Data Association between Event Streams and Intensity Frames under Diverse Baselines

Dehao Zhang[1,2] Qiankun Ding[3] Peiqi Duan[1] Chu Zhou[1] Boxin Shi[1,2,4,5] ✉

[1] NERCVT, School of Computer Science, Peking University
[2] AI Innovation Center, School of Computer Science, Peking University
[3] Yuanpei College, Peking University
[4] Institute for Artificial Intelligence, Peking University
[5] Beijing Academy of Artificial Intelligence

Abstract. This paper proposes a learning-based framework to associate event streams and intensity frames under diverse camera baselines, to simultaneously benefit camera pose estimation under large baselines and depth estimation under small baselines. Based on the observation that event streams are globally sparse (a small percentage of pixels in global frames are triggered with events) and locally dense (a large percentage of pixels in local patches are triggered with events) in the spatial domain, we put forward a two-stage architecture for matching feature maps. LSparse-Net uses a large receptive field to find sparse matches while SDense-Net uses a small receptive field to find dense matches. Both stages apply Transformer modules with self-attention layers and cross-attention layers to effectively process multi-resolution features from the feature pyramid network backbone. Experimental results on public datasets show a systematic performance improvement for both tasks compared to state-of-the-art methods.

1 Introduction

Event cameras are biologically-inspired imaging sensors that are now experiencing a growing research community. Distinct from traditional cameras that record the scene as a sequence of frames, event cameras asynchronously measure log-intensity changes for each pixel and only capture the dynamic visual scenarios. This unique design brings high temporal resolution ($< 10\mu s$), high dynamic range ($> 120dB$), and low power consumption ($< 0.1W$) for event cameras [34, 40], which give event cameras the potential to handle high-speed motion and extreme lighting scenarios with low power consumption, *e.g*, image reconstruction [45, 20, 12, 39, 56, 11], optical flow estimation [49, 16], 3D scene reconstruction [58, 2], tracking [17], scene depth estimation[16], and visual SLAM [38].

Autonomous driving and Augmented Reality (AR) usually adopt the map-based pose estimation to locate the camera pose [3, 26, 28]. A common way of existing methods is to first establish a point cloud through Structure from Motion

✉ Corresponding author: shiboxin@pku.edu.cn

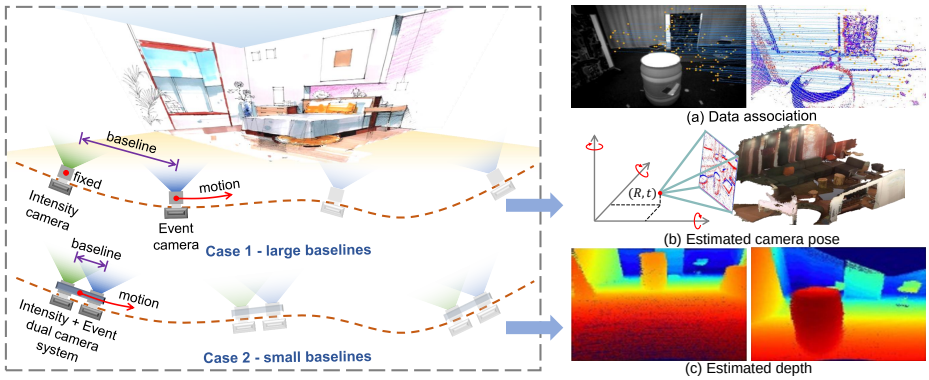

Fig. 1. This paper establishes the data association (a) between event streams and intensity frames under diverse baselines. There are two typical application scenarios: Case 1 – The intensity camera is fixed and the event camera can move freely for pose estimation (b). Case 2 – Two cameras are bundled into one system and shoot the scene in synchronization for depth estimation (c).

(SfM) and obtain the camera pose for one reference image, and then establish *data association* (pixel-level correspondence as shown in Fig. 1 (a)) [68] between subsequent images and the reference image to locate camera pose of the moving device. Estimating the depth through a stereo camera system is also one of the classical topics that are tackled by establishing the data association. The disparity and depth of the scene can be recovered after calibrating the camera parameters and matching all the pixels in two views of a stereo camera system. These image-based camera pose estimation and depth estimation methods perform unsatisfyingly when suffering from over-/under-exposure and motion blur, which coincidentally match the strengths of event cameras: high temporal resolution and high dynamic range can withstand such unfriendly scenarios. This inspires researchers to introduce events to benefit and improve the performance of the two tasks mentioned above.

For camera pose estimation (Fig. 1 (b)), the probabilistic generative event model is applied to jointly process the events triggered at intensity edges and the velocity of the camera [3, 14]. They first obtain the pose of the frame-based camera, and then estimate the pose of the event camera by predicting optical flow maps from event streams. These models require reliable prior information of a reference camera's initial pose and motion, and the data association between two cameras becomes unreliable as the error of the optical flow estimation accumulates with the baselines becoming larger. For depth estimation (Fig. 1 (c)), EMVS [44] first proposes the event-based multi-view stereo method, but the outputting depth maps are sparse. Subsequent deep learning-based algorithms can be divided into two categories. One category depends on a local correlation layer [10, 51, 25, 23, 24], assuming a disparity range and calculating local similarities of deep features only within the range, which lacks generalizability due to

Table 1. Characteristic comparison with state-of-the-art camera pose estimation methods.

	History poses	Motion parameters
Gallego *et al.* [14]	Full history	Yes
Bryner *et al.* [3]	The Last pose	Yes
Ours	**No**	**No**

the assumed disparity range; the other category simply fuses the feature maps of event stream and intensity frame through attention layers and directly predicts the depth map [66, 68], which endures inconsistent baselines between the training and testing sets. The above existing methods demonstrate the need for robust data association between two data modalities, *i.e*, intensity frames and event streams, whether with a large baseline (pose estimation) or a small baseline (depth estimation) between two cameras. However, existing methods such as SIFT [36, 37], ORB [48], or CNN-based ones [61, 8, 9] are not suitable to establish events and frames data association since they mainly focus on local regions. It is vital to have a large receptive field to utilize the information provided by overall edges and contours.

In this paper, we propose a learning-based framework to deal with data association between event streams and intensity frames under diverse baselines. In detail, we use the feature pyramid network (FPN) backbone [35] to extract features and make them more distinguishable. Based on the observation that event streams are globally sparse (a small percentage of pixels in global frames are triggered with events) and locally dense (a large percentage of pixels in local patches are triggered with events) at spatial perspective (as Fig. 1 (a) shows), we put forward a two-stage architecture for matching feature maps. **LSparse-Net** indicates a neural network using a **L**arge receptive field to find **Sparse** matches; **SDense-Net** indicates a neural network using a **S**mall receptive field to find **Dense** matches. Both two stages apply Transformer modules[57] with self-attention layers and cross-attention layers to process the multi-resolution features from the FPN backbone. In tasks with large baselines, where receptive fields differ largely from each other, we can match features through low-level data association; especially for pose estimation, our framework does not require the history of camera poses for estimation [3, 19] (Table 1). In tasks with small baselines, where receptive fields differ slightly from each other, our framework can provide high-level dense data association to refine the final outputs. To summarize, our primary contributions are threefold:

- We introduce Transformer modules to establish data association between event streams and intensity frames, making features more distinguishable with global information, and design a two-stage architecture according to the globally sparse and locally dense characteristics of events streams.
- Our proposed framework supports the establishment of data association between event streams and intensity frames under any baseline without requiring the history of camera poses and any extra clue.
- Our proposed framework demonstrates state-of-the-art performance for both downstream tasks with large baselines (pose estimation) and small baselines (depth estimation).

2 Related Work

Data Association. Establishing data association is a fundamental problem in the SLAM literature [42, 53, 54]. Most SLAM or 3D vision data association are built upon the outputs of frame-based cameras. Conventional optimization-based methods [37, 48] are proposed to use handcrafted local features invariant to rotations and scales to solve this problem. With the development of neural networks, some learning-based approaches [61, 8, 9] can extract significant local features. The learning-based approaches significantly improve the performance on large viewpoint and illumination changes of local features. However, these detector-dependent approaches mainly focus on local regions of images, and could not make full use of global information. They are inherently unable to find similar points from different regions. Unlike other methods which aim to deal with data association between a pair of intensity frames, Gallego *et al.* [14] build the data association between event streams and intensity frames upon the prior knowledge by calculating the optical flow of cameras. Later, Bryner *et al.* [3] propose a maximum-likelihood framework, which optimizes non-linearly according to the camera poses and velocities, to establish the data association.

Event-based Pose Estimation. Existing methods for event-based pose estimation differ in settings. Several methods [58, 5, 60] perform motion correcting for event streams based on depth information and the pose trajectory of cameras. Event streams are treated as intensity frames for local feature extraction and local feature matching by using RANSAC [13] for robust pose estimation. Muglikar *et al.* [41] adopt a similar pipeline for pose estimation with a slight difference that they directly reconstructed event streams into intensity frames. Another set of works [63, 62] calculate pose by maximizing the spatio-temporal consistency of stereo event-based data for camera tracking and 3D reconstruction, without using intensity frames. The most relevant papers [18, 3, 19] to our work are based on a generative event model within a maximum-likelihood framework, where extra clues (*e.g*, pose and motion of cameras, optical flow, *etc*) are required. On the contrary, our work does not require any extra clue.

Event-based Stereo. Existing methods for event-based depth estimation fall into two categories: two event cameras, one event camera and one frame-based camera. For the former category, Zhu *et al.* [64] propose a method which acquires a sparse depth map using a dual event camera stereo imaging system. It requires the velocity of the camera to generate a time-synchronized event disparity volume, and then applies numerical optimization methods to minimize the matching cost between two event disparity volumes. Zou *et al.* [67] recover dense depth maps from sparse event data. Ahmed *et al.* [1] design an end-to-end neural network, which first reconstructs intensity frame from event streams through learning and then calculates dense depth maps based on stereo intensity frames. The latter category is more relevant to our work [59, 68]. The method proposed by Wang *et al.* [59] is similar to that of Ahmed *et al.* [1] as they both transform event streams into intensity frames for depth estimation. The method proposed by Zuo *et al.* [68] is an end-to-end approach, which directly takes event streams and intensity frames as input and outputs a disparity map through pyramid

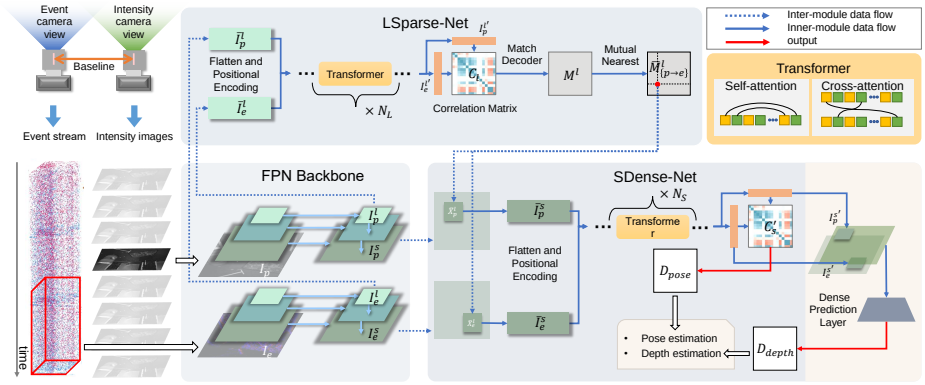

Fig. 2. The architecture of our proposed framework, which consists of three modules: FPN backbone, LSparse-Net, and SDense-Net. The backbone extracts dual-resolution feature maps from the intensity frame I_p and the time surface of the event stream I_e, respectively. LSparse-Net establishes sparse data association from coarse-level features, while SDense-Net establishes dense data association from fine-level features.

attention layers and a U-Net [47] structure. Additionally, Li *et al*. [33] propose STTR, which abandons the cost of volume construction and establishes data association between stereo intensity images using Transformer modules. This work is closely related to our work in terms of models.

3 Proposed Method

In this section, we first introduce the formulation of the problem we target and explain our overall framework in Sec. 3.1 and Fig. 2. Then, we describe our two-stage architecture in Sec. 3.2 and Sec. 3.3 in detail. Finally, we introduce our dense output decoder in Sec. 3.4 to ensure outputs consistency. The implementation details are in Sec. 3.5.

3.1 Problem Formulation and Overall Framework

We aim to establish data association between pair-wise event streams and intensity frames, captured by two separated or bundled cameras in arbitrary poses (Fig. 1), by finding pixel-level matches, without geometric constraints such as homography and epipolar constraint. Events are triggered by an event camera whenever the log intensity change d at a given pixel is larger than a threshold c. Each event is recorded as a four-attribute tuple $\{x, y, t, p\}$, where (x, y) is the coordinates, t is the timestamp, and p is the polarity given by: $p = 1$ if $d \geq c$ and $p = -1$ if $d \leq c$. An event stream is a stacking of such events generated by event cameras. We first transform event streams into four-channel time surfaces [7, 30], whose first two channels record the number of positive and negative events and last two channels record the timestamp of the latest positive and negative events

triggered at each pixel. Then, given an intensity frame $I_p^i \in \mathbb{R}^{H_i \times W_i \times 1}$ and the time surface $I_e^i \in \mathbb{R}^{H_i \times W_i \times 4}$ of an event stream, we formulate the problem as finding pixel-level matches between this pair of data. Specifically, the matching procedure could be expressed as $\boldsymbol{x_e} = \mathcal{F}_\Theta(\boldsymbol{x_p}|I_p^i, I_e^i)$. where $\boldsymbol{x_p}$ and $\boldsymbol{x_e}$ denote the matching pixel coordinates from I_p^i and I_e^i respectively, and $\mathcal{F}_\Theta$ denotes the matching function parameterized by Θ.

Based on the globally sparse and locally dense properties of event streams, we design the matching function as a two-stage framework: The first stage is a matching module aiming to establish sparse data association at the global level and the second stage is a matching module aiming to establish dense data association at the local level. As shown in Fig. 2 , dual-resolution feature maps, $I_p^l, I_e^l \in \mathbb{R}^{H_l \times W_l \times 1}$ with coarse resolution and $I_p^s, I_e^s \in \mathbb{R}^{H_s \times W_s \times 1}$ with fine resolution, are first extracted from the intensity frame and the time surface of event streams by the FPN backbone. After the FPN backbone, regarding the difference of events in density, we design two distinct matching modules: LSparse-Net and SDense-Net. LSparse-Net takes the flattened feature maps $\bar{I}_p^l, \bar{I}_e^l$ from the large receptive fields as input and outputs sparse matches $\bar{M}z_{p\to e}^l \in \{0,1\}^{(H_i \times W_i)\times(H_i \times W_i)}$, where $\bar{M}_{p\to e}^l(\bar{\boldsymbol{x}}_p^l, \bar{\boldsymbol{x}}_e^l) = 1$ represents the match between $\bar{I}_p^l(\bar{\boldsymbol{x}}_p^l)$ and $\bar{I}_e^l(\bar{\boldsymbol{x}}_e^l)$. Unlike LSparse-Net which aims to establish long-range data association, SDense-Net aims to find a dense, local data association. SDense-Net takes feature maps I_p^s, I_e^s from small receptive fields and outputs final dense matching results $\boldsymbol{x}_e = \mathcal{F}_\Theta(\boldsymbol{x}_p|I_p^i, I_e^i)$, where $I_p(\boldsymbol{x}_p)$ matches $I_e(\boldsymbol{x}_e)$. The final loss function then becomes the weighted sum of losses from each matching model. We further add a dense prediction layer to ensure outputs consistency and make the framework compatible with the depth estimation task. We will introduce in detail the design of two stages in the following subsections.

3.2 LSparse-Net

We exploit the global information of each view to achieve the coarse matching, thereby avoiding large matching errors caused by local feature differences between intensity frames and event streams. Specifically, we choose to use Transformer module in LSparse-Net (refer to Fig. 2 top), similar to the design of methods [50, 52], as Transformer module can enlarge each feature's receptive field and thereby include long-range association during matching. Each Transformer module consists of self-attention and cross-attention layers in an alternative order. Feature maps I_p^l, I_e^l, after being forwarded by the Transformer module, become more discernible in the form of $I_p^{l'} \in \mathbb{R}^{H_l \times W_l \times D_l}, I_e^{l'} \in \mathbb{R}^{H_l \times W_l \times D_l}$. Then, we design a correlation layer to exhaustively compute the cosine similarity between each pair of feature descriptors, to build the correlation matrix $C_l \in \mathbb{R}^{H_l \times W_l \times H_l \times W_l}$. Usually, there are more matches in the vicinity of a correct match. This inspires us to design a 4-D CNN structure for matching and filtering the correlation matrix, by using the mutual nearest neighbor filter.

Transformer Module. We use linear Transformer [27] to reduce the computational complexity and apply positional encoding to encode location information

into features. The original Transformer put forward by Vaswani *et al.* [57] is an encoder-decoder architecture, where the encoder consists of sequentially connected encoder layers. For each encoder layer, the most critical feature is the attention layer, which takes query vector Q, key vector K, and value vector V as input. In self-attention layers, Q, K, V are transformed from the same input vector with different weights and in cross-attention layers, Q, K, V are transformed from different input vectors.

Linear Transformer, proposed by Katharopoulos *et al.* [27], aims to reduce the computational costs caused by the dot product between Q and K. The dot product from the attention layer in the original Transformer is substituted with an alternative kernel function:

$$\text{Similarity}(Q, K) = \phi(Q) \cdot \phi(K)^\top, \tag{1}$$

where $\phi(\cdot) = \text{elu}(\cdot) + 1$. As D (the scale of $\phi(Q), \phi(K)$) is much smaller than N (the scale of Q and K), the computational complexity is reduced to $O(N)$.

Positional encoding is added to each element to encode positional information. Following DETR [4] and LoFTR [52], we apply the 2D extension of positional encoding in our Transformer module.

Our Transformer module takes feature maps $\bar{I}_p^l, \bar{I}_e^l$ as input. In the self-attention layer, the two inputs are identical: either $(\bar{I}_p^l, \bar{I}_p^l)$ or $(\bar{I}_e^l, \bar{I}_e^l)$; in cross-attention layer, the two inputs differ from each other: $(\bar{I}_e^l, \bar{I}_p^l)$. This module does not change the shape of feature map, but instead applies the attention layer to encoding more context information into the features for a better recognizability.

Correlation Layer. To acquire pairwise feature similarity, we apply a correlation layer [32] to calculate the cosine similarity between feature descriptors and normalize features with $\ell 2$ norm before and after the correlation layer. The output feature maps from the Transformer module $I_p^{l'} \in \mathbb{R}^{H_l \times W_l \times D_l}, I_e^{l'} \in \mathbb{R}^{H_l \times W_l \times D_l}$ have a spatial size of $H_l \times W_l$ and dimensionality D_l. Let $I_p^{l'}(i, j) \in \mathbb{R}^{D_l}$ denote the feature vector at a spatial location (i, j), then the correlation layer evaluating the pairwise similarities between all locations in feature maps $I_p^{l'}$, $I_e^{l'}$ can be calculated as

$$C_l(i, j, m, n) = I_p^{l'}(i, j) \cdot I_e^{l'}(m, n), \tag{2}$$

where $\cdot$ denotes the scalar product. The final output is the 4-D correlation matrix C_l capturing the similarities between all pairs of spatial locations.

Matching Decoder. The previous correlation layer builds a dense correlation matrix, but it is a significant challenge to determine which pairs are correct matches. To discriminate a reliable match, we apply the network proposed by Rocco *et al.* [46]. Since correct matches tend to have a coherent set of supporting matches surrounding them in the 4-D space, by processing correlation matrix with 4-D convolutional network we can establish a strong locality prior to the relationships between the matches. In our implementation, we apply three layers of 4-D convolutional blocks to capture the match patterns and ReLU activation function in the last layer. The output M^l has only one channel and maintains the same shape as the input.

Mutual Nearest Neighbor Filtering. If $I_p^l(i,j)$ and $I_e^l(m,n)$ matches, it simultaneously means that $I_e^l(m,n)$ is the closest feature to $I_p^l(i,j)$ in I_e^l and $I_p^l(i,j)$ is the closest feature to $I_e^l(m,n)$ in I_p^l. Therefore, we use this rule to further filter the matches and eventually acquire the final sparse results. We denote the final match matrix as $\bar{M}^l$.

Loss Function. In LSparse-Net, we acquire sparse data association. Therefore, we apply negative log-likelihood loss. With respect to each set of ground truth match $M_{\text{gt}}^l = ((i_1, j_1, m_1, n_1), ..., (i_N, j_N, m_N, n_N))$, we calculate the loss as:

$$L_l = -\frac{1}{N} \sum\nolimits_{i,j,m,n} \log\ \bar{M}^l(i,j,m,n). \tag{3}$$

3.3 SDense-Net

LSparse-Net is a sparse matching module capable of detecting long-range association, while SDense-Net is in charge of establishing local dense data association in patches which have been matched by LSparse-Net. SDense-Net (refer to Fig. 2 bottom right) consists of Transformer modules and correlation layers similar to LSparse-Net. The distinction is that, to acquire dense matches, the matching decoder of SDense-Net differs from its counterpart in LSparse-Net.

Matching Decoder. The matching decoder in SDense-Net consists of several convolutional blocks and outputs a two-channel tensor with unchanged resolution. Given the output $C_s \in \mathbb{R}^{H_s \times W_s \times H_s \times W_s}$ from correlation layer, it is transformed to $C_s^{'} \in \mathbb{R}^{H_s \times W_s \times (H_s \times W_s)}$ for the convenience of CNN-based processing, where the third dimension denotes channels. This decoder outputs a two-channel matrix D_{pose}, indicating the displacement of matching pixel's coordinates on X-axis and Y-axis respectively.

Loss Function. In SDense-Net, we acquire dense data association. Therefore, we apply $\ell 2$ loss. Suppose the input of this module has M points, then for each point (i,j) from $\bar{I}_p^s$, there is a corresponding point from $\bar{I}_e^s$, whose loss could be calculated as:

$$L_s = \frac{1}{M} \sum\nolimits_{i,j} \|(i-m, j-n)\|_2. \tag{4}$$

3.4 Dense Prediction Layer

We add a dense prediction layer to ensure dense output consistency and make it compatible with dense outputting tasks such as predicting depth maps or disparity maps. This dense prediction layer consists of several convolution layers and a ReLU activation function at the end. We denote D_{depth} as the output of this layer.

Loss Function. We apply the L_{c} loss for per-pixel supervision to minimize the difference between ground truth and dense predictions. Given a batch of ground truth with pixel labels $\hat{y}$, pixel-wise dense predictions y, and the number of pixels H, L_{c} loss is defined as

$$L_{\text{c}} = \frac{1}{H} \sum\nolimits_{i=1}^{H} \|\hat{y}_i - y_i\|. \tag{5}$$

3.5 Implementation Details

The resolution of event streams and intensity frames is 260×346. The entire model is trained end-to-end with randomly initialized weights. The LSparse-Net consists of two Transformer layers and SDense-Net consists of one Transformer layer. The dimension of Transformer is set to 512. The match decoder in LSparse-Net and the match decoder in SDense-Net each consists of three CNN layers.

The final loss is the weighted sum of the loss functions of LSparse-Net and SDense-Net:

$$L = L_l + \alpha \cdot L_s + \beta \cdot L_c, \tag{6}$$

where α and β are hyper parameters. For the tasks which does not require dense output, we don't use the additional dense output layer and we set $\alpha = 2$ and $\beta = 0$. Otherwise, we set both $\alpha = \beta = 2$.

All of the models are trained using Adam optimizer [29] with an initial learning rate of 1×10^{-3} and a batch size of 16. In training, the learning rate is cut in half by MultiStepLR scheduler in epoch {3, 9, 15}. All experiments are implemented on NVIDIA TITAN RTX GPUs. For all experiments, we use four GPUs for training and validating.

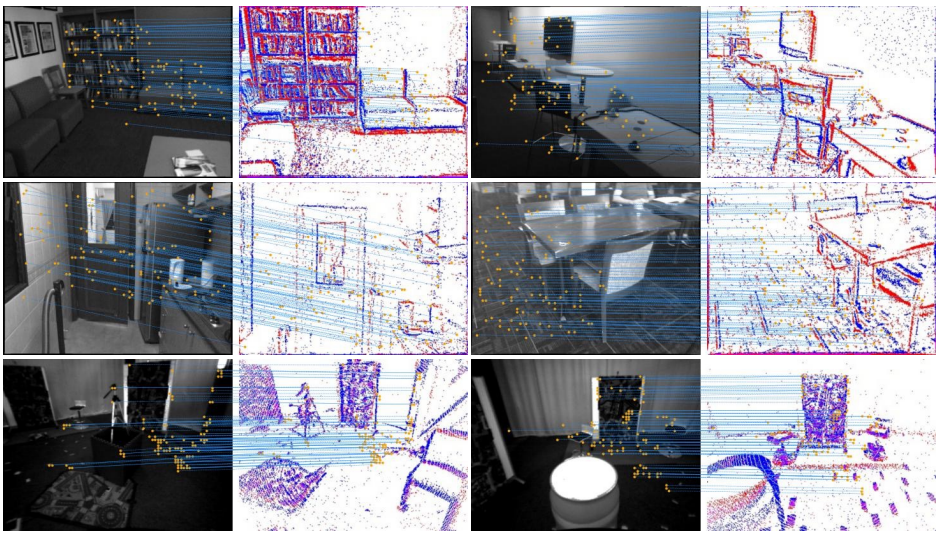

Fig. 3. Exemplars for data association. We randomly sample 100 pairs of matching pixels from all matches for visualization. The first two rows show results of data association on the synthetic data[6], whereas the last row shows results on the real data. As shown, our model can still establish sound data association even when the views of the event stream and the intensity frame differ largely from each other.

Table 2. Pose estimation evaluation on the synthetic data. We report the AUC of the pose error at thresholds (5°, 10°, 20°), where the pose error is defined as the maximum of angular error in rotation and translation.

	Pose estimation AUC		
	5°	10°	20°
Synthetic data	10.40	25.43	39.68

4 Applications

In this section, we first introduce two applications based on the data association we proposed, including pose estimation in 4.1 (for large baselines) and depth estimation in 4.2 (for small baselines). Then, we conduct several ablation studies to verify the validity of each model design choice in 4.3.

4.1 Pose Estimation

To demonstrate that our framework can establish data association under large baselines, we use it to solve the pose estimation problem. We cannot directly use existing benchmark datasets [3, 14] for event-based pose estimation, because our model requires pixel-wise ground truth matches as supervision for training, but existing datasets rarely contain the information of camera poses and high-quality depth images at the same time for us to construct labels. Despite the fact that EVSEC [65] can provide both, it only contains a limited number of scenes, restricting the generalizability of the final model. Eventually, we choose ScanNet [6], an RGB-D video dataset, and generate the corresponding event streams with the event simulator V2E [22] under the default parameter settings to generate the training dataset. ScanNet [6] contains 1613 videos with the ground truth pose and the depth map of each frame. The resolution of the images and depth maps are all 640×480, and the frame rate of videos is 30 fps. We sample a part of the synthetic dataset for training referring to [52, 50] to synthesize event streams by V2E [22]. The sample indices will be provided in the supplementary materials. When generating the event stream corresponding to each intensity frame, we include the latest 20,000 events earlier than the timestamp of the intensity frame, which remains consistent in our training and testing processes on all datasets.

To evaluate the robustness to real data, we choose the scene of an actual room from Bryner *et al*. [3], which consists of a texture-less white wall and some rich-textured objects. The data are acquired by RGB-D cameras and motion capture system, providing depth and pose information. We compare our model (trained only on the synthetic dataset) with recent events and frame-based 6-DoF tracking methods, including the works of Gallego *et al*. [15], and Bryner *et al*. [3]. Note that our method does not require the history of camera poses like Gallego *et al*. [15] and Bryner *et al*. [3], as shown in Table 1, which makes the estimation process more convenient.

Table 3. Pose estimation evaluation on the real data. We report the median accuracy of the results. The position error (Pos.) is given by the Euclidean distance between the ground truth and the estimated event position. The orientation error (Ori.) is measured using the geodesic distance between the ground truth and the estimated event pose.

	Bryner *et al.* [3]		Ours	
	Pos. (cm)	Ori. (°)	Pos. (cm)	Ori. (°)
Room1	9.95	**3.08**	**8.82**	4.12
Room2	9.82	**3.84**	**8.73**	4.63

Test Setting. Referring to the test settings from Bryner *et al.* [3], we respectively choose 284 and 3046 pieces of event streams from trajectory 1 and trajectory 2, and choose the nearest five frames of intensity frames according to the timestamp of the last event to construct data pairs. Instead of relying on the history pose, the purpose of choosing the nearest five intensity frames is to find intensity frames that are spatially overlapped with the current events as reference frames. For a given event stream, to calculate the camera pose corresponding to the last event, we treat the five data pairs as our inputs. After matching those pairs between intensity frames and events, we can establish some correspondence between events and the intensity image. Since the 3D coordinates of intensity frame pixels are known in advance, we can build correspondences between 2D coordinates of events and the 3D coordinates of intensity frame pixels. We solve the PnP problem with the OpenCV `solvePnPRansac` to finally get the camera pose. For the parameter settings, we set `iterationsCount` $= 10000$, `reprojectionError` $= 8.0$, `confidence` $= 0.99$.

Evaluation Protocol and Results. For evaluation on the synthetic dataset, following [31, 52], we report the area under the cumulative curve (AUC) of the pose errors at different thresholds. To recover camera poses, we solve the essential matrix from predicted matches with RANSAC. The evaluation metrics on the synthetic dataset are shown in Table 2. For evaluation on the real-world dataset, we pick the absolute and relative errors of position and rotation as evaluation metrics. As shown in Table 3, our method achieves comparable performance to Bryner *et al.* [3] without relying on any particular initial pose. Note that as analyzed in Bryner *et al.* [3], the ground truth data contain a certain level of noise. Achieving this level of error is almost the limit for any model on this particular dataset. It is demonstrated in Fig. 3 that, for both synthetic and real data, our model is capable of establishing correct matches between event streams and intensity frames in scenes with large baselines and sparse textures, and our model is generalizable from synthetic data to real-world data.

4.2 Depth Estimation

To demonstrate that our framework can establish data association under small baselines, we use it to solve the depth estimation problem. We choose Multi Vehicle Stereo Event Camera Dataset (MVSEC) [65] for training and testing

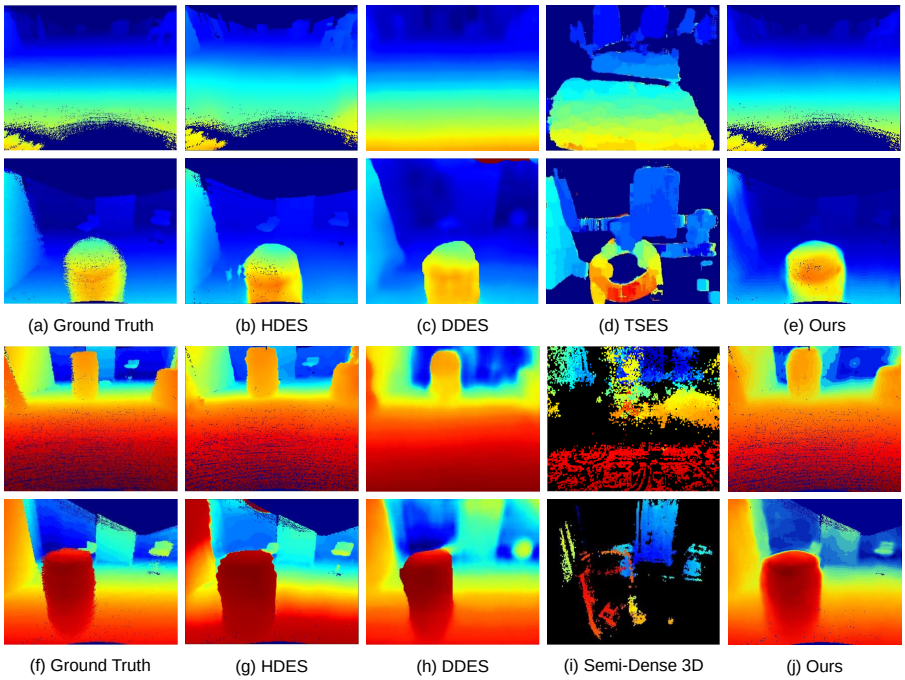

Fig. 4. Qualitative comparison against recent event-based methods on the Indoor Flying dataset. Following [62, 55, 64, 68], we select 4 examples from the dataset and compare with HDES [68], DDES [55], TSES [64], Semi-Dense 3D [43]. From top to bottom, the rows correspond respectively to frame 100 from sequence 1, frame 340 from sequence 1, frame 1700 from sequence 3, and frame 980 from sequence 1. Following HDES [68], we add a mask in our results based on the ground truth, setting pixels as dark blue if their disparity values are missing in the ground truth.

for this task. MVSEC [65] is a widely used event-based stereo dataset collected by LIDAR, IMU, and two event cameras. Each event camera can output event streams and intensity frames with a resolution of 346×260. The product of focal length and the baseline between the two cameras is 19.94. This dataset provides event streams and synchronized intensity frames, depth maps and the poses of cameras calculated by LIDAR and IMU.

For comparison, we choose the Indoor Flying dataset from MVSEC [65], which is taken by a drone in a room with several objects of irregular shapes. As the depth images are sparse on some images, referring to Zhu *et al*. [64], we choose three subsets for training and evaluation, which consist of indoor_flying1: 140-1200, indoor_flying2: 120-1420, indoor_flying3: 73-1616. Following the setup in existing methods [64, 55, 68], we use two subsets for supervised training and one subset for validating and testing. Split 1 means we use the first subset of the sequence as our validation and test dataset. The second and third subsets are used as training data. We use event streams for left-view and intensity frames

Table 4. Depth estimation evaluation on the Indoor Flying dataset. Our method shows clear advantages over DDES [55], SGM [21], TSES [64], and CopNet [62]. Compared with HDES [68], our method demonstrates similar performances in most cases and is slightly ahead in the mean depth error on Split 1. The cells of mean disparity error for HDES [68], DDES [55], and SGM [21] are left blank because the metric of mean disparity error is not calculated for them in their original works.

	Mean depth error (cm)			Mean disparity error (px)			One pixel accuracy (%)		
	Split 1	Split 2	Split 3	Split 1	Split 2	Split 3	Split 1	Split 2	Split 3
HDES [68]	16.0	**28.0**	**18.0**	-	-	-	86.4	49.7	**80.1**
DDES [55]	16.7	29.4	27.8	-	-	-	**89.8**	61.0	74.8
SGM [21]	29.0	36.7	37.9	-	-	-	78.5	64.4	71.0
TSES [64]	36.0	44.0	36.0	0.89	1.98	0.88	82.3	**70.1**	82.3
CopNet [62]	61.0	100.0	64.0	1.03	1.54	1.01	70.0	52.8	70.6
Ours	**15.8**	31.8	19.7	**0.75**	1.82	**0.87**	88.1	50.3	77.4

for right-view as the input of models, and the model outputs disparity maps. Identical to the test setting of pose estimation, for each intensity frame, we include the latest 20,000 events earlier than its timestamp as its corresponding event stream. Eventually, mean disparity error, mean depth error, and one-pixel accuracy are applied to assess our results quantitatively.

We compare our framework with multiple approaches, including methods for stereo event cameras (DDES [55], Semi-Dense 3D [43], CopNet [62]), methods for stereo frame-based cameras (SGM [21]), and methods for stereo systems of a frame-based camera and an event camera (HDES [68]).

Results. Quantitative results are shown in Table 4. The results validate the capability of our framework to solve the basic problem of establishing data association between event streams and intensity frames while being adaptive to multiple tasks. Visual results for comparison are shown in Fig. 4. According to the side-by-side comparison across splits, Split 2 ranks the last. The primary reason is that some data in sequence 2 contain large depths, whereas sequences 1 and 3 do not contain such data. Compared with the state-of-the-art model HDES [68], as shown in the first row, our framework is more accurate in predicting the overall distribution of disparity, whereas HDES [68] predicts better sharp results over the edges, which might be attributed to the design of $L_{\text{smoothness}}$ in HDES [68]. As a whole, using both event streams and intensity frames to estimate depth, our model reaches the state-of-the-art performances.

4.3 Ablation Study

To verify the validity of each module design choice in our framework, we compare the performances of three different sets of module combinations respectively on the two tasks. For pose estimation, as shown in Table 5: 1) Removing SDense-Net while preserving LSparse-Net leads to a significant drop in performances. 2) Changing the layers of the Transformer in LSparse-Net from 2 to 1 leads to a slight drop in AUC. Note that we should not remove the whole LSparse-Net since it would lead to huge video memory consumption. For depth estimation,

Table 5. Ablation study for pose estimation. Three variants of models are trained and evaluated on the synthetic ScanNet [6] dataset. Smaller LSparse-Net denotes that it only contains 1 Transformer layer.

	Pose estimation AUC		
	5°	10°	20°
Remove SDense-Net	2.33	14.23	25.42
Smaller LSparse-Net	9.76	23.42	32.53
Our final model	**10.40**	**25.43**	**39.68**

Table 6. Ablation study for depth estimation. The models below are trained and evaluated on Split 2 of EVSEC [65] dataset.

	Mean depth error (cm)	Mean disparity error (px)	One pixel accuracy (%)
	Split 2	Split 2	Split 2
Remove dense output layer	45.5	2.43	40.8
Remove SDense-Net	37.7	2.02	43.4
Transformer dimension = 256	32.4	1.99	47.3
Our final model	**31.8**	**1.82**	**50.3**

we conduct experiments on Split 2. As shown in Table 6: 1) Removing the dense output layer leads to a significant drop in performances. 2) Removing SDense-Net while preserving LSparse-Net also leads to a drop in performances. 3) Changing the feature dimension for the Transformer in LSparse-Net from 512 to 256 leads to a slight drop in all three metrics. These results demonstrate that our final model achieves the optimal performance with these specific design choices.

5 Conclusions

This paper presents an approach to establish data association between event streams and intensity frames, which not only establishes data association under large baselines and large difference in receptive fields for pose estimation of cameras, but also establishes data association under small baselines and small difference in receptive fields for depth estimation of dual camera system with bundled intensity and event cameras. We achieve this by taking the globally sparse and locally dense feature of event streams into account and establishing data association in a sparse-to-fine manner.

Limitations. Furthermore, we observe that existing real-world datasets for event-based pose estimation and depth estimation using both event streams and intensity frames with high-quality ground truth poses and depth labels are still not ready on large scales. Synthetic datasets, despite their large quantities, are not realistic enough for blur artifacts and HDR properties of events. This is one of the bottlenecks that prevents the proposed method to further explore more reliable data association.

Acknowledgements. This work was supported by National Key R&D Program of China (2021ZD0109803) and National Natural Science Foundation of China under Grant No. 62136001, 62088102.

References

1. Ahmed, S.H., Jang, H.W., Uddin, S.N., Jung, Y.J.: Deep event stereo leveraged by event-to-image translation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 882–890 (2021)
2. Alexis, B., Zihao, W.W., Oliver, C., Aggelos, K.K.: E3D: event-based 3d shape reconstruction. CoRR **abs/2012.05214** (2020), https://arxiv.org/abs/2012.05214
3. Bryner, S., Gallego, G., Rebecq, H., Scaramuzza, D.: Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 325–331 (2019)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229 (2020)
5. Censi, A., Scaramuzza, D.: Low-latency event-based visual odometry. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). pp. 703–710 (2014)
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5828–5839 (2017)
7. Delbruck, T.: Frame-free dynamic digital vision. In: Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society. vol. 1, pp. 21–26 (2008)
8. DeTone, D., Malisiewicz, T., Rabinovich, A.: Toward geometric deep slam. arXiv preprint arXiv:1707.07410 (2017)
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
10. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proc. of International Conference on Computer Vision (ICCV). pp. 2758–2766 (2015)
11. Duan, P., Wang, Z., Shi, B., Cossairt, O., Huang, T., Katsaggelos, A.: Guided event filtering: Synergy between intensity images and neuromorphic events for high performance imaging. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
12. Duan, P., Wang, Z., Zhou, X., Ma, Y., Shi, B.: EventZoom: Learning to denoise and super resolve neuromorphic events. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
14. Gallego, G., Forster, C., Mueggler, E., Scaramuzza, D.: Event-based camera pose tracking using a generative event model. arXiv preprint arXiv:1510.01972 (2015)
15. Gallego, G., Lund, J.E., Mueggler, E., Rebecq, H., Delbruck, T., Scaramuzza, D.: Event-based, 6-dof camera tracking from photometric depth maps. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(10), 2402–2412 (2017)
16. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3867–3876 (2018)

17. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: Asynchronous, photometric feature tracking using events and frames. In: Proc. of European Conference on Computer Vision (ECCV) (2018)
18. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: Asynchronous, photometric feature tracking using events and frames. In: Proc. of European Conference on Computer Vision (ECCV). pp. 750–765 (2018)
19. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: Eklt: Asynchronous photometric feature tracking using events and frames. International Journal of Computer Vision pp. 1–18 (2019)
20. Han, J., Zhou, C., Duan, P., Tang, Y., Xu, C., Xu, C., Huang, T., Shi, B.: Neuromorphic camera guided high dynamic range imaging. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
21. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 328–341 (2007)
22. Hu, Y., Liu, S.C., Delbruck, T.: V2E: From video frames to realistic dvs events. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1312–1321 (2021)
23. Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8981–8989 (2018)
24. Hui, T.W., Tang, X., Loy, C.C.: A lightweight optical flow cnn—revisiting data fidelity and regularization. IEEE transactions on pattern analysis and machine intelligence **43**(8), 2555–2569 (2020)
25. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2462–2470 (2017)
26. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image matching across wide baselines: From paper to practice **129**(2), 517–547 (2021)
27. Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F.: Transformers are rnns: Fast autoregressive transformers with linear attention. In: International Conference on Machine Learning. pp. 5156–5165 (2020)
28. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proc. of International Conference on Computer Vision (ICCV). pp. 2938–2946 (2015)
29. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
30. Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.B.: Hots: a hierarchy of event-based time-surfaces for pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(7), 1346–1359 (2016)
31. Li, H., Li, G., Shi, L.: Super-resolution of spatiotemporal event-stream image. Neurocomputing **335**(MAR.28), 206–214 (2019)
32. Li, X., Han, K., Li, S., Prisacariu, V.: Dual-resolution correspondence networks. Advances in Neural Information Processing Systems **33**, 17346–17357 (2020)
33. Li, Z., Liu, X., Drenkow, N., Ding, A., Creighton, F.X., Taylor, R.H., Unberath, M.: Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In: Proc. of International Conference on Computer Vision (ICCV). pp. 6197–6206 (2021)
34. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. IEEE journal of solid-state circuits **43**(2), 566–576 (2008)

35. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2117–2125 (2017)
36. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of International Conference on Computer Vision (ICCV). vol. 2, pp. 1150–1157 (1999)
37. Lowe, D.G.: Distinctive image features from scale-invariant keypoints **60**(2), 91–110 (2004)
38. Maqueda, A.I., Loquercio, A., Gallego, G., García, N., Scaramuzza, D.: Event-based vision meets deep learning on steering prediction for self-driving cars. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
39. Mostafavi Isfahani, S.M., Nam, Y., Choi, J., Yoon, K.J.: E2sri: Learning to super-resolve intensity images from events. IEEE Transactions on Pattern Analysis and Machine Intelligence
40. Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., Scaramuzza, D.: The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. The International Journal of Robotics Research **36**(2), 142–149 (2017)
41. Muglikar, M., Gehrig, M., Gehrig, D., Scaramuzza, D.: How to calibrate your event camera. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1403–1409 (2021)
42. Neira, J., Tardós, J.D.: Data association in stochastic mapping using the joint compatibility test. IEEE Transactions on robotics and automation **17**(6), 890–897 (2001)
43. Piatkowska, E., Kogler, J., Belbachir, N., Gelautz, M.: Improved cooperative stereo matching for dynamic vision sensors with ground truth evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 53–60 (2017)
44. Rebecq, H., Gallego, G., Mueggler, E., Scaramuzza, D.: EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time. International Journal of Computer Vision **126**(12), 1394–1414 (2018)
45. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: Events-to-video: Bringing modern computer vision to event cameras. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3857–3866 (2019)
46. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. Advances in neural information processing systems **31** (2018)
47. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241 (2015)
48. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Proc. of International Conference on Computer Vision (ICCV). pp. 2564–2571 (2011)
49. Rueckauer, B., Delbruck, T.: Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. Frontiers in Neuroscience **10**, 176 (2016)
50. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4938–4947 (2020)
51. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8934–8943 (2018)

52. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8922–8931 (2021)
53. Tardós, J.D., Neira, J., Newman, P.M., Leonard, J.J.: Robust mapping and localization in indoor environments using sonar data. The International Journal of Robotics Research **21**(4), 311–330 (2002)
54. Thrun, S., Burgard, W., Fox, D.: A probabilistic approach to concurrent mapping and localization for mobile robots. Autonomous Robots **5**(3), 253–271 (1998)
55. Tulyakov, S., Fleuret, F., Kiefel, M., Gehler, P., Hirsch, M.: Learning an event sequence embedding for dense event-based deep stereo. In: Proc. of International Conference on Computer Vision (ICCV). pp. 1527–1537 (2019)
56. Tulyakov, S., Gehrig, D., Georgoulis, S., Erbach, J., Gehrig, M., Li, Y., Scaramuzza, D.: Time Lens: Event-based video frame interpolation. In: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
57. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
58. Vidal, A.R., Rebecq, H., Horstschaefer, T., Scaramuzza, D.: Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. IEEE Robotics and Automation Letters **3**(2), 994–1001 (2018)
59. Wang, Z., Pan, L., Ng, Y., Zhuang, Z., Mahony, R.: Stereo hybrid event-frame (shef) cameras for 3d perception. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 9758–9764 (2021)
60. Weikersdorfer, D., Adrian, D.B., Cremers, D., Conradt, J.: Event-based 3d slam with a depth-augmented dynamic vision sensor. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 359–364 (2014)
61. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: Proc. of European Conference on Computer Vision (ECCV). pp. 467–483 (2016)
62. Zhou, Y., Gallego, G., Rebecq, H., Kneip, L., Li, H., Scaramuzza, D.: Semi-dense 3D reconstruction with a stereo event camera. In: Proc. of European Conference on Computer Vision (ECCV). pp. 235–251 (2018)
63. Zhou, Y., Gallego, G., Shen, S.: Event-based stereo visual odometry. IEEE Transactions on Robotics **37**(5), 1433–1450 (2021)
64. Zhu, A.Z., Chen, Y., Daniilidis, K.: Realtime time synchronized event-based stereo. In: Proc. of European Conference on Computer Vision (ECCV). pp. 433–447 (2018)
65. Zhu, A.Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. IEEE Robotics and Automation Letters **3**(3), 2032–2039 (2018)
66. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In: Proc. of Robotics: Science and Systems (2018)
67. Zou, D., Shi, F., Liu, W., Li, J., Wang, Q., Park, P.K., Shi, C.W., Roh, Y.J., Ryu, H.E.: Robust dense depth map estimation from sparse dvs stereos. In: Proc. of British Machine Vision Conference (BMVC). vol. 1 (2017)
68. Zuo, Y.F., Cui, L., Peng, X., Xu, Y., Gao, S., Wang, X., Kneip, L.: Accurate depth estimation from a hybrid event-rgb stereo setup. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6833–6840