3D Feature Tracking via Event Camera

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

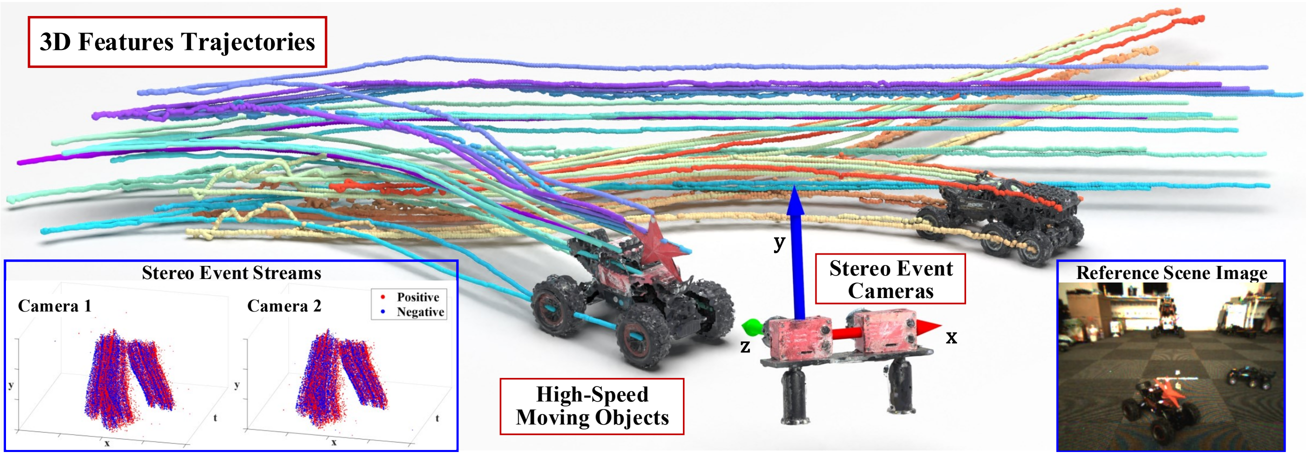


Figure 1. We present the first high-speed 3D feature tracking method via stereo event cameras and the corresponding high-speed 3D feature tracking dataset. Our proposed method takes high temporal resolution event streams captured from stereo event cameras as input, and could predict the long-term feature motion trajectories of multiple high-speed moving objects within the scene at a rate of 250 FPS.

Abstract

*This paper presents the first 3D feature tracking method with the corresponding dataset. Our proposed method takes event streams from stereo event cameras as input to predict 3D trajectories of the target features with high-speed motion. To achieve this, our method leverages a joint framework to predict the 2D feature motion offsets and the 3D feature spatial position simultaneously. A motion compensation module is leveraged to overcome the feature deformation. A patch matching module based on bipolarity hypergraph modeling is proposed to robustly estimate the feature spatial position. Meanwhile, we collect the first 3D feature tracking dataset with high-speed moving objects and ground truth 3D feature trajectories at 250 FPS, named **E-3DTrack**, which can be used as the first high-speed 3D feature tracking benchmark. Our code and dataset could be found at:* `https://github.com/lisiqi19971013/E-3DTrack`.

1. Introduction

Feature tracking aims to predict the long-term trajectories of target features, which is fundamental in many computer vision tasks, *e.g.*, object tracking [36, 37], 3D reconstruction [8, 17], and SLAM [20, 41]. Frame-based feature tracking methods [5, 24, 25, 33, 35] have been extensively investigated in the past decades. However, all existing methods focus on tracking 2D feature trajectories in the image plane.

In real-world scenarios, objects are moving in 3D space, *e.g.*, cars are racing on the road from near to far. The tracking of features with high-speed 3D motion becomes essential. Consequently, there is an imperative need to investigate 3D feature tracking methods capable of predicting feature trajectories for objects undergoing high-speed 3D motion. Such methods hold significant promise for various downstream applications, *e.g.*, VR, AR, and autonomous driving. To the best of our knowledge, existing literature lacks established high-speed 3D feature tracking methodologies.

For the 3D feature tracking of high-speed moving objects, the main challenges lie in three folds. (1) With the limited frame rate of traditional frame-based cameras, the motion of high-speed moving objects may not be consistently captured due to the blind time between consecutive frames. Therefore, how to continually record valid motion information of high-speed moving objects is the first challenge. (2) The second challenge lies in establishing the correlation between the 3D position of the feature and the 2D visual data acquired by cameras to generate a continuous and smooth 3D feature trajectory. (3) To the best of our knowledge, there are currently no existing high-speed 3D feature tracking datasets. This is primarily due to the difficulty in capturing ground truth 3D feature trajectories of high-speed moving objects, which is constrained by the insufficient capture frequency of existing 3D vision sensors. Thus, the lack of high-speed 3D feature tracking dataset is the third challenge, which is also a principal impediment to the advancement of research within this domain.

To overcome the motion capture challenge, we use event cameras to record motion dynamics of high-speed moving objects. Event cameras [7, 32] are bio-inspired vision sensors that asynchronously respond to pixel-wise brightness changes. Specifically, when the logarithmic change of the brightness at a pixel exceeds a certain threshold, *i.e.*, $|\Delta_t \log I(x, y, t)| > C$, where $I(x, y, t)$ is the brightness at pixel (x, y) and timestamp t, an **event** will be triggered, denoted as $e = (t, x, y, p)$, where $p \in \{1, -1\}$ is the polarity. The output **event stream** of event cameras, formed by events triggered by all pixels, showcases their remarkably high temporal resolution (in the order of microseconds) and broad dynamic range (up to 140 dB) [13]. These unique features of event cameras render them promising tools for achieving 3D feature tracking in the context of high-speed moving objects.

To address the aforementioned technical challenge, we propose a high-speed 3D feature tracking method based on stereo event cameras, predicting the long-term 3D trajectories of target features from stereo event streams and template patches. To achieve 3D feature tracking, our proposed method leverages a joint framework to predict the 2D feature motion offsets and the feature spatial position at each timestamp simultaneously. A motion compensation module is leveraged to adapt to the feature deformation, and a patch matching module based on bi-polarity hypergraph modeling is proposed to accurately estimate the feature spatial position. In addition, we introduce a stereo motion consistency mechanism that establishes the constraint between the feature motion offsets and the spatial position to achieve smooth 3D trajectory estimation.

To address the data challenge, we establish a hybrid vision system and curate the first real-world event-based 3D feature tracking dataset, named **E-3DTrack**. Our dataset includes multiple objects demonstrating high-speed motion in the scene, with stereo event cameras capturing high temporal resolution event streams, as shown in Fig. 1. To obtain the ground truth of the 3D feature trajectories, we utilize the Optitrack motion capture system to record the motion trajectory of each moving object. This information is then integrated with the high-precision object point cloud scanned by FARO Quantum ScanArm, resulting in the generation of the ground truth 3D trajectories of each feature at a rate of 250 FPS. To the best of our knowledge, our dataset is the first event-based feature tracking dataset containing high-speed moving objects and providing 3D ground truth feature trajectories.

Our contributions could be summarized as follows:

- We propose the first high-speed 3D feature tracking method based on stereo event cameras, which could track the 3D trajectories of features with high-speed motion.
- We achieve satisfactory 3D feature tracking performance through a motion compensation module for addressing feature deformation, a patch matching module based on bi-polarity hypergraph modeling for accurate estimation of 3D feature positions, and a stereo motion consistency mechanism to establish constraints between feature motion offsets and 3D position.
- We collect the first real-world 3D feature tracking dataset containing multiple high-speed moving objects, named **E-3DTrack**. Our dataset contains stereo event streams and 250 FPS ground truth 3D feature trajectories, which could be used as the 3D feature tracking benchmark.

2. Related Work

Trajectory Prediction via Event Camera. Event-based feature tracking methods have been developed rapidly within the last decade. Earlier works [18, 38] treat the events as a point set and used ICP [6] to estimate feature motion trajectories. Then, EKLT [10] is proposed to obtain feature patch from the reference frame as template, and use the event stream to track the template and predict the trajectory. Meanwhile, some event-by-event trackers [2, 3] are proposed to exploit the asynchronicity of event camera, *e.g.*, eCDT [16] employs a clustering method to cluster adjacent events, and uses cluster descriptors to find continual feature tracks. Recently, DeepEvT [26] is proposed as the first data-driven event-based feature tracking method, which achieves state-of-the-art 2D feature tracking performance.

An alternative approach for trajectory prediction is optical flow estimation, wherein the pixel-level motion field is predicted using the input event stream. Compared with feature tracking, these methods [1, 4, 12, 29, 30] focus more on estimating the motion field between adjacent moments and lack modeling of long-term trajectory consistency.

However, all these existing trajectory prediction methods could only predict 2D feature trajectories in the image plane while the real objects are moving in 3D space, *i.e.*, the predicted feature motion trajectories are information-deficient.

Event-based 3D Position Estimation. As the 2D feature trajectories can be predicted, a simple and straightforward solution is to use a monocular or stereo depth estimation method to predict the depth of the feature and calculate the 3D position. In recent years, several event-based

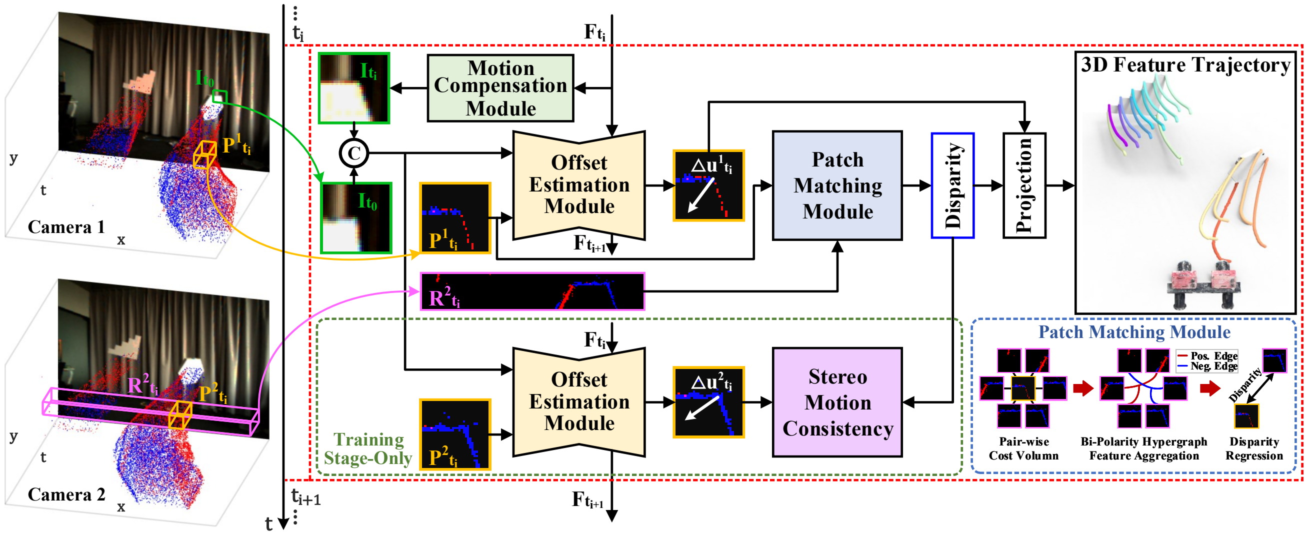


Figure 2. Our proposed method takes stereo event streams as input to predict the 3D trajectory of the target feature provided in the initial template patch I_{t_0}. For a subsequent timestamp t_i, the deformed template patch I_{t_i} is predicted using the motion compensation module. Then, I_{t_i}, I_{t_0}, and the events P_{t_i} triggered within the spatiotemporal neighboring patch of the predicted feature position $\mathbf{u}_{t_i}$ are forwarded into the offset estimation module to estimate the feature motion offsets $\Delta\mathbf{u}_{t_i}$. Meanwhile, a patch matching module based on bi-polarity hypergraph modeling is leveraged to predict the disparity. Finally, a projection operation is performed to update the 3D trajectory.

monocular [11, 14, 40] or stereo [28, 39] depth estimation methods are proposed, which could estimate the depth map from the input single-view or multi-view event streams.

However, we will show that the simple combination of these two types of methods could not achieve satisfying long-term 3D feature trajectories prediction performance in Sec. 5.2. Therefore, high-speed 3D feature tracking is still a challenging open problem.

3. Method

In this section, we commence with an overview of the pipeline in Sec. 3.1, subsequently delving into the detailed architecture in Sec. 3.2, and conclude by outlining the supervision of our method in Sec. 3.3.

3.1. Overview

As shown in Fig. 2, our proposed method takes stereo event streams as input to predict 3D feature trajectories in camera 1 coordinate system. The target features are contained in gray-scale template patches at the initial moment. This is the common setting for event-based feature tracking, *e.g.*, EKLT [10] and DeepEvT [26]. Our method leverages a joint framework to predict features' 2D motion offsets and the 3D spatial positions simultaneously at each timestamp, and further obtain the 3D trajectories through projection.

Specifically, let $\mathcal{E}^j = \left\{ e_k^j = (t_k^j, u_k^j, v_k^j, p_k^j) \right\}$ denote the event stream captured by an event camera, where $j = 1, 2$ denotes camera 1 and camera 2, respectively, e_k^j is the k-th event captured by camera j. The feature to be tracked is provided in a $d \times d$ template patch I_0 captured by camera 1 at initial moment t_0. Then, the feature trajectory is predicted step-by-step. For a subsequent timestamp t_i, we calculate the 2D feature coordinates $\mathbf{u}^1_{t_i} = (u_{t_i}, v_{t_i})$ projected in camera 1 based on the predicted 3D feature position $\mathbf{X}_{t_i} = (x_{t_i}, y_{t_i}, z_{t_i})$ at the previous step. To calculate the feature trajectory, the events $\mathcal{E}^1_i$ triggered in the $d \times d$ patch around $\mathbf{u}^1_{t_i}$ and within the time bin $[t_i, t_{i+1}]$ are leveraged to provide feature motion information. Then, $\mathcal{E}^1_i$ is converted into grid-based event patch $P^1_{t_i}$ using the event representation method proposed in [26].

As shown in Fig. 2, the 3D movement of the object may cause deformation of the feature template patch. To tackle this challenge, we leverage a motion compensation module to predict the deformed template patch $\tilde{I}_{t_i}$ at timestamp t_i. Then, $\tilde{I}_{t_i}$ and I_{t_0} are concatenated and forwarded into the offset estimation module together with $P^1_{t_i}$ to predict the 2D feature motion offset $\Delta\mathbf{u}^1_{t_i}$. To further estimate the 3D position of the target feature, we use the events triggered within the same d rows as $P^1_{t_i}$ from $\mathcal{E}^2$, *i.e.*, $\mathcal{E}^2_i = \{e^2_k | u_{t_i} - \frac{d-1}{2} \leq u^2_k \leq u_{t_i} + \frac{d-1}{2}, t_i \leq t^2_k \leq t_{i+1}\}$, to generate event row patch $R^2_{t_i}$ using the same event representation method. Then, the disparity $d_{t_{i+1}}$ could be predicted from $P^1_{t_i}$ and $R^2_{t_i}$ using our proposed patch matching module based on bi-polarity hypergraph modeling. In addition, inspired by the fact that the 2D motion offsets in both camera planes have a constraint with the disparity change, we use the event patch $P^2_{t_i}$ of camera 2 to compute the offset $\Delta\mathbf{u}^2_{t_i}$ in the training stage, and propose a stereo motion consistency mechanism to enhance the trajectory prediction. Finally, the 3D feature position $\mathbf{X}_{t_{i+1}}$ at t_{i+1} is obtained by projection according to $\Delta\mathbf{u}^1_{t_i}$ and $d_{t_{i+1}}$. In practice, the patch size is $d = 31$, and the length of the time bin is set to 4 ms, *i.e.*, $t_{i+1} - t_i = 4$ ms. Thus, our proposed method could track the long-term 3D feature trajectories at 250 FPS.

3.2. Model Architecture

Offset Estimation Module. As mentioned above, at timestamp t_i, we use an offset estimation module to predict the feature motion offsets projected in the camera plane, which takes $\tilde{I}_{t_i}$, I_{t_0}, and P_{t_i} as input. Inspired by the great success of DeepEvT [26], we use a similar two-branch Feature Pyramid Network (FPN) [19] to extract multi-modal features from event patch P_{t_i} and the template patches I_{t_i} and I_{t_0}, respectively. The FPN contains 4 down-sample layers and 4 up-sample layers. Then, the bottleneck feature of FPN is leveraged to calculate the correlation map between the event patch and the feature template patch. The correlation map is further concatenated with the multi-modal feature and forwarded into a joint encoder with 4 down-sample layers and a ConvLSTM [34] layer to obtain fused feature F_{t_i}. Then, we use a linear layer to compute the weights of $F_{t_{i-1}}$ and F_{t_i} and explicitly fuse the temporal information. Finally, a linear layer is leveraged to generate predicted feature motion offsets. Detailed network architecture is provided in the supplementary material. Using the offset estimation module, the feature motion offset $\Delta\mathbf{u}_{t_i}$ projected in the camera plane could be estimated.

Motion Compensation Module. As shown in Fig. 2, high-speed 3D moving objects may have depth change and rotation, which may cause feature shape deformation. Therefore, tracking with the initial template patch may lead to fatal errors or even incorrectly tracking other features. To tackle this problem, we leverage a motion compensation module to correct the template patch at each moment. Specifically, the feature template patch may have scaling, rotation, and shear changes. It should be noted that translation is not considered since the feature motion offset is already predicted. At the timestamp t_i, the fused temporal feature $F_{t_{i-1}}$ is leveraged as input to predict the scale factors s_x, s_y, rotation angle θ, and shear factors t_x, t_y using 2 linear layers. Then, the affine transform is performed according to the predicted transform factors:

$$\tilde{I}_{t_i}(u,v) = \begin{bmatrix} \beta s_x, \alpha s_y \\ -\alpha s_x, \beta s_y \end{bmatrix} \begin{bmatrix} 1, a \\ -b, 1+ab \end{bmatrix} I_{t_0}(u,v), \qquad (1)$$

where $\alpha = \sin\theta$, $\beta = \cos\theta$, $a = \tan t_x$, and $b = \tan t_y$. Using the motion compensation module, the corrected template patch $\tilde{I}_{t_i}$ at each timestamp could be obtained.

Patch Matching Module. To further estimate the 3D position of the target feature, we propose a patch matching module based on bi-polarity hypergraph modeling to obtain the spatial position of the feature by predicting the disparity. Different from traditional stereo matching, for the 3D feature tracking task, the target feature is contained in the local event patch $P^1_{t_i}$. Therefore, the disparity could only be predicted from the local patch instead of global information. Under such condition, mismatching will occur since the target scene may contain multiple similar features distributed in space and $P^1_{t_i}$ only contains local information. Therefore, we propose a bi-polarity hypergraph-based high-order correlation modeling mechanism to eliminate mismatching.

As mentioned in Sec. 3.1, for each timestamp t_i, we use the event patch $P^1_{t_i}$ around $\mathbf{u}^1_{t_i}$ and the corresponding event row patch $R^2_{t_i}$ from camera 2 to achieve patch matching. Specifically, we use 4 convolutional layers to extract features $\mathbf{M}^1_{t_i} \in \mathbb{R}^{d\times d\times c}$ and $\mathbf{M}^2_{t_i} \in \mathbb{R}^{d\times W\times c}$ from $P^1_{t_i}$ and $R^2_{t_i}$, respectively, where c is the feature channel and W is the image width, *i.e.*, the number of candidate matching position. We further calculate the cost volume $\mathbf{C}_{t_i} \in \mathbb{R}^{W\times c}$ composed of the feature similarity between $\mathbf{M}^1_{t_i}$ and $\mathbf{M}^2_{t_i}$ at each matching position, which represents the pair-wise similarity between $P^1_{t_i}$ and the sub-patch of $R^2_{t_i}$ at each matching position. Then, the W matching positions are used as vertices to construct bi-polarity hypergraphs. Compared to the pair-wise correlation contained in the cost volume, each hyperedge of a hypergraph could connect multiple vertices, *i.e.*, high-order correlations among multiple vertices could be constructed. In practice, we use the Euclidean distance of the vertex feature as metric and calculate the k nearest neighbors of each vertex. For each vertex, we use a hyperedge to connect the vertices in its k neighbor vertices with spatial distance smaller than a certain threshold δ. Therefore, a positive hypergraph $\mathcal{G}^+$ with the adjacency matrix $\mathbf{H}^+$ could be constructed. Besides, for each vertex, vertices with spatial distance larger than δ in its k neighbor vertices are connected by another hyperedge. Thus, a negative hypergraph $\mathcal{G}^-$ with the adjacency matrix $\mathbf{H}^-$ could be constructed. Each hyperedge of $\mathcal{G}^+$ connects matching patches that are semantic similar and spatially close to $P^1_{t_i}$. These connections are expected to be enhanced. In contrast, each hyperedge of $\mathcal{G}^-$ connects matching patches that are semantic similar but spatially distant from $P^1_{t_i}$, which are interference and needs to be suppressed. Then, inspired by [9], we propose a feature aggregation method based on bi-polarity hypergraphs:

$$\begin{aligned}\hat{\mathbf{C}}_{t_i} = \mathbf{C}_{t_i} &+ \sigma\left(\left(\mathbf{D}^+_v\right)^{-1}\mathbf{H}^+\left(\mathbf{D}^+_e\right)^{-1}\left(\mathbf{H}^+\right)^\top\mathbf{C}_{t_i}\boldsymbol{\Theta}^+\right) \\ &- \sigma\left(\left(\mathbf{D}^-_v\right)^{-1}\mathbf{H}^-\left(\mathbf{D}^-_e\right)^{-1}\left(\mathbf{H}^-\right)^\top\mathbf{C}_{t_i}\boldsymbol{\Theta}^-\right),\end{aligned} \qquad (2)$$

where $\mathbf{D}^*_e$ and $\mathbf{D}^*_v$ are the diagonal matrices of hyperedge degree and vertex degree, respectively. $\boldsymbol{\Theta}^*$ is the learnable parameter, and $\sigma(\cdot)$ is the non-linear activation function. Using Eq. (2), features are aggregated to enhance vertices with similar features and spatial close and suppress vertices with similar features but spatially distant. Finally, $\hat{\mathbf{C}}_{t_i}$ is forwarded into a 1D convolutional layer with the kernel size of

3 to regress the matching result. Using the patch matching module, the disparity d_{t_i} of the feature is predicted.

Projection. After the feature motion offsets $\Delta\mathbf{u}_{t_i}$ and the disparity d_{t_i} are predicted, the 3D feature coordinates $\mathbf{X}_{t_{i+1}}$ at t_{i+1} could be computed using projection.

3.3. Supervision and Loss Functions

Stereo Motion Consistency. For objects moving in 3D space captured by stereo cameras, the 2D motion offsets are strongly constrained with the disparity. Meanwhile, our offset estimation module is also deeply coupled with the patch matching module. Therefore, inspired by [21], we leverage a stereo motion consistency constraint to reinforce this correlation. Consider a point $\mathbf{X} = (x, y, z)$, it's 2D coordinates in the camera plane could be calculated by $\mathbf{u} = (u, v) = \frac{f}{s}\frac{(x,y)}{z}$, where f is the camera focal length and s the coordinate convert factor. For calibrated stereo cameras with the baseline distance of b, the disparity of $\mathbf{X}$ is $d = \frac{f}{s}\frac{b}{z}$. By taking the time derivative, we could obtain that $\frac{\Delta d}{\Delta t} = -\frac{f}{s}\frac{b}{z^2}\frac{\Delta z}{\Delta t}$. Therefore, we have:

$$d_{t_i} - d_{t_{i-1}} = \Delta d = -\frac{f}{s}\frac{b}{z_{t_i}^2}(z_{t_i} - z_{t_{i-1}}). \tag{3}$$

For the 2D motion offsets, we could similarly obtain that $\frac{\Delta\mathbf{u}}{\Delta t} = \frac{f}{zs}(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t}) - \frac{f}{z^2 s}\frac{\Delta z}{\Delta t}(x, y)$, *i.e.*, we have $\Delta\mathbf{u} = (\Delta u, \Delta v) = \frac{f}{zs}(\Delta x, \Delta y) - \frac{f\Delta z}{z^2 s}(x, y)$. In practice, suppose the coordinates of a feature in camera 1 at timestamp t_i is $\mathbf{X}_{t_i}^1 = (x_{t_i}, y_{t_i}, z_{t_i})$, then the coordinates in camera 2 is $\mathbf{X}_{t_i}^2 = (x_{t_i} - b, y_{t_i}, z_{t_i})$. Therefore, we have:

$$\begin{aligned} &\Delta u_{t_i}^1 - \Delta u_{t_i}^2 = \frac{f}{s}\frac{b}{z_{t_i}^2}\Delta z_{t_i} = -\frac{f}{s}\frac{b}{z_{t_i}^2}(z_{t_i} - z_{t_{i-1}}) \\ &\Delta v_{t_i}^1 - \Delta v_{t_i}^2 = 0 \end{aligned} . \tag{4}$$

Therefore, we could obtain the stereo motion constraint $\Delta u_{t_i}^1 - \Delta u_{t_i}^2 = d_{t_i} - d_{t_{i-1}}$ from Eq. (3) and Eq. (4).

According to the stereo motion constraint, we introduce the stereo motion consistency loss:

$$\mathcal{L}_i^{\text{smc}} = \mathcal{L}_1(\Delta u_{t_i}^1 - \Delta u_{t_i}^2, d_{t_i} - d_{t_{i-1}}) + \mathcal{L}_1(\Delta v_{t_i}^1, \Delta v_{t_i}^2), \tag{5}$$

where $\mathcal{L}_1(\cdot, \cdot)$ is the Manhattan Distance.

Loss Functions. Since our proposed method could predict the 3D feature coordinate $\mathbf{X}_{t_i}$ at each timestamp, we use the Manhattan Distance between the predicted trajectories and ground truth trajectories as supervision:

$$\mathcal{L}_i^{\text{traj}} = \mathcal{L}_1(\mathbf{X}_{t_i}, \mathbf{X}_{t_i}^{\text{gt}}). \tag{6}$$

Since both the offset estimation module and the patch matching module severely affect the 3D trajectory prediction accuracy, we compute the ground truth 2D feature offsets $\mathbf{u}_{t_i}^{1^{\text{gt}}}$ and disparity $d_{t_i}^{\text{gt}}$ at each timestamp based on ground truth 3D trajectory through projection and use them as supervision. In practice, the offset estimation is supervised with the loss function:

$$\mathcal{L}_i^{\text{off}} = \mathcal{L}_1(\Delta\mathbf{u}_{t_i}^1, \Delta\mathbf{u}_{t_i}^{1^{\text{gt}}}). \tag{7}$$

Table 1. Comparison of our E-3DTrack dataset with other existing event-based feature tracking datasets.

Dataset	Dim.	Motion	Scenario	GT Freq.
EC [27]	2D	Homo.	Static	200
EDS [15]	2D	Homo.	Static	150
E-3DTrack	3D	Non-homo.	Dynamic	250

The disparity prediction is supervised with:

$$\mathcal{L}_i^{\text{disp}} = \mathcal{L}_1(d_{t_i}, d_{t_i}^{\text{gt}}). \tag{8}$$

Finally, our model is trained end-to-end with the supervision of the following total loss function:

$$\mathcal{L} = \sum_{i=1}^{N}(\mathcal{L}_i^{\text{traj}} + \mathcal{L}_i^{\text{off}} + \mathcal{L}_i^{\text{disp}} + \alpha\mathcal{L}_i^{\text{smc}}), \tag{9}$$

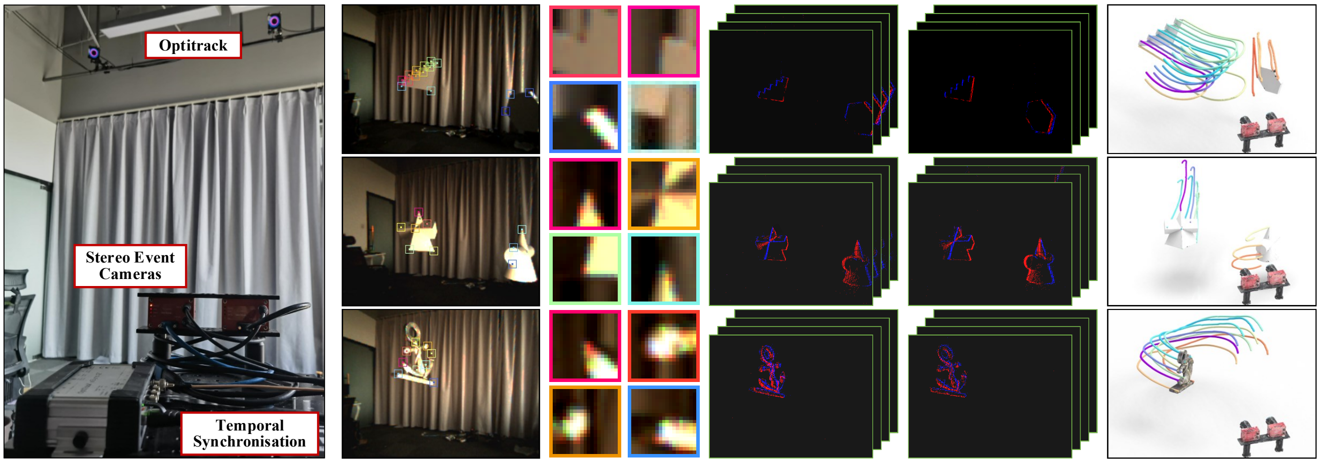
where α is a hyper-parameter and N is the sequence length.

4. 3D Feature Tracking Dataset: E-3DTrack

In addressing the deficiency of high-speed 3D feature tracking datasets, we establish a hybrid vision system containing stereo event cameras and Optitrack, as shown in Fig. 3 (a), and curate the first event-based 3D feature tracking dataset, named **E-3DTrack**. Compared to existing event-based feature tracking datasets that contain only static scenes and 2D trajectories, our dataset is the first to contain high-speed moving objects and ground truth 3D feature trajectories.

Limited by the capturing frequency of 3D vision sensors (*e.g.*, <30 FPS for LiDAR), it is difficult to accurately record the 3D feature trajectories of high-speed moving objects at a high frame rate. To tackle this problem, we use the motion capture system, *i.e.*, Optitrack, to record the trajectory of each object attached with fixed markers. To explicitly obtain feature-level 3D trajectories, we use a scanner, *i.e.*, FARO Quantum ScanArm, to capture the high precision point cloud of each object. Then, the 3D affine transform, incorporating a homogeneous scale, is calculated from the object coordinate system to the Optitrack coordinate system based on the markers' coordinates. This leads to the acquisition of the time-series point cloud sequence of the moving objects under the Optitrack coordinate system. Finally, the feature trajectories can be derived from the time-series point cloud sequence based on the feature point index. Hence, our dataset comprises ground truth 3D feature trajectories of high-speed moving objects at 250 FPS, surpassing the capturing frequency of most existing 3D vision sensors.

Using our hybrid vision system, we captured 40 high-speed motion scenarios containing a total of 1300 sequences. We randomly select 10 scenarios as the test set, and the remaining 30 scenarios are selected as the training set. Note that due to the cross-scene division, the scene in the test set are unseen in the training set. More details of our dataset are provided in the supplementary material.



(a) Our hybrid vision system. (b) Samples of our dataset. From left to right: reference frame, feature patch, stereo event streams, and ground truth 3D feature trajectory.

Figure 3. (a) Our hybrid vision system. (b) Samples of our E-3DTrack dataset. The first column is the reference frame at the initial moment, and the features to be tracked are marked in each frame. Some feature template patches are zoomed in for display in the second column. The stereo event streams and the ground truth 3D feature trajectories are shown in the last three columns, respectively.

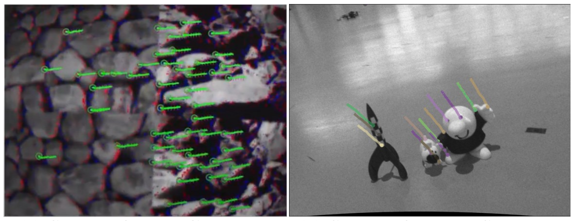
Table 1 shows the comparison of our E-3DTrack dataset with other existing event-based feature tracking datasets, including Event Camera dataset (EC) [27] and Event-aided Direct Sparse Odometry (EDS) dataset [15]. The main advantages of our dataset are in the following three aspects.

- **3D trajectory.** Our dataset is the first feature tracking dataset containing ground truth 3D trajectories, enabling the feature motion trajectory estimation in 3D space.
- **Non-homogeneous motion.** Our dataset is the first event-based feature tracking dataset containing high-speed moving objects. Existing EC and EDS datasets mainly contain stationary scenarios. Thus, feature motions are caused by the camera movement. Since there are no moving objects in the scene, the motions of all features are almost homogeneous, as shown in Fig. 4. In contrast, the feature motions in our dataset are non-homogeneous, which is more conducive to applications.
- **Accurate ground truth.** Our dataset contains ground truth 3D feature trajectories captured from Optitrack. In contrast, since the DAVIS346 event camera could record event streams and 25 FPS video simultaneously, the ground truth 2D trajectories in EC and EDS datasets are obtained using frame-based feature tracking method KLT [25], or further triangulating KLT tracks using camera poses and reprojecting them to the frames. Thus, our dataset contains more accurate ground truth trajectories.

Figure 3 (b) shows some samples of our E-3DTrack dataset. We visualize the reference frames, feature template patches, stereo event streams, and the ground truth 3D feature trajectories of each sample.

5. Experiments

In this section, we first introduce the experimental settings. Then, we analyze the quantitative and qualitative comparisons, respectively. Finally, we conduct ablation studies to demonstrate the effectiveness of each proposed module.



(a) Sample from EC Dataset (b) Sample from EDS Dataset

Figure 4. Examples from existing EC [27] and EDS [15] dataset.

5.1. Experimental Settings

Comparison Methods. Since there are no existing high-speed 3D feature tracking methods, we use existing event-based trajectory prediction methods to obtain 2D feature trajectories, and use stereo depth estimation methods to further obtain the 3D feature trajectory. Specifically, we combine the event-based optical flow estimation method E-RAFT [12], event-based feature tracking methods EKLT [10] and DeepEvT [26] with event-based stereo depth estimation methods TSES [39] and SDE [28], respectively, as our baseline comparison methods.

Metrics. To evaluate our proposed method and other comparison methods, we use the Tracked Feature Ratio (TFR, higher is better), Feature Age [26] (FR, higher is better), and the Root Mean Squared Error (RMSE, lower is better) as the metrics. TFR is calculated as the ratio of the time that the spatial distance between the predicted 3D trajectory and the ground truth 3D trajectory is less than a certain threshold c to the total sequence time. See detailed definition in the supplementary material.

Implementation Details. Our method is implemented based on PyTorch [31]. Our model is trained end-to-end

Table 2. Quantitative results on our E-3DTrack dataset. Feature age (FA), tracked feature ratio (TFR), and root mean square error (RMSE) are selected as the metrics. Bold numbers represent the best scores, and underlined numbers represent the second-best scores.

Method	$\text{FA}_{(0.1\text{m})} \uparrow$	$\text{FA}_{(0.15\text{m})} \uparrow$	$\text{FA}_{(0.2\text{m})} \uparrow$	$\text{TFR}_{(0.1\text{m})} \uparrow$	$\text{TFR}_{(0.15\text{m})} \uparrow$	$\text{TFR}_{(0.2\text{m})} \uparrow$	RMSE $\downarrow$
E-RAFT [12] + TSES [39]	0.0409	0.0664	0.092	0.1701	0.2667	0.3439	0.4726
E-RAFT [12] + SDE [28]	0.1385	0.2399	0.3204	0.3121	0.4726	0.5806	0.3368
EKLT [10] + TSES [39]	0.0232	0.0429	0.0628	0.1180	0.1961	0.2685	0.4806
EKLT [10] + SDE [28]	0.1026	0.1856	0.2584	0.2421	0.3738	0.4700	0.4034
DeepEvT [26] + TSES [39]	0.0713	0.1117	0.1452	0.3786	0.4991	0.5818	0.3549
DeepEvT [26] + SDE [28]	<u>0.2314</u>	<u>0.3462</u>	<u>0.4339</u>	<u>0.5782</u>	<u>0.7060</u>	<u>0.7765</u>	<u>0.1889</u>
E-3DTrack (Ours)	**0.2601**	**0.4179**	**0.5428**	**0.6928**	**0.8164**	**0.8772**	**0.1181**

Table 3. Comparison of inference time on E-3DTrack dataset.

Method	E-RAFT + SDE	DeepEvT + SDE	Ours
Time (ms/step)	154.83	<u>93.22</u>	**40.30**

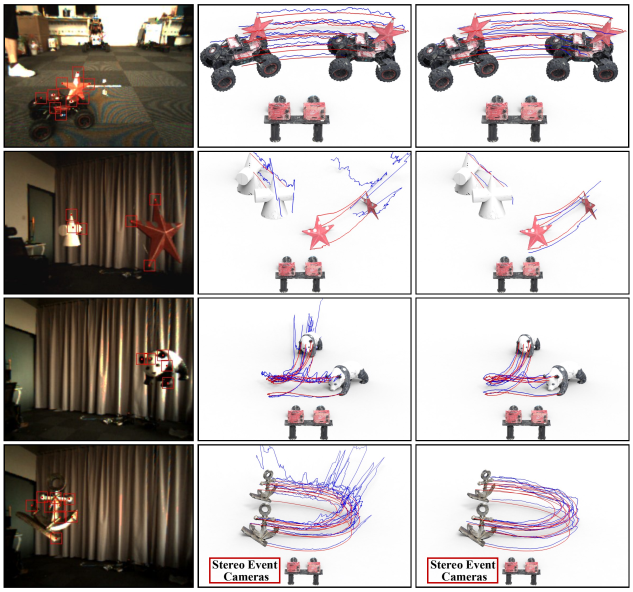
for 100 epochs with a batch size of 16. The optimization method is AdamW [23], and the cosine annealing schedule [22] is leveraged. The learning rate decays from 2×10^{-4} to 1×10^{-6} within 100 epochs. The hyper-parameters are selected as $\alpha = 0.25$ in Eq. (9), $k = 3$ and $\delta = 16$ for bi-polarity hypergraph construction.

5.2. Quantitative Comparison

Table. 2 shows the quantitative comparison of our proposed method with other comparison methods. From the table, we could observe that our proposed method significantly outperforms all comparison methods and achieve state-of-the-art performance. Specifically, compared with the second-best method, *i.e.*, the combination of the state-of-the-art 2D event-based feature tracking method DeepEvT [26] and stereo depth estimation method SDE [28], our proposed method reduces the RMSE by 37.5% and improves the FA by 12.4%, 20.7%, and 25.1% in terms of $c = 0.1$ m, 0.15 m, and 0.2 m, respectively.

Compared to comparison methods that achieve trajectory prediction and depth estimation separately, our proposed method leverages a joint framework to track the 3D feature trajectories of high-speed moving objects. This indicates that for 3D moving objects, the feature trajectory in the camera plane is highly correlated with the 3D position. The simple combination of 2D trajectory prediction and 3D position estimation will lead to fatal errors. Instead, our proposed method tracks 3D trajectories accurately using the stereo motion consistency constraint. Meanwhile, compared to traditional stereo depth estimation methods, our proposed patch matching module uses a high-order correlation modeling mechanism based on bi-polarity hypergraph to eliminate mismatching of similar features, further enhancing the 3D feature tracking robustness.

Table. 3 shows the inference time comparison of our proposed method with other comparison methods. Specifically, we test the inference time of each tracking update step. From the table, we could observe that compared to the second-best method, *i.e.*, DeepEvT + SDE, our proposed method reduces the inference time by 56.8% while achieving better tracking performance. This demonstrates the computational efficiency of our proposed method.



(a) Reference Feature Patch (b) DeepEvT + SDE (c) Ours

Figure 5. Qualitative comparison on our E-3DTrack dataset. From left to right: the reference feature patch, the ground truth feature trajectories (red), the feature trajectories (blue) predicted by DeepEvT [26] + SDE [28] and our proposed method, respectively.

5.3. Qualitative Comparison

Figure 5 shows the qualitative 3D feature tracking results of our proposed method and the second-best comparison method, *i.e.*, DeepEvT [26] + SDE [28]. The predicted 3D trajectories and the ground truth trajectories are shown in blue and red, respectively. From the figure, we could observe that our proposed method achieves more robust 3D feature tracking. As shown in the first row, the comparison method achieves adequate feature tracking performance when facing simple scenarios where the object motions do not contain significant depth changes. Such scenarios are similar to 2D feature tracking. Similar observations could be found in the second row. For the white geometric model

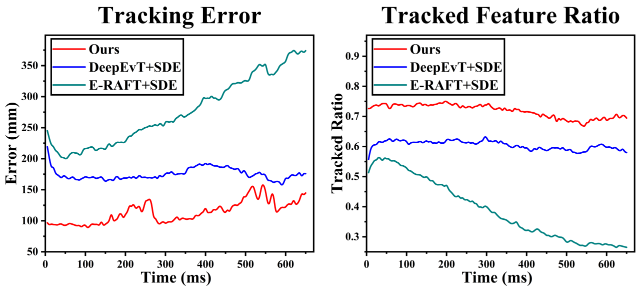


Figure 6. Results of the mean tracking error (left) and the feature tracked ratio (right) over tracking time.

with slight depth variation, the comparison method achieves 3D feature tracking with slight oscillations. However, for the red star with large depth variation and rotation, it could not be tracked accurately by the comparison method. The last two rows show two extreme scenarios, *i.e.*, the 3D motions of the objects are with large depth variation and rotation, which will cause significant feature shape deformation. Under such scenarios, our comparison method tracks the features with fatal errors. In contrast, our proposed method tracks the 3D trajectories of the high-speed moving features robustly and continuously due to our motion compensation module and patch matching module.

Figure 6 further shows the tracking error (RMSE) and the tracked feature ratio (TFR) over time on our E-3DTrack dataset. The threshold is selected as $c = 0.1$ m to calculate TFR. From the figure, we could observe that our proposed method can continuously track 3D trajectories of target features, *i.e.*, our method maintains a high TFR consistently. From the figure, we could also observe that the TFR of E-RAFT + SDE is comparable with DeepEvT + SDE in initial stage, but gradually decreases over time. This is because the optical flow estimation method is lack of long-term consistent modeling. In contrast, our proposed method maintains a high TFR and a low tracking error over all time.

5.4. Ablation Experiments

To demonstrate the effectiveness of each proposed module, we validate the performance of our model with and without the motion compensation module (denoted as MC), stereo motion consistency mechanism (denoted as $\mathcal{L}^{\text{smc}}$), and the bi-polarity hypergraph-based high-order correlation modeling mechanism (denoted as BiHCM), respectively. The ablation experimental results are shown in Tab. 4. See supplementary material for detailed settings.

Bi-Polarity Hypergraph Modeling. From Tab. 4 we could observe that compared with our base model (row (1)), the addition of BiHCM will increase TFR from 0.5586 to 0.6082. Compared with our full model, the removal of the BiHCM will lead to an RMSE increase of 20.8%. This is due to the fact that our proposed BiHCM could enhance the connection between patches with similar features that are spatially close, and suppress patches with similar features but spatially distant, which could eliminate mismatching and further improve 3D feature tracking performance.

Table 4. Ablation experiments on our E-3DTrack dataset.

	BiHCM	$\mathcal{L}^{\text{smc}}$	MC	$\text{TFR}_{0.1\text{ m}}\uparrow$	RMSE↓
(1)	✗	✗	✗	0.5586	0.1807
(2)	✓	✗	✗	0.6082	0.1505
(3)	✗	✓	✗	0.5942	0.1512
(4)	✗	✗	✓	0.5660	0.1624
(5)	✓	✓	✗	0.6705	0.1268
(6)	✓	✗	✓	0.6599	0.1312
(7)	✗	✓	✓	0.6441	0.1427
(8)	✓	✓	✓	0.6928	0.1181

Stereo Motion Consistency. As shown in Tab. 4, compared with the base model, the addition of stereo motion consistency constraint will reduce RMSE from 0.1807 to 0.1512. Compared with the full model, the removal of the stereo motion consistency constraint will increase RMSE by 11.1%. This is due to the fact that the stereo motion consistency could effectively constrain the correlation between the 2D trajectory and 3D spatial position of the objects, making our method predict more accurate and smooth 3D trajectory.

Motion Compensation. As shown in Tab. 4, compared with the base model and the full model, the addition and removal of the motion compensation module resulted in 10.1% and 7.4% decrease and increase in RMSE, respectively. With the addition of the motion compensation module, our proposed method could better deal with feature deformation caused by depth changes and rotations of moving objects, and achieve more robust 3D feature tracking.

These ablation experiments demonstrate the effectiveness of each proposed module.

6. Conclusion

In this paper, we propose the first high-speed 3D feature tracking method that takes stereo event streams as input to estimate 3D feature trajectories. Our proposed method leverages a joint framework to obtain 3D feature trajectories by estimating the feature motion offsets and spatial position simultaneously. A motion compensation module and a patch matching module based on bi-polarity hypergraphs are proposed to achieve robust feature tracking. Meanwhile, the first 3D feature tracking dataset containing high-speed moving objects and ground truth 3D feature trajectories at 250 FPS is constructed, named E-3DTrack, which can be use as the first 3D feature tracking benchmark.

7. Acknowledgment

This work was supported by National Natural Science Funds of China (No. 62021002 and No. 62088102), Beijing Natural Science Foundation (No. 4222025).

References

[1] Himanshu Akolkar, Sio-Hoi Ieng, and Ryad Benosman. Real-Time High Speed Motion Prediction Using Fast Aperture-Robust Event-Driven Visual Flow. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1):361–372, 2020. 2

[2] Ignacio Alzugaray and Margarita Chli. ACE: An Efficient Asynchronous Corner Tracker for Event Cameras. In *Int. Conf. on 3D Vis.*, pages 653–661. IEEE, 2018. 2

[3] Ignacio Alzugaray and Margarita Chli. HASTE: Multi-Hypothesis Asynchronous Speeded-up Tracking of Events. In *The British Machine Vision Conference*, page 744, 2020. 2

[4] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 884–892, 2016. 2

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up Robust Features (SURF). *Comput. Vis. and Image Underst.*, 110(3):346–359, 2008. 1

[6] Paul J Besl and Neil D McKay. Method for Registration of 3-D Shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, pages 586–606, 1992. 2

[7] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 dB 3 μs Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE J. of Solid-State Circuits*, 49(10):2333–2341, 2014. 2

[8] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3867–3876, 2018. 1

[9] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. HGNN$^+$: General Hypergraph Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3181–3199, 2023. 4

[10] 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, 3, 6, 7

[11] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining Events and Frames Using Recurrent Asynchronous Multimodal Networks for Monocular Depth Prediction. *IEEE Robot. and Autom. Lett.*, 6(2):2822–2829, 2021. 3

[12] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-RAFT: Dense Optical Flow from Event Cameras. In *Int. Conf. 3D Vis.*, pages 197–206. IEEE, 2021. 2, 6, 7

[13] Gallego Guillermo, Delbruck Tobi, Michael Orchard Garrick, Bartolozzi Chiara, Taba Brian, Censi Andrea, Leutenegger Stefan, Davison Andrew, Conradt Jorg, Daniilidis Kostas, and Scaramuzza Davide. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2

[14] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *Int. Conf. on 3D Vis.*, pages 534–542. IEEE, 2020. 3

[15] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. Event-aided direct sparse odometry. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5781–5790, 2022. 5, 6

[16] Sumin Hu, Yeeun Kim, Hyungtae Lim, Alex Junho Lee, and Hyun Myung. eCDT: Event Clustering for Simultaneous Feature Detection and Tracking. In *Int. Conf. Intel. Robot. Syst.*, pages 3808–3815. IEEE, 2022. 2

[17] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera. In *Eur. Conf. Comput. Vis.*, pages 349–364, 2016. 1

[18] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *Int. Conf. Intell. Robot. Syst.*, pages 16–23. IEEE, 2016. 2

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2117–2125, 2017. 4

[20] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally Optimal Contrast Maximisation for Event-based Motion Estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6349–6358, 2020. 1

[21] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Flow2stereo: Effective Self-Supervised Learning of Optical Flow and Stereo Matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6648–6657, 2020. 5

[22] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *Int. Conf. Learn. Represent.*, 2017. 7

[23] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *Int. Conf. Learn. Represent.*, 2019. 7

[24] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.*, 60:91–110, 2004. 1

[25] Bruce D Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, pages 674–679, 1981. 1, 6

[26] 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. 2, 3, 4, 6, 7

[27] 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. of Robot. Resear.*, 36(2): 142–149, 2017. 5, 6

[28] 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. 3, 6, 7

[29] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single Image Optical Flow Estimation with an Event Camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1669–1678. IEEE, 2020. 2

[30] Federico Paredes-Vallés, Kirk YW Scheper, and Guido CHE De Croon. Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events

to Global Motion Perception. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(8):2051–2064, 2019. 2

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An Imperative Style, High-Performance Deep Learning Library. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 6

[32] Lichtsteiner Patrick, Posch Christoph, and Delbruck Tobi. A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE J. of Solid-State Circuits*, 43 (2):566–576, 2008. 2

[33] Jianbo Shi and Carlo Tomasi. Good Features to Track. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 593–600. IEEE, 1994. 1

[34] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *Adv. Neural Inform. Process. Syst.*, 28, 2015. 4

[35] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point. *Int. J. Comput. Vis.*, 9(137-154):3, 1991. 1

[36] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object Tracking by Jointly Exploiting Frame and Event Domain. In *Int. Conf. Comput. Vis.*, pages 13043–13052, 2021. 1

[37] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking Transformers for Event-Based Single Object Tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8801–8810, 2022. 1

[38] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-Based Feature Tracking with Probabilistic Data Association. In *IEEE Int. Conf. Robot. Autom.*, pages 4465–4470. IEEE, 2017. 2

[39] Alex Zihao Zhu, Yibo Chen, and Kostas Daniilidis. Real-time Time Synchronized Event-Based Stereo. In *Eur. Conf. Comput. Vis.*, pages 433–447, 2018. 3, 6, 7

[40] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 989–997, 2019. 3

[41] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-Based Visual Inertial Odometry. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5391–5399, 2017. 1