signal could be recorded by DAVIS346 through an-

*Corresponding author

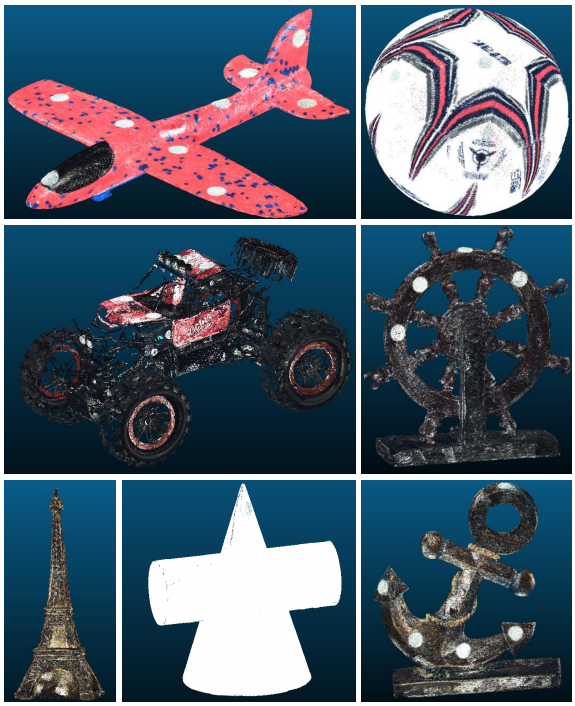


Figure 1. Some example object point clouds.

other sync cable. Then, the event stream timestamp could be aligned accordingly based on the Optitrack timestamp.

To achieve spatial calibration, we use a chessboard calibration board and attached several reflective markers to it. The calibration processes are as follows:

- Move the calibration board in 3D space and collect corresponding video and 3D coordinates using time-synchronized stereo event cameras and Optitrack.
- Perform camera calibration according to the chessboard to obtain the intrinsic and extrinsic parameters of the stereo event cameras.
- Perform image distortion rectification and stereo rectification for stereo event cameras.
- Obtain the 2D coordinates of the markers from the rectified images. Obtain the 3D coordinates at the same timestamp from Optitrack.
- Compute the rotation and the translation vectors from the world (Optitrack) coordinate system to the camera coordinate system by minimizing the reprojection error from 3D-2D marker point correspondences using the non-linear Levenberg-Marquardt minimization scheme [2].

In practice, for each calibration, we capture 50 seconds of stereo videos at about 40 FPS using DAVIS346 and the corresponding Optitrack spatial coordinates sequence at 250 FPS. We try to cover all positions at different depths in the camera's field of view. These data are used to calibrate our system. To ensure the data quality, we calibrate the system every day before we start recording .

After the temporal synchronization and spatial calibration are achieved, we use our system to collect our dataset. Specifically, we use multiple objects with attached markers to move in 3D space. Here we further provide the data processing details.

- Raw stereo event streams, reference frames at the initial moment, and the spatial coordinates sequence of the markers of each object are obtained from the stereo event cameras and Optitrack, respectively.
- Achieve distortion rectification to reference frames and event streams according to the calibration parameters. Achieve temporal alignment.
- Calculate the time-series 3D markers' coordinates sequence in the camera coordinate system according to the rotation and the translation vectors.
- We use FARO Quantum ScanArm to scan the high-precision object point cloud. The marker attached to the object is also scanned. In practice, each object point cloud contains over 2,000,000 points. Figure 1 shows some example object point clouds used in our dataset.
- Manually obtain the coordinates of all marker points in the object coordinate system. Calculate the 3D affine transform with a homogeneous scale at each timestamp based on the markers' coordinates in the object coordinate system and the markers' coordinates in the world coordinate system (from Optitrack at each timestamp).
- Generate time-series object point cloud sequence in camera coordinate system based on the 3D affine transform.
- Manually obtain the point index in the object point cloud of each feature. Obtain the time-series features' coordinates, *i.e.*, feature trajectories, in camera coordinate system based on point cloud sequence and feature indexes.

Thus, using our hybrid vision system and data processing pipeline, high-precision high-speed 3D feature trajectories could be obtained.

4. Method Details

4.1. Tracking Pipeline

Similar to the previous event-based feature tracking methods DeepEvT [3] and EKLT [1], the features to be tracked are provided in the image template patches at the initial moment t_0. In each subsequent timestamp, target features are tracked using corresponding event patches step by step. It should be noticed that at the initial moment t_0, the 2D coordinates $\mathbf{u}_{t_0} = (u_{t_0}, v_{t_0})$ of each target feature are provided. Therefore, the corresponding event patch could be obtained based on $\mathbf{u}_{t_0}$. Then, for each subsequent timestamp, the feature coordinates are predicted from the previous step. Specifically, for a feature located as $\mathbf{u}_{t_i} = (u_{t_i}, v_{t_i})$, the event patch P_{t_i} is calculated from the events $\mathcal{E}_i$ triggered in the $d \times d$ patch around $\mathbf{u}_{t_i}$ and within the time bin $[t_i, t_{i+1}]$,

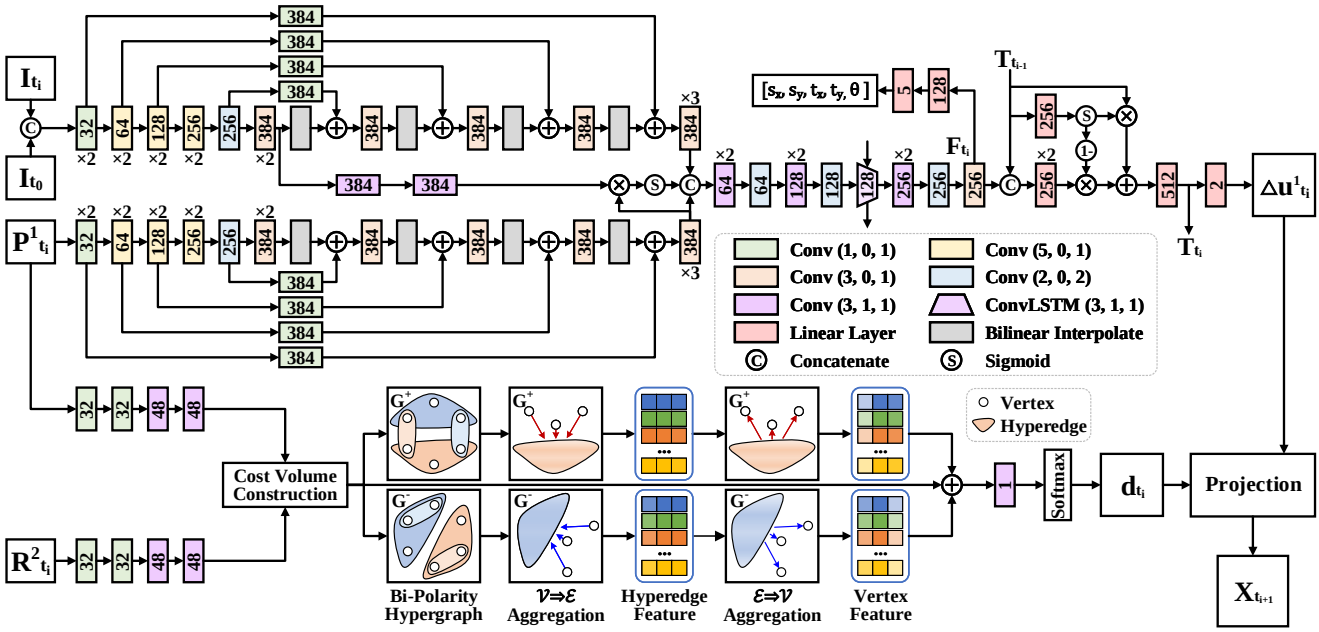


Figure 2. Detailed network architecture of our proposed method. The output channel of each layer is shown in the corresponding box.

formulated as:

$$\mathcal{E}_i = \{e_k | u_{t_i} - \frac{d-1}{2} \leq u_k \leq u_{t_i} + \frac{d-1}{2}, \\ v_{t_i} - \frac{d-1}{2} \leq v_k \leq v_{t_i} + \frac{d-1}{2}, \\ t_i \leq t_k \leq t_{i+1}\} \quad (1)$$

where $e_k = (t_k, u_k, v_k, p_k)$ is the k-th events captured by the event camera.

For the training stage, as mentioned in the paper, we predict the motion offsets in both camera planes, *i.e.*, $\Delta\mathbf{u}^1_{t_i}$ and $\Delta\mathbf{u}^2_{t_i}$, to calculate the stereo motion consistency constraint. To achieve this, we additionally leverage the coordinates of the target feature in camera 2, *i.e.*, $\mathbf{u}^2_{t_0}$, at the initial moment t_0 as input. Therefore, the motion offsets of the feature in camera 2, *i.e.*, $\Delta\mathbf{u}^2_{t_i}$, at each timestamp could be obtained using the offset estimation module. Therefore, the stereo motion consistency loss could be calculated and leveraged as the supervision of our model.

In contrast, in the inference stage, our proposed model only takes the feature coordinates in camera 1 at the initial moment as input. This is due to the fact that we hope the 3D trajectories of features can be predicted using as simple inputs as possible in real-world applications. Obviously, manually obtaining the coordinates of the target feature in both cameras at the initial moment is not conducive to the application. Therefore, our method could use the same prior knowledge as the 2D feature tracking method as input (*i.e.*, the feature coordinates in a single camera) to achieve 3D feature tracking.

4.2. Network Architecture

Figure 2 shows the detailed network architecture of our proposed method. The number in each box denotes the number of output channels of the corresponding layer. As shown in the figure, at each timestamp t_i our proposed method takes the initial feature template patch I_{t_0}, the deformed template patch I_{t_i}, the event patch $P^1_{t_i}$ from camera 1, and the event row patch $R^2_{t_i}$ from camera 2 as input to calculate the feature position at the next timestamp.

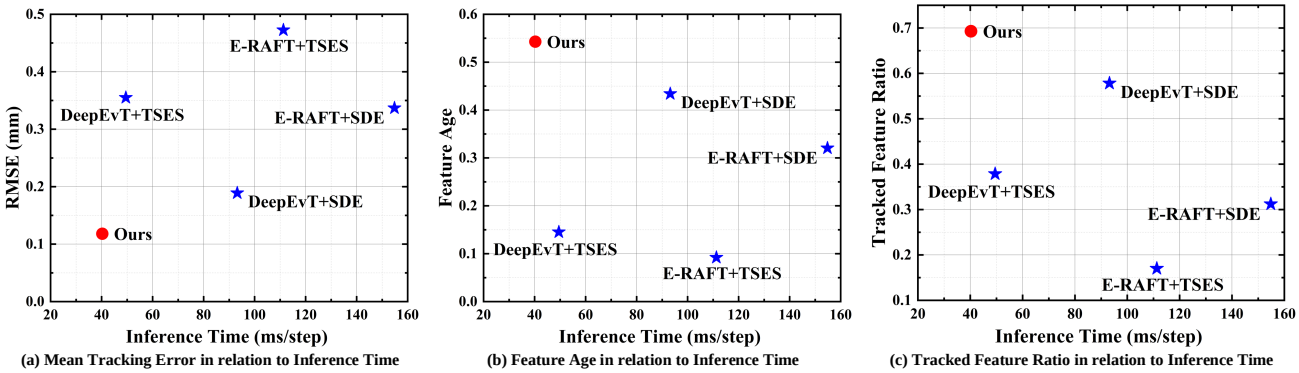
5. Experimental Details

5.1. Metric

We use Feature Age [3] (FR, higher is better), Tracked Feature Ratio (TFR, higher is better), and Root Mean Squared Error (RMSE, lower is better) as the metrics. Let $\mathbf{T}^{\text{pred}} = \left\{\mathbf{X}^{\text{pred}}_{t_i} = (x^{\text{pred}}_{t_i}, y^{\text{pred}}_{t_i}, z^{\text{pred}}_{t_i})\right\}_{i=1}^{N}$ denotes the predicted trajectory of a target feature, where N is the trajectory sequence length. The corresponding ground truth is denoted as $\mathbf{T}^{\text{gt}}$. Feature age is calculated as the ratio of the timestamp when the distance between $\mathbf{T}^{\text{pred}}$ and $\mathbf{T}^{\text{gt}}$ exceeds a certain threshold c for the first time to the sequence length:

$$\text{FA}_c = \frac{\arg\min_i \|\mathbf{X}^{\text{gt}}_{t_i} - \mathbf{X}^{\text{pred}}_{t_i}\|_2 > c}{N}. \quad (2)$$

Tracked feature ratio is calculated as the ratio of the total time that the spatial distance between $\mathbf{T}^{\text{pred}}$ and $\mathbf{T}^{\text{gt}}$ is less



(a) Mean Tracking Error in relation to Inference Time (b) Feature Age in relation to Inference Time (c) Tracked Feature Ratio in relation to Inference Time

Figure 3. Mean tracking error, feature age, and tracked feature ratio of each method in relation to inference time.

than a certain threshold c to the total sequence length:

$$\text{TFR}_c = \frac{\left|\left\{i \middle| \|\mathbf{X}_{t_i}^{\text{gt}} - \mathbf{X}_{t_i}^{\text{pred}}\|_2 < c\right\}\right|}{N}. \tag{3}$$

Different from feature age which only concerns the moment when $\mathbf{T}^{\text{pred}}$ deviates from $\mathbf{T}^{\text{gt}}$ for the first time, the tracked feature ratio calculates the total time that $\mathbf{T}^{\text{pred}}$ is close to $\mathbf{T}^{\text{gt}}$ over the complete sequence length. Our motivation for designing this metric is that we observe that in many cases the tracking methodology could correct the predicted trajectory even when the error at the initial moment is quite large.

Root mean squared error is calculated as:

$$\text{RMSE} = \frac{\sum_{i=1}^{N}\|\mathbf{X}_{t_i}^{\text{gt}} - \mathbf{X}_{t_i}^{\text{pred}}\|_2}{N}. \tag{4}$$

5.2. Computational Efficiency

Figure 3 shows the computation efficiency of our proposed method and other baseline methods. We show the (a) mean tracking error, (b) feature age, and (c) tracked feature ratio of each method in relation to their corresponding inference time, respectively. For the mean tracking error (Fig. 3 (a), lower is better), the bottom left corner represents the goal, and for the feature age (Fig. 3 (b), higher is better) and tracked feature ratio (Fig. 3 (c), higher is better), the top left corner represents the goal. From the figure, we could observe that our proposed method could significantly outperform other baseline methods.

5.3. Ablation Experiments Settings

In our ablation experiment, we test the performance of our proposed method add or remove the $\mathcal{L}^{\text{smc}}$, motion compensation (MC) module, and the bi-polarity hypergraph-based high-order correlation modeling (BiHCM) mechanism, respectively. In practice, the removal of $\mathcal{L}^{\text{smc}}$ is to remove $\mathcal{L}^{\text{smc}}$ from the total loss function Eq. (10), and use the remaining loss function as supervision to train our model. The removal of MC is to remove the motion compensation module and use the initial template patch I_{t_0} to replace I_{t_i} as the input of the offset estimation module. The removal of BiHCM is to remove the bi-polarity hypergraph construction and replace the hypergraph-based feature aggregation (Eq. (2)) with a Linear layer, *i.e.*, remove the hypergraph structure-guided feature optimization and retain $\mathbf{\Theta}$ in Eq. (2), which can show the effectiveness of BiHCM.

(a) 2D feature tracking results.

Method	FA_5 ↑	FA_{10} ↑	RMSE↓
E-RAFT	0.044	0.105	24.431
EKLT	0.047	0.112	23.100
DeepEvT	<u>0.510</u>	<u>0.745</u>	<u>6.833</u>
Ours	**0.673**	**0.844**	**4.888**

(b) Stereo matching results.

Method	MAE↓	RMSE↓
TSES	31.296	0.437
SDE	<u>6.001</u>	<u>0.140</u>
Ours	**5.110**	**0.113**

Table 1. 2D quantitative comparison.

5.4. 2D Comparison

Table 1a shows the 2D feature tracking results of our method and other existing methods. RMSE and Feature Age with $c = 5, 10$ pixels are used as the metrics. Table 1b shows the stereo matching results of our method and other existing methods. The Mean Average Error (MAE) of disparity and the RMSE of back-projection results according to predicted disparity and GT 2D feature coordinates are selected as metrics. From the table, we could observe that our method could outperform DeepEvT (28.5%) and SDE (14.8%) individually. Two-stage baselines fail mainly because of cumulative errors due to lack of consistent constraint. Using our joint framework and $\mathcal{L}^{\text{smc}}$ could achieve more remarkable improvements (37.5%).

References

[1] 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. 2

[2] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. Methods for Non-Linear Least Squares Problems. 2004. 2

[3] 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. 1, 2, 3

[4] Yeongwoo Nam, Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. Stereo Depth From Events Cameras: Concentrate and Focus on the Future. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6114–6123, 2022. 1