Learning Normal Flow Directly From Events

Dehao Yuan
dhyuan@umd.edu

Levi Burner
lburner@umd.edu

Jiayi Wu
jiayiwu@umd.edu

Minghui Liu
minghui@umd.edu

Jingxi Chen
ianchen@umd.edu

Yiannis Aloimonos
jyaloimo@umd.edu

Cornelia Fermüller
fermulcm@umd.edu

Abstract

Event-based motion field estimation is an important task. However, current optical flow methods face challenges: learning-based approaches, often frame-based and relying on CNNs, lack cross-domain transferability, while model-based methods, though more robust, are less accurate. To address the limitations of optical flow estimation, recent works have focused on normal flow, which can be more reliably measured in regions with limited texture or strong edges. However, existing normal flow estimators are predominantly model-based and suffer from high errors.

In this paper, we propose a novel supervised point-based method for normal flow estimation that overcomes the limitations of existing event learning-based approaches. Using a local point cloud encoder, our method directly estimates per-event normal flow from raw events, offering multiple unique advantages: 1) It produces temporally and spatially sharp predictions. 2) It supports more diverse data augmentation, such as random rotation, to improve robustness across various domains. 3) It naturally supports uncertainty quantification via ensemble inference, which benefits downstream tasks. 4) It enables training and inference on undistorted data in normalized camera coordinates, improving transferability across cameras. Extensive experiments demonstrate our method achieves better and more consistent performance than state-of-the-art methods when transferred across different datasets. Leveraging this transferability, we train our model on the union of datasets and release it for public use. Finally, we introduce an egomotion solver based on a maximum-margin problem that uses normal flow and IMU to achieve strong performance in challenging scenarios. Codes are available at github.com/dhyuan99/VecKM_flow.

1. Introduction

Event-based motion field estimation is a challenging problem with significant potential for visual motion interpretation tasks, primarily due to its high temporal resolution, wide dynamic range, and low latency. Image motion is critical for various applications, such as egomotion estimation [67], video interpolation [14], and motion deblurring [54].

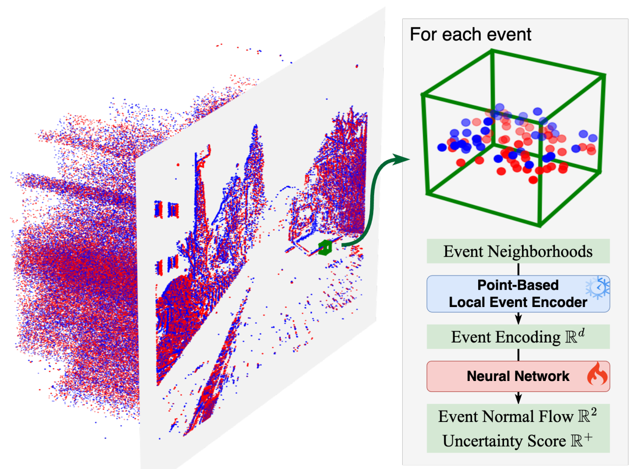

Figure 1. We propose a point-based network for estimating normal flow from raw event data. We discover multiple key advantages of this point-based approach compared with existing learning-based approaches. An event and its neighborhood are first encoded as a fixed-dimensional vector, which is then input to a network trained in a supervised way to predict normal flow. This approach achieves high accuracy while maintaining strong transferability across different domains and datasets. Besides, we demonstrate the usefulness of the estimated normal flow in a new egomotion solver that is shown to remain robust even during aggressive camera motions.

Most works on image motion focus on estimating optical flow (OF). Learning-based OF estimators like [21] perform impressively when evaluated within specific domains or datasets but suffer from accuracy degradation when applied across different domains, as shown in [25, 26]. In contrast, model-based OF estimators like [46] are more robust to domain shifts, but their accuracy is limited, particularly in scenarios where the event textures are sparse.

The lack of robustness in optical flow estimators is mainly due to the local aperture problem [2]. In regions with limited texture or strong linear edges, only the motion component perpendicular to the edges (normal flow) can be reliably measured, while the motion parallel to the

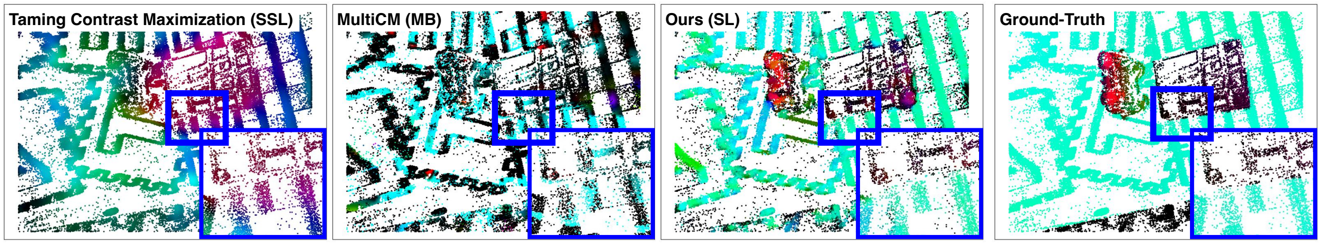

Figure 2. Our point-based method produces accurate and sharp predictions in the presence of independently moving objects, while other methods [35, 46] fail. All models (if learning-based) are trained on DSEC and evaluated on EVIMO2. The flows are displayed in HSV color space, where the hue represents the flow direction, and the brightness represents the flow magnitude.

edges remains ambiguous. To address this issue, optical flow estimators use CNNs and RNNs to enlarge their receptive fields, which makes the models prone to overfitting.

With this observation, some existing methods focus on predicting normal flow (NF) and demonstrate that normal flow is useful for tasks like egomotion estimation [24, 29, 43]. However, current NF estimation approaches are predominantly model-based, relying on fitting a plane to the local space-time event surface [9, 33]. These approaches suffer from limited accuracy. Furthermore, the difficulty in obtaining ground-truth normal flow data has impeded learning-based normal flow estimators.

As shown in Figure 1 and 2, we explore supervised normal flow estimation while ensuring transferability across domains and datasets. Unlike most OF estimators that use CNNs on event frames, we propose a novel approach based on encoding 3D point sets and a novel loss function. This framework directly trains a network to estimate per-event normal flow from a local event cloud, supervised by ground-truth optical flow. To efficiently handle large event volumes, we adopt VecKM [59], a scalable and descriptive local point cloud encoder. We identify several unique advantages of estimating normal flow with this point-based network.

1. **Temporally and Spatially Sharp Predictions**. By predicting per-event flow based on the Euclidean neighborhood of every event, our method produces sharp predictions, especially for independently moving objects.
2. **Richer data augmentation.** Our point-based approach enables a wider range of augmentations, such as uniform random rotations, which significantly enhances the estimator's accuracy and robustness across various domains.
3. **Uncertainty quantification**. Our point-based method can compute prediction uncertainty with a simple ensemble inference, offering valuable information for downstream tasks like egomotion estimation.
4. **Strong transferability.** The method uses only event neighborhoods and we train on undistorted events in normalized camera coordinates. This improves transferability when training and testing on different datasets.

Finally, we show the effectiveness of the predicted normal flows through aggressive egomotion estimation. We introduce a novel geometric-based egomotion solver that utilizes normal flows in conjunction with IMU measurements. This completes the pipeline integrating transferable event-based normal flow estimation with egomotion estimation. Our contribution can be summarized as follows:

- We show that normal flow can be better estimated by using a special point-based encoder VecKM, as opposed to model-based methods or PointNet. This, and the properties of normal flow, results in more accurate and transferrable image motion estimation compared to learning-based optical flow estimators.
- We introduce a novel geometric-based egomotion solver only using normal flows and IMU measurements, which remains robust under aggressive egomotion scenarios.
- We extensively evaluate our point-based flow estimator on multiple datasets and multiple transfer settings.

2. Related Work

2.1. Event-Based Optical Flow Estimation

In the early stages, **model-based** methods were studied for event-based optical flow estimation. The methodologies include extensions of the Lucas-Kanade algorithm [5, 8], feature matching [27, 28, 45, 58], contrast maximization (CM) [18], Multi-CM [46], plane fitting [3, 9], filter banks [6, 12], time surface matching [4, 10, 50], iterative deblurring [53].

Recently, **learning-based frame-based** methods, such as E-RAFT [21], have dominated event-based optical flow estimation by leveraging correlation volumes. Many techniques, such as multi-modality [48, 49, 60, 66], motion aggregation [22, 26, 35, 57], synthetic datasets [25, 30, 31] have improved these models. However, their performance drops significantly when tested across different domains.

Meanwhile, **spiking neural networks** (SNNs) are applied to event-based optical flow estimation due to their efficiency in processing asynchronous data [15, 23, 37, 55, 62, 64]. SNNs offer energy efficiency and compatibility with neuromorphic hardware. However, their complex training and dependency on specialized hardware make them less practical than conventional neural networks.

Despite the existing research, **learning-based point-**

based methods remain unexplored in the event camera community, mainly due to the challenges of training point-based estimators on dense events. Our work addresses this gap.

2.2. Point-Based Networks for Event Cameras

Point-based networks have enabled direct processing of point cloud data where PointNet [38] was the pioneering work. Subsequently, many feature extractors were developed to improve point cloud processing [32, 39, 40, 52, 61, 63]. All existing point-based networks follow a common pipeline. Input point clouds are sampled and grouped into centered neighborhoods, transforming the data from $(n, 3)$ to $(n, K, 3)$, where K is the number of neighboring points. However, this method is impractical for event data due to the huge values of n and K. Fortunately, with the invention of VecKM [59], a descriptive and scalable local point cloud encoder that eliminates the need for explicit grouping and sampling, it is now feasible to apply point-based networks to event data. We explore the potential in this paper.

Prior to the introduction of VecKM, several attempts were made to apply point-based networks to event-based vision tasks [41, 42, 44, 51, 56]. However, traditional point-based networks require downsampling of the local point cloud neighborhoods (to reduce K), limiting their use to only high-level tasks like action recognition, which do not demand precise modeling of event geometry. No previous work has successfully used point-based networks for low-level tasks such as normal flow prediction, which requires an accurate representation of event geometry.

2.3. Normal Flow and its Applications

Normal flow refers to the component of optical flow that is perpendicular to the edges or parallel to image gradients:

$$\mathbf{n} = -\frac{\nabla I \cdot \mathbf{u}}{||\nabla I||^2} \nabla I \tag{1}$$

where ∇I is the image gradient, $\mathbf{u}$ is the optical flow vector, and $\mathbf{n}$ is the normal flow. *Normal flow can be estimated from a local neighborhood because it depends only on local spatial-temporal intensity changes.* Unlike optical flow, it is not affected by the aperture problem [2], as it only captures motion along the image gradient direction. Because normal flow is a projection of optical flow, it satisfies the following constraint, which will be used extensively in this paper:

$$\mathbf{n} \cdot (\mathbf{u} - \mathbf{n}) = 0 \tag{2}$$

Normal flow is typically estimated by fitting planes to very small event cloud neighborhoods [9, 33, 36, 43], a simple and reliable method across various datasets. However, the approach encounters difficulties when the edge is curved, or the local region features a corner.

Normal flow has been applied to motion and structure estimation [24, 29, 43]. Traditionally, 3D motion has been computed from normal flow via classification approaches [11, 17] using the so-called depth positivity constraint, relating the 2D to the 3D measurements. Researchers recently developed ways to incorporate the depth positivity constraint into optimization frameworks [7], and neural networks [34] to estimate 3D motion.

The egomotion estimation algorithm proposed in this paper builds on the approach of [7], which also relies on the depth positivity constraint and optimization. However, we reformulate the optimization problem as training a support vector classifier, enhancing stability and accuracy.

3. Methodology

3.1. Problem Definition and Overview

Point-Based Normal Flow Prediction. The input to our point-based normal flow estimator is a sequence of events $(\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_N)$, where $\mathbf{e}_k = (t_k, x_k, y_k)$[1]. We assume access to GT per-event optical flow $(\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N)$ for supervision (see Appendix 9 for how to obtain it). The output of the estimator is a sequence of normal flow predictions $(\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2, \cdots, \hat{\mathbf{n}}_N)$, where $\hat{\mathbf{n}}_k \in \mathbb{R}^2$. Each prediction $\hat{\mathbf{n}}_k$ is determined by the centered neighboring events of $\mathbf{e}_k$:

$$\begin{aligned} &\hat{\mathbf{n}}_k = f\big(\mathcal{N}(\mathbf{e}_k)\big), \text{ where } \mathcal{N}(\mathbf{e}_k) := \\ &\Big\{\mathbf{e}_j - \mathbf{e}_k : \Big|\Big|\Big(\frac{t_j - t_k}{\delta t}, \frac{x_j - x_k}{\delta x}, \frac{y_j - y_k}{\delta y}\Big)\Big|\Big|_2 < 1\Big\} \end{aligned} \tag{3}$$

where δt, δx, δy are the hyper-parameters controlling the neighborhood size. Note we allow the number of neighboring events to be different, i.e., $|\mathcal{N}(\mathbf{e}_k)|$ can be different for different events $\mathbf{e}_k$. As discussed in Section 2.3, the neighboring events contain sufficient information to estimate normal flow because it is determined by intensity gradients, which are inherently local.

Uncertainty Quantification (UQ). After predicting the per-event normal flows $(\hat{\mathbf{n}}_k)_{k=1}^N$, we estimate uncertainty scores $(\sigma_k)_{k=1}^N$ of the predictions, where $\sigma_k \geq 0$. As shown later, UQ naturally arises as a byproduct of our point-based estimator, which we found is useful for improving performance in downstreaming tasks like egomotion estimation.

Overview. Our solution to the problem is outlined in Figure 1. Each component of the pipeline is detailed as follows. Section 3.2 introduces the local events encoder, which efficiently transforms the local events $\mathcal{N}(\mathbf{e}_k)$ into a representative vector. Section 3.3 introduces a novel loss function that guides the network to predict normal flows, supervised by optical flow ground-truth. Section 3.4 introduces the data augmentation tricks used during training, which is a unique advantage of our point-based method. Section 3.5 introduces how to compute uncertainty scores during inference

[1] We omit polarity here because we found the polarity does not improve prediction accuracies.

time. Section 3.6 details an egomotion estimation algorithm that uses normal flow inputs, where we enhance the robustness and stability of the algorithm from [7].

3.2. Local Events Encoder

We introduce how to encode the neighboring events $\mathcal{N}(\mathbf{e}_k)$ into a vector representation for each event $\mathbf{e}_k$. Given the high volume of input events($\sim$ 80k events every 20 ms), the local events encoder must be scalable and efficient.

We use a recently developed a local geometry encoder named VecKM [59], which is designed to process large point clouds efficiently. VecKM models local events as samples from a kernel mixture and uses random Fourier features to transform the kernel mixture into a vector representation. Such formulation allows using all neighboring points to compute the local geometric encoding without downsampling the neighborhood. Besides, it eliminates the need for explicitly grouping and sampling the event neighborhoods. These merits make VecKM well-suited for handling high-volume event data and dense local event regions.

VecKM Encoding of Local Events. Given the normalized events $X_{N\times 3} = \{(\frac{t_k}{\delta t}, \frac{x_k}{\delta x}, \frac{y_k}{\delta y})\}_{k=1}^{N}$ in Eqn. (3), the VecKM local events encoding $G_{N\times d}$ is computed by:

$$\begin{aligned} J_{N\times N} &= adjacency\_matrix(X_{N\times 3}) \\ \mathcal{A}_{N\times d} &= \exp(iX_{N\times 3}A_{3\times d}) \\ G_{N\times d} &= normalize\big((J_{N\times N}\mathcal{A}_{N\times d})./\mathcal{A}_{N\times d}\big) \end{aligned} \quad (4)$$

J is a sparse adjacency matrix where the (j,k)-entry is 1 if $\mathbf{e}_j$ and $\mathbf{e}_k$ are close, and 0 otherwise. $A_{3\times d}$ is a randomized fixed matrix with entries drawn from a normal distribution with a mean of 0 and variance of 25. The function $\exp(i\cdot)$ is the element-wise Euler formula, and ./ denotes the element-wise complex number division.

The neighbor information is captured in the sparse adjacency matrix J and incorporated into the local event encoding G through matrix multiplication. The centralizing step is implicitly achieved by the element-wise division. This formulation eliminates the need for explicitly grouping and sampling event neighborhoods, making it descriptive and scalable. Kindly refer to [59] for detailed derivation.

Qualitative Evaluation of VecKM Encoding. VecKM [59] proves that $G[k,:] \in \mathbb{C}^d$ in Eqn. (4) effectively represents local events $\mathcal{N}(\mathbf{e}_k)$. Specifically, the local events distribution can be reconstructed from the encoding $G[k,:]$. Figure 3 shows examples of local events and the distribution reconstruction from the local events encoding. The strong alignment between the events and the reconstructed distributions suggests that VecKM generates a representative encoding.

Normal Flow Prediction. After computing the local events encoding $G_{N\times d}$ from Eqn. (4), we transform each event's encoding $G[k,:]$ to normal flow prediction $\hat{\mathbf{n}}_k$ using a multi-layer perceptron (MLP) [47]. The MLP is trained by a novel loss function introduced in the next section.

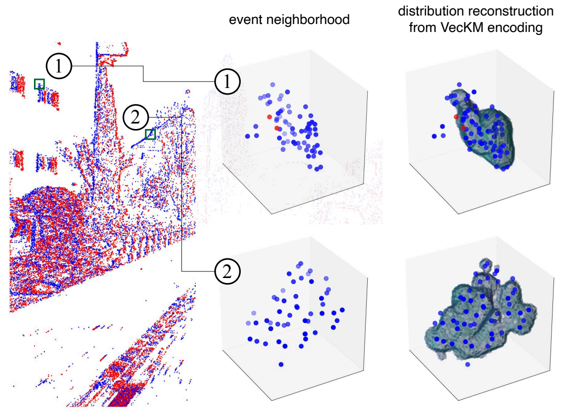

Figure 3. VecKM allows us to inspect encoding quality by visualizing the events distribution reconstruction (shown in gray) from VecKM's local events encoding. The reconstructed 3D distribution closely aligns with the original (blue and red) events, demonstrating that VecKM's encoding effectively represents the event data.

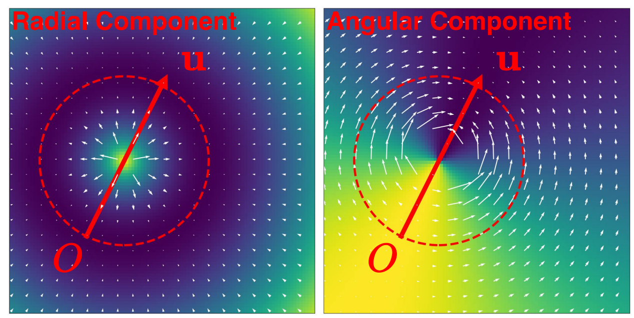

Figure 4. Loss maps and gradient fields of the motion field loss function. Our motion field loss function consists of radial and angular components. Given the GT optical flow $\mathbf{u}$, the radial component guides the predicted flow to lie on the circle with $\mathbf{u}$ as the diameter. The angular component guides the predicted flow to align with $\mathbf{u}$, which prevents the trivial prediction of zero flow.

3.3. Normal Flow Learning with a Two-Term Loss

In frame-based vision, *normal flow* has been defined as the flow component along the image gradient (or perpendicular to the local edge). In event space, however, no obvious definition exists for an image gradient that can be computed per-event and so obtaining ground truth (GT) normal flow for an event camera is challenging. Instead, we use GT optical flow as supervision, and guide the network to predict normal flow (the component of optical flow along some direction) by training with a novel motion field loss function. This loss function is composed of two components: a radial and an angular component, as illustrated in Figure 4.

The **radial component** guides the predicted flow to satisfy the normal flow magnitude constraint (2). Geometrically, this means the predicted flow lies on a circle where the GT optical flow represents the diameter. When the normal flow constraint is satisfied, the radial loss becomes zero.

Note that predicting zero flow is a trivial solution for minimizing the radial component. To avoid such prediction, we introduce an **angular component**, which encourages the predicted flow to have the same direction as the optical flow.

Heuristically, training on the sum of these two components allows the network to predict the full optical flow when possible. If the texture information is insufficient to predict the full optical flow, the network resorts to predicting the normal flow. We show that our intention is achieved in Figure 7 in Appendix 11.1.

Mathematically, given the GT optical flow, the normal flow is predicted using the following loss function:

$$Radial(\mathbf{u}, \hat{\mathbf{n}}) = \log\left(\frac{\epsilon + ||\hat{\mathbf{n}} - \mathbf{u}/2||}{\epsilon + ||\mathbf{u}/2||}\right)^2 \quad (5)$$

$$Angular(\mathbf{u}, \hat{\mathbf{n}}) = -\frac{(\hat{\mathbf{n}} - \mathbf{u}/2) \cdot \mathbf{u}}{||\hat{\mathbf{n}} - \mathbf{u}/2|| \cdot ||\mathbf{u}||} \quad (6)$$

Figure 4 shows the resulting loss map and gradient field. Note, that the two components gradients are orthogonal everywhere, so the optimization of one component does not interfere with the other.

3.4. Improve Transferability by Data Augmentation

Diverse data augmentation is a unique advantage of our point-based network, which improves the transferability of the estimator. Below, we enumerate the data augmentation tricks applied during training. We denote the input events as $X_{N\times 3}$ and their corresponding GT optical flow as $U_{N\times 2}$.

Random Rotation. Motion field estimation is an equivariant task, meaning that rotating the events on the image plane causes the motion field to rotate by the same angle. Thanks to this, we apply a uniformly sampled random rotation angle $\theta \in [0, 2\pi)$ to the event cloud and GT flow. The augmented inputs and targets are obtained by:

$$\begin{aligned} X_{N\times 3} &\leftarrow X_{N\times 3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \\ U_{N\times 2} &\leftarrow U_{N\times 2} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \end{aligned} \quad (7)$$

This forces the network to estimate direction without bias.

Random Scaling. We scale the event streams by a random scalar $\alpha \in (0.75, 1.25)$. The augmented inputs and targets are obtained by: $X \leftarrow X * \alpha, U \leftarrow U$. This improves the estimator's performance on small and large objects.

Random Sampling. We sample $50\% \sim 100\%$ percentage of the events and their corresponding flows. This makes the estimator robust to density variation.

3.5. Inference with Uncertainty Quantification

As explained in Section 3.4 **Random Rotation**, motion field estimation is an equivariant task, which we exploit to estimate an uncertainty score based on how well the predicted flow remains equivariant to rotation. Specifically, we sample K rotation angles and infer the normal flow with the rotated events. Then we de-rotate the predicted normal flow with the corresponding rotation angle:

$$\hat{U} = estimator\left(X_{N\times 3}\begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & R(\theta) \end{bmatrix}\right)R(\theta)^{-1} \quad (8)$$

By doing so, we obtain an ensemble of K predicted flows for each event, and can test their consensus to compute an uncertainty score. Specifically, we use the circular standard deviation [1] of the ensemble as the uncertainty score. The final prediction is given by the average of the ensemble in polar coordinates. Predictions corresponding with uncertainty above a threshold are discarded.

3.6. Egomotion Estimation from Normal Flow

After obtaining the normal flow predictions and uncertainty scores, we discard events with uncertainty higher than a threshold. We then use the normalized event coordinates and their flow predictions to estimate camera translation, assuming an initial rotational velocity estimate from the IMU sensor that is usually available in many scenarios.

Our egomotion solver uses the depth positivity constraint, which states that all world points are in front of the camera, i.e., have positive depth. Existing methods [7, 17, 34] utilize this constraint by maximizing the negative depths. However, the solution to this problem is not unique with respect to normal flow because depth positivity only considers the sign (or the direction) of the normal flow and is thus a weak constraint. So, when estimated normal flow is used, errors in the estimated normal flow direction determine the output. Thus we reformulate the problem as maximum margin problem solved by training a support vector classifier, which results in a robust estimate.

Given an estimate of the angular velocity Ω from the IMU, we can calculate the rotational component $\mathbf{u}_{\mathbf{x}}^{rot} = B_{\mathbf{x}}\Omega$ of the optical flow at pixel $\mathbf{x} = (x, y)$ where $B_{\mathbf{x}}$ is:

$$B_{\mathbf{x}} = \begin{bmatrix} xy & -(x^2+1) & y \\ (y^2+1) & -xy & -x \end{bmatrix} \quad (9)$$

Then the magnitude of the derotated normal flow at pixel $\mathbf{x}$, called $n_{\mathbf{x}}$, can be calculated through the relation $n_{\mathbf{x}} = \|\hat{\mathbf{n}}_{\mathbf{x}}\| - \mathbf{g}_{\mathbf{x}}^T \mathbf{u}_{\mathbf{x}}^{rot}$ where $\mathbf{g}_{\mathbf{x}} \in \mathbb{R}^{2\times 1}$ is the direction of normal flow with unit norm. Then $n_{\mathbf{x}}$ is related to translation by:

$$n_{\mathbf{x}} = \frac{1}{Z_{\mathbf{x}}}(\mathbf{g}_{\mathbf{x}}^T A_{\mathbf{x}})V \quad (10)$$

Where $V \in \mathbb{R}^{3\times 1}$ denotes the direction of translation, $Z_{\mathbf{x}} \in \mathbb{R}$ denotes depth, and $A_{\mathbf{x}}$ is:

$$A_{\mathbf{x}} = \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \quad (11)$$

Since the depth $Z_{\mathbf{x}}$ is positive, the following product, denoted as $\rho_{\mathbf{x}}(V)$, should be positive for all $\mathbf{x}$:

$$\rho_{\mathbf{x}}(V) = n_{\mathbf{x}}(\mathbf{g}_{\mathbf{x}}^T A_{\mathbf{x}} V) > 0 \tag{12}$$

We impose this constraint by formulating Eqn. (12) into a linear support vector classification without intercept, namely solving V such that $(\mathbf{g}_{\mathbf{x}}^T A_{\mathbf{x}})V$ has a target sign decided by $n_{\mathbf{x}}$. The pseudo-code is given below. p is the number of events. A, B are obtained through Eqn. (9) and (11), N are the norms of the predicted normal flows, G are the directions of the predicted normal flows with unit lengths, and Ω_0 is the rotational estimate from the IMU sensor.

Algorithm 1 Egomotion Solver

```
Input: A_{p×2×3}, B_{p×2×3}, N_{p×1}, G_{p×2}, Ω_0 ∈ R^{3×1}.
Output: V ∈ R^{3×1}.
Q_{p×3} = batch_mat_mul(G, A)
R_{p×1} = N − batch_mat_mul(G, B) × Ω_0
Q_{2p×3} = concat(Q, −Q)
R_{2p×1} = sign(concat(R, −R))
svm = LinearSVM(fit_intercept=False)
V_{3×1} = svm.fit(Q, R).coef_
V = V / ||V||
```

4. Experiments on Normal Flow Estimation

Datasets. We use MVSEC [65], EVIMO2 [13], and DSEC [20] to evaluate the accuracy and transferability of our normal estimator. We undistort and transform the events to normalized camera coordinates (focal length one). We train on each dataset and evaluate the model on all three datasets, resulting in nine combinations. Additionally, we train our model on the union of the three (and potentially more) datasets. We evaluate it and release it for public use.

Note that MVSEC has different camera resolution and distortion compared to EVIMO2 and DSEC. Therefore, there is a substantial domain gap among the cameras used in the three datasets.

Evaluation Metrics. Quantitatively, we use projection endpoint error (PEE) and percentage of sign correctness ($\%Pos$), following the convention in [34], to evaluate the accuracy of normal flow predictions.

$$PEE(\mathbf{u}, \hat{\mathbf{n}}) = \left|\left|\frac{\mathbf{u}\cdot\hat{\mathbf{n}}}{||\hat{\mathbf{n}}||} - ||\hat{\mathbf{n}}||\right|\right| \tag{13}$$

$$\%Pos(\mathbf{u}, \hat{\mathbf{n}}) = percentage(\mathbf{u}\cdot\hat{\mathbf{n}} > 0) \tag{14}$$

PEE measures how well the normal flow magnitude constraint (2) is satisfied. It measures the error in the length of the normal flow. $\%Pos$ measures how well the predicted normal flow has correct orientation as determined by the GT optical flow. Note that since the direction of normal flow is independent of motion, optical flow also satisfies the constraint equation (2) and thus also minimizes PEE.

Qualitative Evaluation. In this link, we include flow prediction videos for every evaluation. Qualitative evaluation is crucial for assessing aspects like prediction sharpness and handling independently moving objects, which are not fully captured by quantitative metrics. We show visualizations in Figure 5 and upload flow prediction videos to this link. See Appendix 8 for detailed video descriptions.

Compared Models. We compare our point-based learning-based normal flow estimator with state-of-the-art event-based optical flow estimators. We compare against MultiCM [46], E-RAFT [21], and Taming Contrast Maximization (TCM) [35], which are all frame-based estimators using model-based, supervised, and self-supervised learning approaches. Additionally, we compare against two point-based normal flow estimators: PCA [9] and PointNet [38].

Implementation details and hyper-parameter setting are presented in Appendix 10.

4.1. Evaluation on MVSEC

As shown by the quantitative results in Table 1, **our point-based method is on par with state-of-the-art frame-based methods despite using only local information.** While TCM achieves lower PEE than our method on MVSEC, this is because MVSEC mainly features low-resolution, slow-moving, static scenes without independently moving objects. This produces sparse events and limits the performance of our estimator. In these scenarios, TCM and E-RAFT using CNNs, can leverage larger receptive fields (through stacked convolution and pooling layers) to smooth the flow predictions. As we will show in Section 4.2, TCM and E-RAFT's performance deteriorates when scenes include independently moving objects, whereas our method remains robust.

Our method outperforms the point-based estimators, PCA (MB) and PointNet (SL) by achieving lower PEE and higher $\%Pos$, highlighting the benefits of supervised training. Additionally, PointNet requires explicit grouping of events into neighborhoods and sampling, while our method does not. This allows us to use information from more neighboring events to predict the flows and yield better performance. In cross-dataset evaluations, our estimator outperforms PCA and PointNet when trained on DSEC and evaluated on MVSEC even though PointNet was trained on MVSEC. When our method is trained on EVIMO-`imo`, the performance remains comparable though the EVIMO-`imo` train set containing only 2.74 minutes of data.

4.2. Evaluation on EVIMO2

EVIMO2-`imo` sequences feature fast independently moving objects. Table 2 presents quantitative results on the samsung_mono camera and Figure 5 visualizes the flow predic-

		Input	Training Set	Indoor Flying 1		Indoor Flying 2		Indoor Flying 3		Outdoor Day 1		Average	
				PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑
MultiCM	MB	F	-	0.993	97.8%	1.378	98.1%	1.191	98.2%	1.422	93.2%	1.246	96.8%
PCA	MB	P	-	1.460	76.2%	1.586	76.3%	1.552	75.0%	1.548	79.9%	1.537	76.9%
E-RAFT	SL	F	M	1.836	76.1%	2.867	74.7%	1.951	75.4%	**0.677**	97.8%	1.833	81.0%
			D	0.797	92.5%	1.163	91.6%	0.948	92.9%	0.846	97.0%	0.939	93.5%
TCM	SSL	F	M	0.319	95.1%	0.638	91.3%	0.490	94.4%	0.948	97.3%	0.599	94.5%
			D	**0.303**	93.9%	**0.546**	92.0%	**0.437**	93.6%	0.778	97.1%	**0.516**	94.1%
PointNet	SL	P	M	0.973	96.7%	1.428	95.6%	1.224	97.0%	1.032	97.1%	1.164	96.6%
Ours	SL	P	M	0.970	98.4%	1.090	99.4%	1.040	99.6%	0.880	**99.2%**	0.995	**99.2%**
			D	0.922	99.3%	1.216	99.2%	1.282	99.6%	1.004	97.8%	1.106	99.0%
			E	1.669	99.0%	1.483	**99.6%**	1.671	99.7%	0.854	98.6%	1.419	**99.2%**
			M+D+E	0.968	**99.5%**	1.057	99.5%	1.065	**99.8%**	0.879	97.3%	0.992	99.0%

Table 1. Quantitative results on MVSEC. The estimators are classified into model-based (MB), supervised learning (SL), self-supervised learning (SSL), frame-based (F), point-based (P). They are trained on MVSEC (M), DSEC (D), EVIMO-`imo` (E).

		Input	Training Set	Scene_13_00		Scene_13_05		Scene_14_03		Scene_14_04		Scene_14_05		Average (8 scenes)	
				PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑	PEE ↓	% Pos ↑
MultiCM	MB	F	-	1.509	53.2%	4.315	75.7%	1.611	79.2%	1.800	73.2%	2.768	72.9%	1.800	68.5%
PCA	MB	P	-	1.573	88.2%	2.035	87.5%	1.580	91.9%	1.784	90.3%	1.823	89.4%	1.712	87.8%
E-RAFT	SL	F	M	1.370	71.9%	2.406	90.6%	1.356	69.5%	1.458	64.6%	2.186	67.1%	1.470	70.2%
			D	0.843	88.9%	1.185	97.5%	0.517	88.1%	0.538	85.9%	0.908	86.3%	0.705	87.9%
TCM	SSL	F	M	0.823	85.6%	3.201	95.3%	1.111	86.3%	1.532	86.0%	2.445	82.2%	1.383	84.6%
			D	0.774	87.3%	2.541	95.1%	0.872	87.8%	1.090	86.5%	1.640	84.1%	1.105	85.3%
PointNet	SL	P	E	1.047	88.1%	0.924	97.7%	0.848	98.3%	0.892	96.2%	1.053	96.6%	0.933	95.0%
Ours	SL	P	M	0.713	95.6%	0.269	99.3%	0.676	98.8%	0.651	98.1%	0.806	98.2%	0.551	97.1%
			D	0.590	96.6%	**0.230**	**99.8%**	0.575	**99.8%**	0.625	**99.5%**	**0.567**	**99.4%**	0.463	**97.9%**
			E	0.497	**96.7%**	0.399	99.2%	**0.478**	99.2%	0.515	98.8%	0.584	98.6%	0.423	97.9%
			M+D+E	**0.465**	96.2%	0.308	99.2%	0.544	99.3%	**0.467**	98.8%	0.568	98.5%	**0.396**	97.8%

Table 2. Quantitative results on the first five scenes of EVIMO2-`imo`'s evaluation set. The complete table is in Appendix 12.

tion colored in HSV. As shown on the Table 2, **our method, significantly outperforms all compared estimators in the presence of independently moving objects even when trained on other datasets.** In addition, as shown in Figure 5, our method effectively preserves the boundaries of the IMOs, closely matching the ground truth. While baseline methods achieve fair scores in Table 2, their predictions are often wrong in key scenarios. As shown in Figure 5, background flow estimates are often influenced by moving objects, and sudden camera acceleration causes flow estimates to be biased. Although these moments are short in the testing set and hence not fully reflected in Table 2, they represent significant limitations of the baseline methods.

Our estimator shows highly consistent performance and strong generalizability across different scenes, motion types and datasets as shown in Table 2. This is due to the model's use of geometric features from local event neighborhoods, which capture domain-invariant patterns essential for normal flow estimation. Besides, our approach employs normalized camera coordinates, utilizes extensive data augmentation, and prevents global information from affecting estimates in regions with independently moving objects (IMOs), which all lead to the strong performance.

4.3. Evaluation on DSEC

Since DSEC does not release the ground-truth optical flow for the testset, we (only) use 1 scene in the training set to train our model and hold out 12 scenes for quantitative evaluation. As presented in Table 3, our method achieves the same performance when trained on MVSEC, EVIMO, or a single sequence from DSEC, which demonstrates its ability to be trained with a small amount of data while still achieving good performance. The MVSEC-model incurs a slight performance drop due to a relatively large difference in distortion and event density. In the night scenes, events flickering produces many noise events that are irrelevant to the object motion. Therefore, the performance drops compared to day scenes, but our uncertainty score can filter out those noise events to some extent, as discussed in Appendix 13.

training set	8 day scenes		4 night scenes	
	PEE	%Pos	PEE	%Pos
M	1.631	89.0%	2.145	86.0%
E	1.075	90.8%	1.598	81.0%
D (1 scene)	1.043	90.7%	1.357	82.8%

Table 3. Average performance over 12 scenes on DSEC. Scenes splitting and per-scene evaluation are presented in Appendix 13.

4.4. Qualitative Evaluation on More Datasets

We evaluate our estimator qualitatively on DSEC-testset [20], FPV [16], and VECtor [19], which lack ground-truth optical flow. Figure 10 in Appendix 8 and the flow prediction videos in appendix show **our method performs consistently well across datasets from different cameras**.

5. Experiments on Egomotion Estimation

We evaluate our egomotion solver on the EVIMO2 `sfm` split, which includes scenes with fast camera motions and a focus of expansion far from the camera frame. The normal

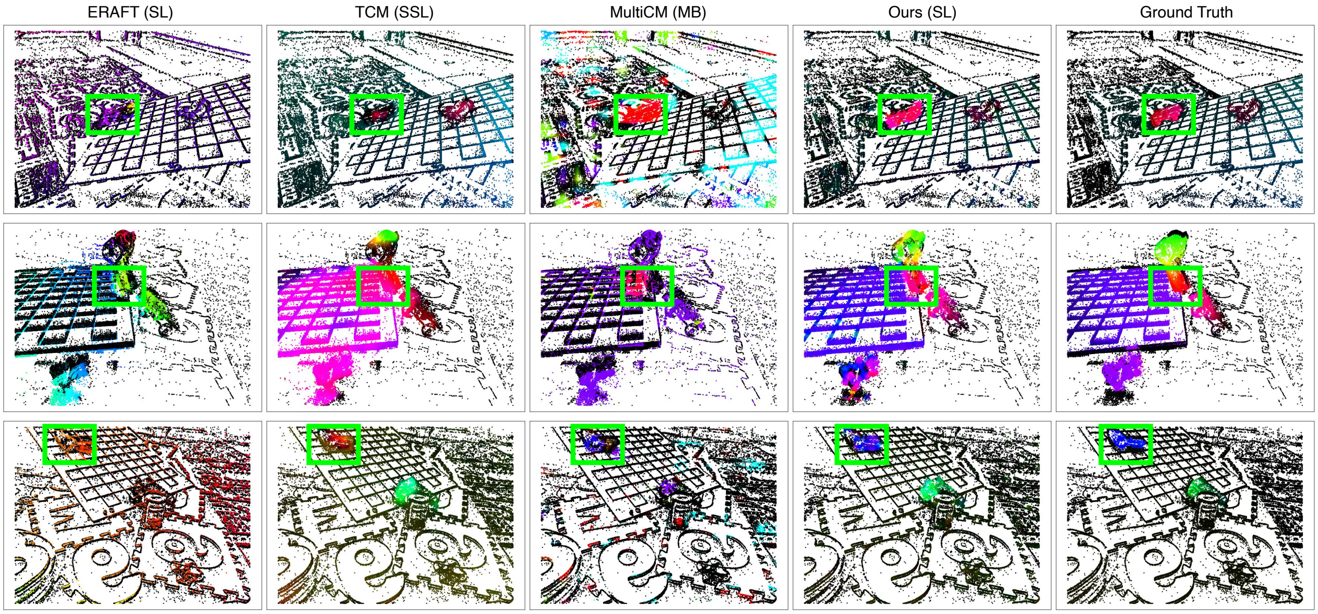

Figure 5. Visualization of flow prediction on EVIMO2-`imo`. The flows are displayed in HSV color space. Our method effectively preserves the boundary of the IMOs, while other methods fail.

flow is estimated using our method trained on the EVIMO2-`imo` training split. Only events with uncertainty scores $\sigma_k < 0.15$ are inputted to egomotion solver. Table 4 compares our solver using SVM, to a method based on [7] that estimates translation by maximizing negative depth using rotation estimates from the IMU. The results demonstrate that our normal flow predictions and egomotion solver yield more accurate egomotion estimations. Figure 11 plots the estimated translation given by the two compared methods, scaled to m/s using ground truth.

Seq.	sfm 03_04_0	sfm 03_02_0	sfm 03_02_1	sfm 03_00_0	sfm 03_03_0
Method			RMS V (m/s)		
Depth Pos.	0.217	0.500	0.243	0.237	0.135
SVM	**0.136**	**0.227**	**0.156**	**0.141**	**0.116**
Seq.	sfm 03_01_0	sfm 03_02_2	sfm 03_03_2	sfm 03_02_3	sfm 03_03_1
Method			RMS V (m/s)		
Depth Pos.	0.369	**0.091**	0.137	**0.213**	0.265
SVM	**0.219**	0.127	**0.108**	0.279	**0.136**

Table 4. Egomotion estimation error on EVIMO2-`sfm`.

6. Ablation Studies

We perform extensive ablation studies on our proposed method. We summarize the conclusion in this section, leaving the experiment statistics to Appendix 11.

Effectiveness of motion field loss. We compare the model's performance against a baseline trained with conventional optical flow loss function. The model trained with our proposed motion field loss achieves significantly lower error, highlighting the effectiveness of our loss function.

Effectiveness of uncertainty quantification (UQ). We find the flow prediction errors are positively correlated with the uncertainty scores. We also find the egomotion estimation error is minimized by choosing a suitable threshold. Both findings highlight the effectiveness of the UQ. Finally, we study how many ensembles of predictions are needed to generate reliable UQ.

Computation Cost. Our estimator has a moderate computation cost. Our runtime is slightly slower than frame-based methods like TCM [35] with memory usage below 6GB. The computation is generally feasible on entry-level GPUs.

7. Conclusion and Future Work

We introduce a point-based method for normal flow estimation that overcomes limitations in existing model and learning based methods by using local information without explicit grouping. This allows improved cross domain transfer and good performance in the presence of independently moving objects. Despite highly scalable, the method may face computational challenges with higher camera resolutions. Thus, future work can consider optimizing the encoding and transformation to normal flow. Additionally, our method currently depends on ground-truth flow for training. A self-supervised approach could further improve its usability. Finally, while our method benefits from using local information, future work can consider careful incorporation of global information that may be beneficial for tasks such as action recognition and object detection.

References

[1] scipy.stats.circstd. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.circstd.html. Accessed: 2024-11-07. 5

[2] The aperture problem. https://elvers.us/perception/aperture. Accessed: 2024-09-24. 1, 3

[3] Himanshu Akolkar, Sio-Hoi Ieng, and Ryad Benosman. Real-time high speed motion prediction using fast aperture-robust event-driven visual flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):361–372, 2020. 2

[4] Mohammed Almatrafi, Raymond Baldwin, Kiyoharu Aizawa, and Keigo Hirakawa. Distance surface for event-based optical flow. *IEEE transactions on pattern analysis and machine intelligence*, 42(7):1547–1556, 2020. 2

[5] Francisco Barranco, Cornelia Fermüller, and Yiannis Aloimonos. Contour motion estimation for asynchronous event-driven cameras. *Proceedings of the IEEE*, 102(10):1537–1556, 2014. 2

[6] Francisco Barranco, Cornelia Fermüller, and Yiannis Aloimonos. Bio-inspired motion estimation with event-driven sensors. In *Advances in Computational Intelligence: 13th International Work-Conference on Artificial Neural Networks, IWANN 2015, Palma de Mallorca, Spain, June 10-12, 2015. Proceedings, Part I 13*, pages 309–321. Springer, 2015. 2

[7] Francisco Barranco, Cornelia Fermüller, Yiannis Aloimonos, and Eduardo Ros. Joint direct estimation of 3d geometry and 3d motion using spatio temporal gradients. *Pattern Recognition*, 113:107759, 2021. 3, 4, 5, 8

[8] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012. 2

[9] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2):407–417, 2013. 2, 3, 6

[10] Vincent Brebion, Julien Moreau, and Franck Davoine. Real-time optical flow for vehicular perception with low-and high-resolution event cameras. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15066–15078, 2021. 2

[11] Tomáš Brodskỳ, Cornelia Fermüller, and Yiannis Aloimonos. Structure from motion: Beyond the epipolar constraint. *International Journal of Computer Vision*, 37:231–258, 2000. 3

[12] Tobias Brosch, Stephan Tschechne, and Heiko Neumann. On event-based optical flow detection. *Frontiers in neuroscience*, 9:137, 2015. 2

[13] Levi Burner, Anton Mitrokhin, Cornelia Fermüller, and Yiannis Aloimonos. Evimo2: an event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. *arXiv preprint arXiv:2205.03467*, 2022. 6

[14] Jingxi Chen, Brandon Y Feng, Haoming Cai, Mingyang Xie, Christopher Metzler, Cornelia Fermuller, and Yiannis Aloimonos. Timerewind: Rewinding time with image-and-events video diffusion. *arXiv preprint arXiv:2403.13800*, 2024. 1

[15] Javier Cuadrado, Ulysse Rançon, Benoit R Cottereau, Francisco Barranco, and Timothée Masquelier. Optical flow estimation from event-based cameras and spiking neural networks. *Frontiers in Neuroscience*, 17:1160034, 2023. 2

[16] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019. 7

[17] Cornelia Fermüller. Passive navigation as a pattern recognition problem. *International journal of computer vision*, 14(2):147–158, 1995. 3, 5

[18] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3867–3876, 2018. 2

[19] Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. Vector: A versatile event-centric benchmark for multi-sensor slam. *IEEE Robotics and Automation Letters*, 7(3):8217–8224, 2022. 7

[20] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *International Conference on 3D Vision (3DV)*, 2021. 6, 7

[21] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *2021 International Conference on 3D Vision (3DV)*, pages 197–206. IEEE, 2021. 1, 2, 6

[22] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. Dense continuous-time optical flow from event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[23] Jesse Hagenaars, Federico Paredes-Valles, and Guido de Croon. Self-supervised learning of event-based optical flow with spiking neural networks. In *Advances in Neural Information Processing Systems*, pages 7167–7179. Curran Associates, Inc., 2021. 2

[24] Jinghang Li, Bangyan Liao, Xiuyuan Lu, Peidong Liu, Shaojie Shen, and Yi Zhou. Event-aided time-to-collision estimation for autonomous driving. *arXiv preprint arXiv:2407.07324*, 2024. 2, 3

[25] Yijin Li, Zhaoyang Huang, Shuo Chen, Xiaoyu Shi, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Blinkflow: A dataset to push the limits of event-based optical flow estimation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3881–3888. IEEE, 2023. 1, 2

[26] Haotian Liu, Guang Chen, Sanqing Qu, Yanping Zhang, Zhijun Li, Alois Knoll, and Changjun Jiang. Tma: Temporal motion aggregation for event-based optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9685–9694, 2023. 1, 2

[27] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. BMVC, 2018. 2

[28] Min Liu and Tobi Delbruck. Edflow: Event driven optical flow camera with keypoint detection and adaptive block matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(9):5776–5789, 2022. 2

[29] Xiuyuan Lu, Yi Zhou, and Shaojie Shen. Event-based visual inertial velometer. *arXiv preprint arXiv:2311.18189*, 2023. 2, 3

[30] Xinglong Luo, Kunming Luo, Ao Luo, Zhengning Wang, Ping Tan, and Shuaicheng Liu. Learning optical flow from event camera with rendered dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9847–9857, 2023. 2

[31] Xinglong Luo, Ao Luo, Zhengning Wang, Chunyu Lin, Bing Zeng, and Shuaicheng Liu. Efficient meshflow and optical flow estimation from event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19198–19207, 2024. 2

[32] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022. 3

[33] Elias Mueggler, Christian Forster, Nathan Baumli, Guillermo Gallego, and Davide Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *2015 IEEE international conference on Robotics and Automation (ICRA)*, pages 4874–4881. IEEE, 2015. 2, 3

[34] Chethan M Parameshwara, Gokul Hari, Cornelia Fermüller, Nitin J Sanket, and Yiannis Aloimonos. Diffposenet: Direct differentiable camera pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6845–6854, 2022. 3, 5, 6

[35] Federico Paredes-Vallés, Kirk YW Scheper, Christophe De Wagter, and Guido CHE De Croon. Taming contrast maximization for learning sequential, low-latency, event-based optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9695–9705, 2023. 2, 6, 8

[36] Bas J Pijnacker Hordijk, Kirk YW Scheper, and Guido CHE De Croon. Vertical landing for micro air vehicles using event-based optical flow. *Journal of Field Robotics*, 35(1): 69–90, 2018. 3

[37] Wachirawit Ponghiran, Chamika Mihiranga Liyanagedera, and Kaushik Roy. Event-based temporally dense optical flow estimation with sequential learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9827–9836, 2023. 2

[38] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3, 6

[39] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[40] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022. 3

[41] Hongwei Ren, Yue Zhou, Yulong Huang, Haotian Fu, Xiaopeng Lin, Jie Song, and Bojun Cheng. Spikepoint: An efficient point-based spiking neural network for event cameras action recognition. *arXiv preprint arXiv:2310.07189*, 2023. 3

[42] Hongwei Ren, Yue Zhou, Jiadong Zhu, Haotian Fu, Yulong Huang, Xiaopeng Lin, Yuetong Fang, Fei Ma, Hao Yu, and Bojun Cheng. Rethinking efficient and effective point-based networks for event camera classification and regression: Eventmamba. *arXiv preprint arXiv:2405.06116*, 2024. 3

[43] Zhongyang Ren, Bangyan Liao, Delei Kong, Jinghang Li, Peidong Liu, Laurent Kneip, Guillermo Gallego, and Yi Zhou. Motion and structure from event-based normal flow. *arXiv preprint arXiv:2407.12239*, 2024. 2, 3

[44] Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. Eventnet: Asynchronous recursive event processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3887–3896, 2019. 3

[45] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Fast event-based optical flow estimation by triplet matching. *IEEE Signal Processing Letters*, 29:2712–2716, 2022. 2

[46] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow. In *European Conference on Computer Vision*, pages 628–645. Springer, 2022. 1, 2, 6

[47] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. *arXiv preprint arXiv:1705.09792*, 2017. 4

[48] Zhexiong Wan, Yuchao Dai, and Yuxin Mao. Learning dense and continuous optical flow from an event camera. *IEEE Transactions on Image Processing*, 31:7237–7251, 2022. 2

[49] Zhexiong Wan, Yuxin Mao, Jing Zhang, and Yuchao Dai. Rpeflow: Multimodal fusion of rgb-pointcloud-event for joint optical flow and scene flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10030–10040, 2023. 2

[50] Zengyu Wan, Yang Wang, Zhai Wei, Ganchao Tan, Yang Cao, and Zheng-Jun Zha. Event-based optical flow via transforming into motion-dependent view. *IEEE Transactions on Image Processing*, 2024. 2

[51] Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Space-time event clouds for gesture recognition: From rgb cameras to event cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1826–1835. IEEE, 2019. 3

[52] Yue Wu, Jiaming Liu, Maoguo Gong, Zhixiao Liu, Qiguang Miao, and Wenping Ma. Mpct: Multiscale point cloud transformer with a residual network. *IEEE Transactions on Multimedia*, 2023. 3

[53] Yilun Wu, Federico Paredes-Vallés, and Guido CHE De Croon. Lightweight event-based optical flow estimation via iterative deblurring. In *2024 IEEE International Con-*

ference on Robotics and Automation (ICRA)*, pages 14708–14715. IEEE, 2024. 2

[54] Tianyi Xiong, Jiayi Wu, Botao He, Cornelia Fermuller, Yiannis Aloimonos, Heng Huang, and Christopher A Metzler. Event3dgs: Event-based 3d gaussian splatting for fast egomotion. *arXiv preprint arXiv:2406.02972*, 2024. 1

[55] Yingfu Xu, Guangzhi Tang, Amirreza Yousefzadeh, Guido de Croon, and Manolis Sifalakis. Event-based optical flow on neuromorphic processor: Ann vs. snn comparison based on activation sparsification. *arXiv preprint arXiv:2407.20421*, 2024. 2

[56] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3323–3332, 2019. 3

[57] Yaozu Ye, Hao Shi, Kailun Yang, Ze Wang, Xiaoting Yin, Yining Lin, Mao Liu, Yaonan Wang, and Kaiwei Wang. Towards anytime optical flow estimation with event cameras. *arXiv preprint arXiv:2307.05033*, 2023. 2

[58] Hongzhi You, Yijun Cao, Wei Yuan, Fanjun Wang, Ning Qiao, and Yongjie Li. Vector-symbolic architecture for event-based optical flow. *arXiv preprint arXiv:2405.08300*, 2024. 2

[59] Dehao Yuan, Cornelia Fermuller, Tahseen Rabbani, Furong Huang, and Yiannis Aloimonos. A linear time and space local point cloud geometry encoder via vectorized kernel mixture (VecKM). In *Proceedings of the 41st International Conference on Machine Learning*, pages 57871–57886. PMLR, 2024. 2, 3, 4

[60] Chi Zhang, Chenxu Jiang, and Lei Yu. Cross-modal learning for optical flow estimation with events. *Signal Processing*, page 109580, 2024. 2

[61] Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Starting from non-parametric networks for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2023. 3

[62] Yisa Zhang, Hengyi Lv, Yuchen Zhao, Yang Feng, Hailong Liu, and Guoling Bi. Event-based optical flow estimation with spatio-temporal backpropagation trained spiking neural network. *Micromachines*, 14(1):203, 2023. 2

[63] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 3

[64] Rui Zhao, Ruiqin Xiong, Jing Zhao, Zhaofei Yu, Xiaopeng Fan, and Tiejun Huang. Learning optical flow from continuous spike streams. *Advances in Neural Information Processing Systems*, 35:7905–7920, 2022. 2

[65] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3 (3):2032–2039, 2018. 6

[66] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018. 2

[67] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019. 1