

Real-time Motion Segmentation with Event-based Normal Flow

Sheng Zhong*, Zhongyang Ren*, Xiya Zhu, Dehao Yuan, Cornelia Fermüller, Yi Zhou

Abstract—Event-based cameras are bio-inspired sensors with pixels that independently and asynchronously respond to brightness changes at microsecond resolution, offering the potential to handle visual tasks in challenging scenarios. However, due to the sparse information content in individual events, directly processing the raw event data to solve vision tasks is highly inefficient, which severely limits the applicability of state-of-the-art methods in real-time tasks, such as motion segmentation—a fundamental task for dynamic scene understanding. Incorporating normal flow as an intermediate representation to compress motion information from event clusters within a localized region provides a more effective solution. In this work, we propose a normal flow-based motion segmentation framework for event-based vision. Leveraging the dense normal flow directly learned from event neighborhoods as input, we formulate the motion segmentation task as an energy minimization problem solved via graph cuts, and optimize it iteratively with normal flow clustering and motion model fitting. By using a normal flow-based motion model initialization and fitting method, the proposed system is able to efficiently estimate the motion models of independently moving objects with only a limited number of candidate models, which significantly reduces the computational complexity and ensures real-time performance, achieving nearly a $800\times$ speedup in comparison to the open-source state-of-the-art method. Extensive evaluations on multiple public datasets fully demonstrate the accuracy and efficiency of our framework. Our code is released at <https://github.com/NAIL-HNU/EventMotionSeg> to facilitate further research in this field.

I. INTRODUCTION

Neuromorphic visual sensors, also known as event cameras, are bio-inspired devices that asynchronously capture pixel-level brightness changes with microsecond temporal resolution. Compared to conventional frame-based cameras, they offer higher dynamic range [1] and lower latency, making them ideal candidates for robotic perception tasks, such as optical flow estimation [2, 3], Simultaneous Localization and Mapping [4]–[7], and motion segmentation [8]–[10] under high-speed motion or challenging lighting conditions.

Motion segmentation with event cameras involves partitioning event streams into distinct clusters, each corresponding to either the background or an independently moving

Sheng Zhong, Xiya Zhu and Yi Zhou are with the Neuromorphic Automation and Intelligence Lab (NAIL) at School of Artificial Intelligence and Robotics, Hunan University, Changsha, China.

Zhongyang Ren conducted this work while he was at the Neuromorphic Automation and Intelligence Lab (NAIL), School of Artificial Intelligence and Robotics, Hunan University, Changsha, China.

Dehao Yuan is with Capital One.

Cornelia Fermüller is with University of Maryland, College Park.

* denotes equal contribution.

Corresponding author: Yi Zhou. Email: eeyzhou@hnu.edu.cn.

This work was supported by the National Key Research and Development Project of China under Grant 2023YFB4706600.

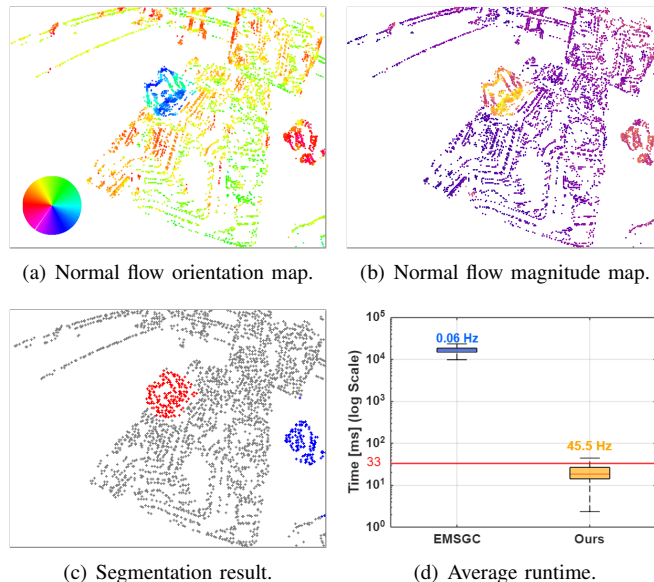


Fig. 1: The proposed system takes the normal flow generated by VecKM.Flow [11] as input and performs motion segmentation in real-time based on the normal flow constraint. (a) and (b) show the orientation and magnitude of the normal flow, respectively, with a circular color diagram in (a) indicating the angle-color correspondence. (c) presents the motion segmentation results of our system, with different colors representing distinct motion models. (d) compares the average runtime between our system and EMSGC [8] under identical setup. The mean operating frequency of each method are displayed above the corresponding box, with our system achieving a speedup of nearly $800\times$ compared to EMSGC.

object (IMO). The introduction of event cameras proves particularly beneficial in high-dynamic environments [12], enabling the capture of fast movements but hardly incurring any motion blur, as is common in traditional cameras. However, the camera’s ego-motion induces additional complexity, as it generates events across the entire image plane [13]–[15], complicating the segmentation process. The key challenge in addressing this problem is determining how to effectively assess data association corresponding to each IMO. One influential approach is motion compensation [16] [17], in which events are warped according to geometric models, and the goodness of fitting is evaluated using the variance of the Image of Warped Events (IWE). The majority of the literature [9, 14, 18]–[20] typically follows the motion compensation scheme to cluster events based on the goodness of the motion model fit, measured by various focal loss

metrics. Deep learning-based methods [10, 21] have also been explored to solve this task. Both types of methods have proven effective, but they either rely on prior knowledge of IMOs or require ground truth labels, which are not always available in real-world applications due to the lack of event data annotations.

To mitigate this dependency on such prior information, a spatio-temporal graph and Markov Random Field (MRF) are introduced in EMSGC [8], formulating a graph-based energy function. This approach is among the first to address event-based motion segmentation via graph-cut optimization. Though effective, EMSGC [8] still has strong limitations in the computational cost of graph construction, naive initialization strategy, and iterative motion model fitting, limiting its real-time performance.

Normal flow, the partial component of optical flow along the direction of the image gradient, provides a potential solution to these limitations. Recently, De et al. [11] propose VecKM_Flow, which is capable of directly learning dense normal flow from the event neighborhood. Leveraging the dense normal flow as input, we propose a normal flow-based motion segmentation framework building upon EMSGC that achieves comparable segmentation accuracy while enabling real-time performance, which is critical for this class of tasks. The contributions of this paper can be summarized as follows:

- A normal flow-based motion segmentation framework, which enables accurate identification of IMOs in the scene without relying on prior knowledge. This is achieved by formulating the motion segmentation task as an energy minimization problem solved via graph cuts, where normal flow clustering and motion model fitting are performed iteratively.
- A normal flow-based motion model initialization and fitting method, which enables fast estimation of motion models for IMOs using only a limited number of candidate models. This substantially reduces computational complexity and enables real-time system performance.
- A comprehensive evaluation on multiple public datasets demonstrates the efficiency and accuracy of the proposed motion segmentation framework. Our code will be open-sourced to facilitate further research in this field.

Outline: The rest of this paper is organized as follows. We begin by reviewing the event-based motion segmentation problem and the concept of event-based normal flow (Sec. II). Next, we present our method, providing a detailed discussion that highlights the contributions outlined in Sec. III. The experimental evaluation is then provided in Sec. IV, followed by a conclusion in Sec. V.

II. RELATED WORK

In this section, we provide a detailed survey of recent progress on both event-based segmentation (Sec. II-A) and normal flow (Sec. II-B).

A. Event-based motion segmentation

A naive idea [14] of identifying IMOs is to directly apply global motion compensation [16] to the event stream using parametric models. Then clusters of events that deviate from the motion model (residual motion) are labeled as IMOs. Compared to [14], [15] improves event-wise segmentation by jointly estimating event-object associations and object motion parameters, achieving significantly better performance on datasets while distinguishing subtle relative motion differences between clusters. Despite these improvements, the choice of motion models remains critical, as incorrect or inadequate model choices can still lead to segmentation failures. Parameshwara *et al.* [19, 20] advance event clustering by incorporating motion compensation alongside feature tracking, cluster splitting, and merging. This integration strengthens robustness over extended sequences but faces challenges in merging spatially distant clusters exhibiting similar motion.

Zhou *et al.* [8] further discuss the clustering nature of the segmentation problem and designed a space-time graph representation to solve this problem in a joint optimization manner. They reformulate the expectation-maximization (EM) [22] approach into a minimization framework using negative IWE, ensuring compatibility with graph-cut algorithms. Inspired by this framework, Lu *et al.* [23] develop a cascaded two-stage multi-model fitting scheme that addresses explicitly the cluster merging problem. Mitrokhin *et al.* [21] introduce a graph convolutional network for scene motion segmentation, which effectively learns to perform foreground-background segmentation tasks. Using Inertial Measurement Unit (IMU) measurements as camera motion, [9] projects events to a reference time and extracts an assumed columnar structure from the point cloud to identify moving objects. However, the reliance on strong shape assumptions restricts its applicability across diverse real-world datasets. A multi-modal model for instance-level moving object segmentation is proposed in [10], yet its reliance on RGB images limits applicability in most scenarios where such data is unavailable.

B. Event-based normal flow

Characterizing the relative motion between the camera and dynamic scenes, optical flow [24] is a representation of the movement of pixels over time, encoding the direction and velocity of motion. However, since event cameras capture only pixels where brightness changes occur, they can recover only partial components of optical flow—specifically, the normal flow—through local event data [2, 25]. Estimation of relative camera motion from normal flow observations has been studied in [26]. For event cameras, normal flow has been used for drones’ course estimation [27] and obstacle avoidance [28]. Ren *et al.* [29] propose that the normal flow constraint can serve as an alternative to full optical flow, addressing motion and structure problems in static scenes. Similar usage of event-based normal flow has also been witnessed for state estimation [30, 31] and autonomous driving [32]. Nevertheless, their method for computing normal

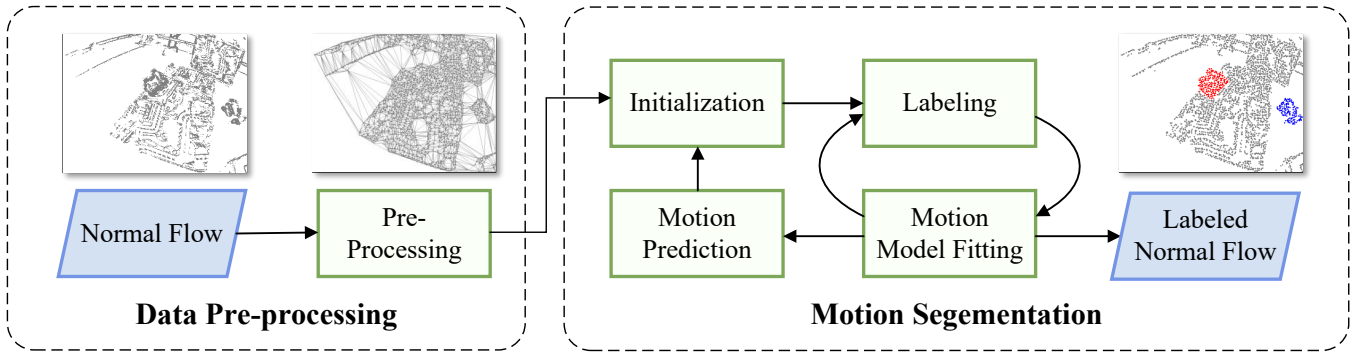


Fig. 2: *Flowchart of the proposed system.* The proposed system comprises two independently operating modules. The data pre-processing module downsamples the input dense normal flow and constructs a spatial graph via Delaunay triangulation [39]. The motion segmentation module iteratively alternates between normal flow clustering (Labeling) and motion model fitting to segment the normal flow associated with IMOs.

flow is relatively simplistic, and the sparse normal flow they obtain lacks robustness to noise, limiting its effectiveness in dynamic and noisy environments.

The computation of visual flow from event data has been witnessed from both geometric [2, 25, 33] and data-driven [3, 34, 35] perspectives. Only recently, VecKM_Flow [11] and its real-time variation [36] are introduced to directly learn event-wise normal flow from event neighborhoods through a scalable local event encoder named VecKM [37], offering an improvement in handling dynamic conditions. Building on normal flow, [38] uses hierarchical clustering for motion segmentation and ego-motion estimation. Though achieving fair accuracy, the approach is computationally expensive and nowhere near real-time performance.

III. METHODOLOGY

We detail our method in this section. First, we provide an overview of the proposed system (Sec. III-A). Second, we present preliminaries on problem formulation, including normal flow and motion model fitting based on normal flow (Sec. III-B). Third, we discuss how to formulate motion segmentation as an energy minimization problem solved via iterative normal flow clustering and motion model fitting (Sec. III-C). Finally, we introduce the efficient initialization strategy employed in the proposed system (Sec. III-D).

A. System Overview

The goal of our motion segmentation framework is to find a labeling function $L(\mathbf{n}) : \mathcal{N} \rightarrow \mathcal{L} = \{1, \dots, N\}$, which assigns to the input dense normal flow $\mathbf{n} \in \mathcal{N}$ different labels $l \in \mathcal{L}$ based on the motion models $\mathcal{M} = \{\mathbf{m}_1, \dots, \mathbf{m}_N\}$, thereby determining the positions and number of IMOs in the scene, with the overall operational logic being analogous to that of EMSGC [8]. Specifically, the proposed system comprises two independently operating modules and takes as input the dense normal flow generated by VecKM_flow [11], as illustrated in Fig. 2. In the data pre-processing module, the normal flow is downsampled at fixed time intervals, followed by the construction of a spatial graph via Delaunay

triangulation [39], which is then fed into the motion segmentation module. In the motion segmentation module, an initial set of candidate motion models is first generated through an initialization step (Sec. III-D). Subsequently, an iterative process of normal flow clustering and motion model fitting is performed until the motion models converge, upon which the segmented normal flow is output (Sec. III-C). Notably, after motion prediction, regions containing IMOs are employed to initialize the motion models for the next segmentation. This effectively reduces the number of candidate motion models required, thereby enhancing system efficiency.

B. Problem Statement Preliminary

Before detailing our method, we first revisit two essential concepts as preliminaries, including normal flow and motion model fitting based on normal flow.

1) *Event-based Normal Flow and Constraint:* Event cameras capture motion primarily perpendicular to edges, providing only partial observations of optical flow \mathbf{u} in the direction of the local image gradient (∇I) . This is referred to as normal flow $\mathbf{n} = \mathbf{u}_\perp$. Using the definition of the time surface, i.e. $\Sigma_e(\mathbf{x}) : \mathbf{x} \rightarrow t$, which maps 2D pixel coordinates to 1D timestamps, the normal in event space is defined as:

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla \Sigma_e(\mathbf{x})}{\|\nabla \Sigma_e(\mathbf{x})\|_2}. \quad (1)$$

Here, $\nabla \Sigma_e(\mathbf{x})$ represents the spatial gradient of the time surface at \mathbf{x} . Leveraging the relationship between both flows, Ren *et al.* [29] demonstrate that event-based normal flow can serve as an alternative to full optical flow by introducing a constraint, as expressed in the following equation, referred to as the normal flow constraint:

$$\mathbf{n}(\mathbf{x})^\top \mathbf{u}(\mathbf{x}; \theta) - \|\mathbf{n}(\mathbf{x})\|_2^2 \doteq 0. \quad (2)$$

The normal flow constraint can effectively replace optical flow in solving various geometric model fitting problems (with parameters denoted as θ). It provides a computationally efficient alternative, especially in dynamic scenes, and enhances robustness in tasks like motion estimation and scene geometry, where traditional flow estimation may be challenging or prone to errors.

2) *Motion model fitting on normal flow*: Similar to motion model fitting via the contrast maximization framework [16] for event data, normal flow derived from local event observations can be employed for geometric model fitting tasks [29]. Assuming objects are not very large, usually motion models that implicitly model the geometry of the scene are assumed for segmentation. To maintain generality, we employ the affine motion model as previous works do [8, 14]. The basic 4-parameter affine model can be written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \rho \cos \theta & -\rho \sin \theta & t_x \\ \rho \sin \theta & \rho \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (3)$$

where x' and y' represent the pixel coordinates after the transformation, and (ρ, θ, t_x, t_y) are the parameters defining the affine transformation. Then, the homogeneous coordinate of optical flow $\hat{\mathbf{u}}$ can be denoted as:

$$\hat{\mathbf{u}} = \left(\begin{bmatrix} \rho \cos \theta & -\rho \sin \theta & t_x \\ \rho \sin \theta & \rho \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} - I \right) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (4)$$

Combining this homogeneous representation with normal flow constraint in Eq. 2, we derive

$$\mathbf{n}^\top \begin{bmatrix} \rho \cos \theta - 1 & -\rho \sin \theta & t_x \\ \rho \sin \theta & \rho \cos \theta - 1 & t_y \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \|\mathbf{n}\|_2^2, \quad (5)$$

which can be used to assess the goodness of fitting.

C. Solution for Motion Segmentation

As illustrated in Sec. III-A, we solve the problem by alternately minimizing the two subproblems of normal flow labeling and motion model fitting of each labeled cluster to optimize L and \mathcal{M} , respectively. Following a similar strategy as employed in EMSGC [8], the motion segmentation task is formulated as an energy minimization problem that involves fitting multiple motion models:

$$E(L, \mathcal{M}) \doteq E_D(L, \mathcal{M}) + \lambda_P E_P(L) + \lambda_M E_M(L), \quad (6)$$

where $E_D(L, \mathcal{M})$ is the data term, representing the model fitting error, while $E_P(L)$ and $E_M(L)$ correspond to the smoothness term [40] and label cost term [41], respectively. The weights λ govern the contribution of each term.

We begin by introducing the labeling process. In the pre-processing step, the input event-wise normal flow is constructed into a 2D spatial graph via Delaunay triangulation [39], establishing adjacency relationships. Once the candidate motion models \mathcal{M} obtained from initialization are fixed, Eq. 6 is transformed into an MRF problem [42] that can be solved using the alpha-expansion algorithm [43], thereby yielding the labeling function L .

As for motion model fitting, after fixing the labels L , data term $E_D(\mathcal{M})$ is the only remaining component in Eq. 6. Thus, the model fitting process could be resolved by either a linear or a nonlinear optimization method. For the linear

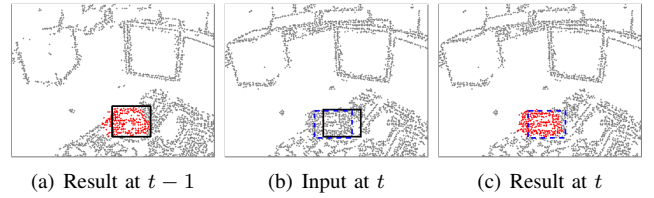


Fig. 3: *Procedure of motion prediction*. (a) Segmentation result at $t - 1$, with the black solid box indicating the system-generated region containing an IMO. (b) Normal flow input at t , with the blue dashed box indicating the predicted IMO region after motion prediction. Normal flow within this region is used to initialize a candidate motion model. (c) Segmentation result at t , where the IMO primarily resides within the predicted box.

solution, according to Eq. 5, the fitting error of each normal flow observation can be reformulated as:

$$[n_x x, n_x y, n_x, n_y x, n_y y, n_y] \mathbf{m} = \|\mathbf{n}\|_2^2, \quad (7)$$

where n_x, n_y denote the component along the x and the y axes of normal flow, and \mathbf{m} is the vector representation of affine transformation:

$$\mathbf{m} = [\rho \cos \theta - 1 \quad -\rho \sin \theta \quad t_x \quad \rho \sin \theta \quad \rho \cos \theta - 1 \quad t_y]^\top. \quad (8)$$

By stacking multiple normal flow observations from the same label, we can construct a linear equation:

$$\begin{pmatrix} n_{1x}x, n_{1x}y, n_{1x}, n_{1y}x, n_{1y}y, n_{1y} \\ \vdots \\ n_{kx}x, n_{kx}y, n_{kx}, n_{ky}x, n_{ky}y, n_{ky} \end{pmatrix} \mathbf{m} = \begin{pmatrix} \|\mathbf{n}_1\|_2^2 \\ \vdots \\ \|\mathbf{n}_k\|_2^2 \end{pmatrix}. \quad (9)$$

After solving the above overdetermined system, the affine model parameters can be decoupled by

$$\begin{cases} \rho = \sqrt{((m_1 + m_5)/2 + 1)^2 + ((m_2 - m_4)/2)^2} \\ \theta = \arcsin\left(\frac{m_4 - m_2}{2\rho}\right) \\ v_x = m_3 \\ v_y = m_6 \end{cases}, \quad (10)$$

where m_i is the i th entry of \mathbf{m} . Although the linear solution is efficient, the presence of outliers can cause instability when the number of normal flow values is large. Therefore, the proposed system typically formulates the model fitting error (Eq. 5) as a least squares problem:

$$E_D(\mathcal{M}) = \sum_{\mathbf{x}_k \in \varepsilon_l} (\mathbf{n}(\mathbf{x}_k)^\top \mathbf{u}(\mathbf{x}_k, \boldsymbol{\theta}) - \|\mathbf{n}(\mathbf{x}_k)\|_2^2)^2. \quad (11)$$

Then, the affine model parameters $\boldsymbol{\theta} = (\rho, \theta, t_x, t_y)$ are optimized using the Levenberg-Marquardt algorithm.

D. Initialization

We now detail how the optimization procedure is initialized. Leveraging the characteristics of normal flow and the motion continuity of IMOs, we propose a simple and efficient initialization strategy, which consists of two parts:

TABLE I: Summary of characteristics of the datasets used.

Dataset	Camera	Resolution	Env.	HDR
EED [14]	DAVIS240	240×180	Indoor	Yes
EVIMO [18]	DAVIS346	346×260	Indoor	No
EMSGC [8]	DAVIS346	346×260	Outdoor	No

1) *Fast Sampling*: Upon receiving new normal flow in the motion segmentation module, we first sample n instances of normal flow with significantly different translation vectors. These translation vectors are then used to initialize the translation components (t_x, t_y) of the affine motion model as defined in Eq. 3. This approach leverages the inherent translational motion information encapsulated within the normal flow, as it constitutes the projection of optical flow onto the gradient direction. For the scale factor ρ and rotation angle θ , which typically exhibit only minor variations, we directly set them to initial values of 1 and 0, respectively, yielding n complete candidate motion models.

2) *Motion Prediction*: After performing motion segmentation at time $t-1$, for each label identified as belonging to an IMO, we first apply region growing to generate a bounding box that encompasses the IMO as much as possible. Given the corresponding motion model and the system’s operation rate, we then predict the box’s position at time t , as shown in Fig. 3. Before beginning the motion segmentation at time t , the normal flow within each predicted box is used to fit a candidate motion model by nonlinear optimization. The candidate motion models obtained using this method can effectively improve the accuracy of motion segmentation when motion prediction is reliable. Consequently, when these models are available, the number of models generated by fast sampling is reduced (6 in our implementation when available, and 12 otherwise).

The proposed initialization strategy can initialize a sufficient number of candidate motion models with minimal computational cost while achieving high-quality motion segmentation results. In contrast, EMSGC performs an N-level subdivision operation to divide the event volume and then initializes 85 candidate motion models on the sub-volumes through motion compensation, which results in significantly higher computational complexity. Detailed computational performance can be found in Sec. IV-C. This highlights the advantage of using normal flow over directly operating on raw event streams.

IV. EXPERIMENT

In this section, we evaluate the proposed system. We first describe the experimental datasets and evaluation metrics (Sec. IV-A), followed by quantitative and qualitative comparisons against other methods across multiple benchmarks (Sec. IV-B). Finally, we analyze the computational efficiency of the proposed system (Sec. IV-C).

TABLE II: Quantitative evaluation on the EED dataset, with the evaluation metric being detection rate (in %).

Sequence	EMSMC [15]	EMSGC [8]	Ours
<i>Fast drone</i>	96.30	96.30	96.30
<i>Lighting variation</i>	80.51	93.51	98.70
<i>Occlusions</i>	92.31	100.00	100.00
<i>What is background</i>	100.00	100.00	100.00
Average	92.28	97.45	98.75

TABLE III: Quantitative evaluation on the EVIMO dataset, with the evaluation metric being IoU.

Sequence	EMSGC [8]	Ours
<i>box</i>	0.30	0.56
<i>table</i>	0.46	0.55
Average	0.38	0.55

A. Datasets and Evaluation Metrics

To evaluate the system performance, we conduct evaluations on three publicly available event-based motion segmentation datasets, with key characteristics detailed in Tab. I. The Extreme Event Dataset (EED) [14] is one of the first open-source datasets for IMO detection and tracking. Each sequence contains camera ego-motion and IMOs, collected in laboratory environments featuring controlled high dynamic range conditions. EVIMO [18], also captured in laboratory settings but under better illumination, features sequences with up to three IMOs and provides dense segmentation masks for evaluation. Unlike the aforementioned datasets, EMSGC [8] provides outdoor sequences containing non-rigid IMOs (e.g., pedestrians), on which we conduct qualitative comparisons with the EMSGC algorithm. In addition, we record a small number of sequences in a room equipped with the NOKOV Mars26H Motion Capture system, which are used for computational time analysis.

Evaluation Metrics. Due to the varying formats of ground-truth (GT) provided by different datasets, we employed two distinct evaluation metrics. The first metric is detection rate, introduced in [14], which evaluates bounding box overlap between detected and GT objects. A detection is considered successful when it meets the following criteria:

$$\mathcal{B}_D \cap \mathcal{B}_G > 0.5 \text{ and } \mathcal{B}_D \cap \mathcal{B}_G > \mathcal{B}_D \cap \overline{\mathcal{B}_G}, \quad (12)$$

where \mathcal{B}_D refers to the estimated convex hull, \mathcal{B}_G the GT bounding box, and $\bar{\cdot}$ denotes the operation of the set complement.

The second evaluation metric is Intersection over Union (IoU) [19], which is calculated as the ratio of the intersection area between the predicted mask \mathcal{M}_D and the GT mask \mathcal{M}_G to the area of their union:

$$\text{IoU} = \frac{\mathcal{M}_D \cap \mathcal{M}_G}{\mathcal{M}_D \cup \mathcal{M}_G}, \quad (13)$$

A higher IoU indicates better alignment and more accurate segmentation.

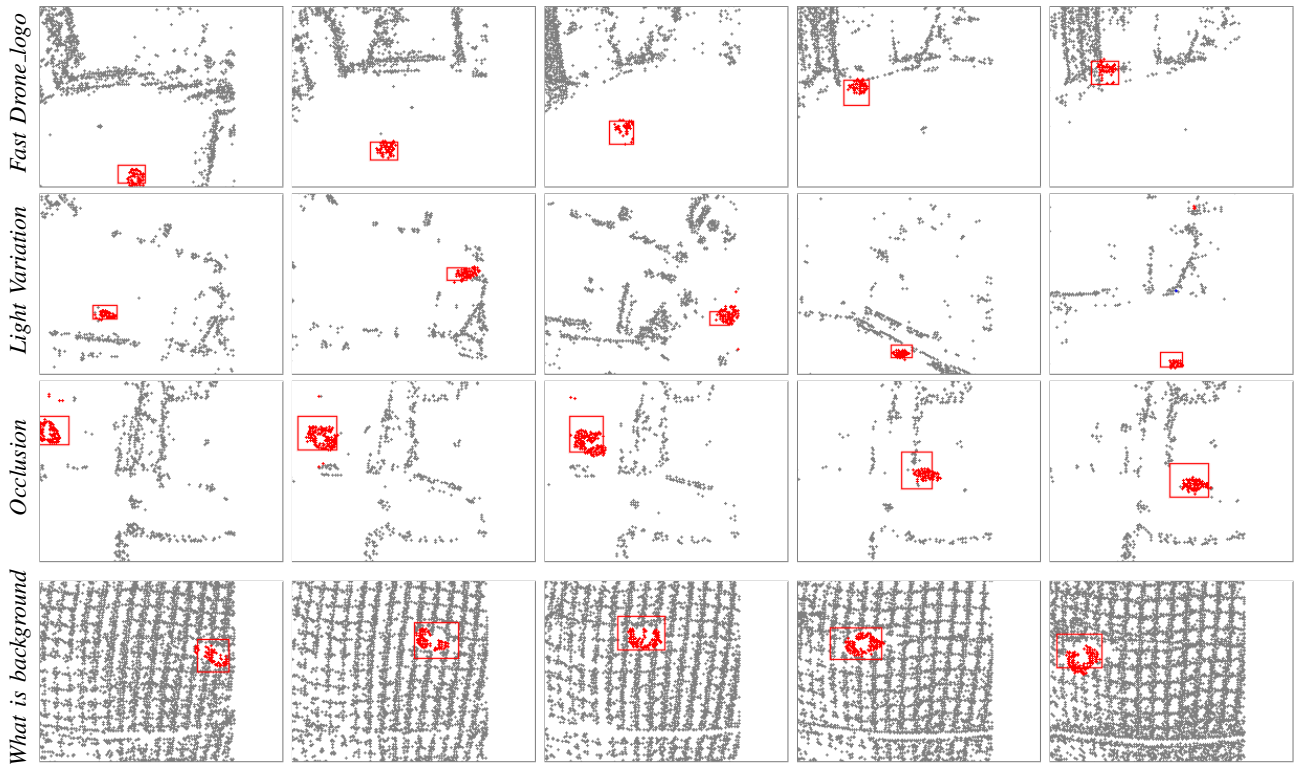


Fig. 4: Segmentation results on the EED dataset [14]. Time runs from left to right. The ground truth bounding boxes are denoted by red rectangles. Since the boxes are manually annotated on the grayscale images, and the timestamps cannot be perfectly aligned with the segmentation results, offsets are witnessed, especially for fast-moving IMOs.

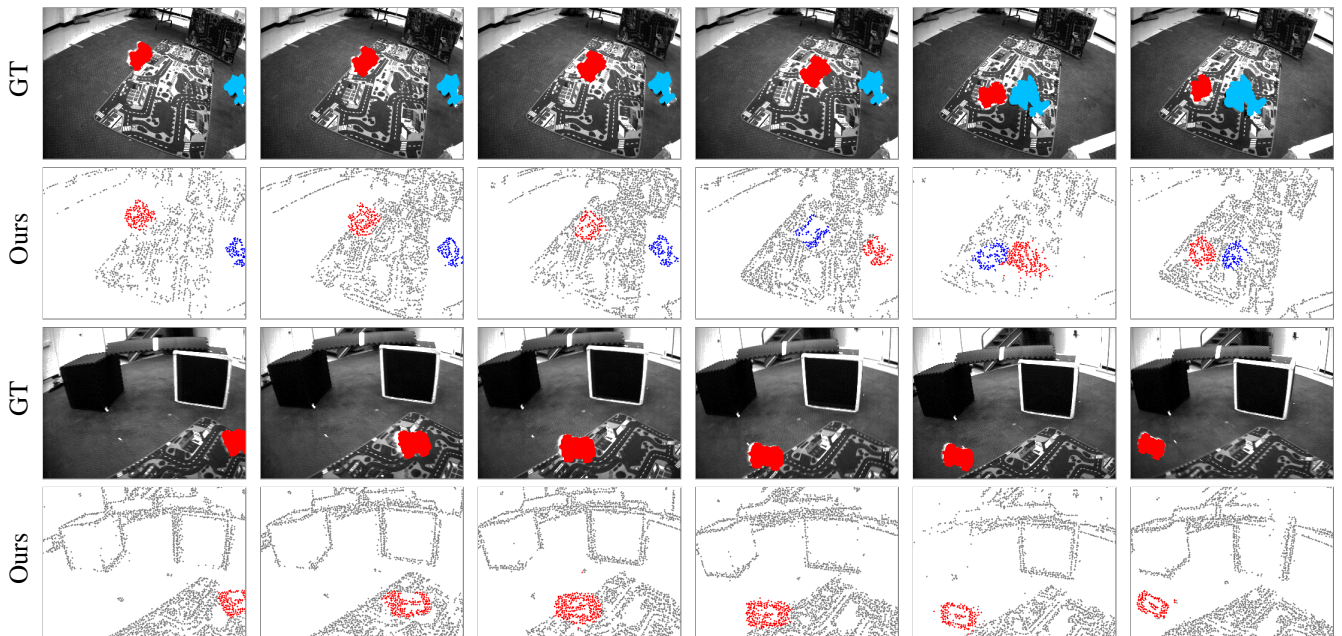


Fig. 5: Segmentation results on the EVIMO dataset [8], on sequences *Table* (rows 1-2) and *Boxed* (rows 3-4). Time runs from left to right. The grayscale images are used for visualization only. The color of each label is determined by the number of normal flow associated with it, meaning that the label color for the same IMO may vary at different time. The ground truth mask is not perfectly aligned with the IMOs in the grayscale image due to the presence of motion blur.

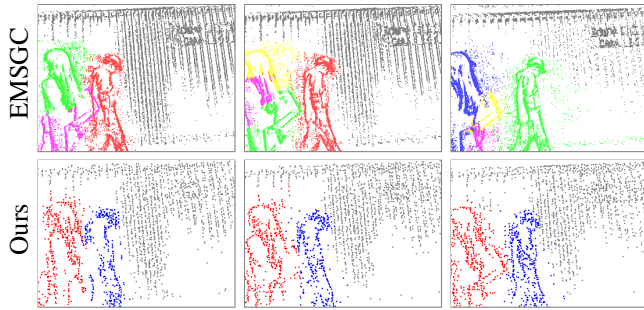


Fig. 6: *Qualitative comparison on the EMSGC dataset [8]. Time runs from left to right. Different colors indicate different motion models.*

TABLE IV: Computational performance [time: ms].

Node	Function	EMSGC [8]	Ours
Data Pre-processing	Pre-processing	87.65	25.50
	Initialization	5575.97	0.25
Motion Segmentation	Labeling & Fitting	10892.85	21.73
	Subtotal	16468.82	21.98

B. Quantitative and Qualitative Evaluation

We first evaluate our system on the EED dataset [14], which covers four extreme scenarios: high-speed motion, illumination changes, occlusions, and moving behind background nets. As shown in Tab. II, our method outperforms two classic motion-compensation-based algorithms, EMSMC [15] and EMSGC [8]. The qualitative results can be found in Fig. 4, where the GT bounding boxes are highlighted with red rectangles. Notably, despite the emergence of several event-based motion segmentation frameworks in recent years, we are unable to compare with these methods due to the use of different datasets and evaluation metrics, as well as the closed-source code.

The second dataset is EVIMO [18], where our method achieves significantly better segmentation results than EMSGC, as shown in Tab. III. The relatively low values stem from two factors: temporal misalignment between our segmentation results and the ground truth timestamps, and the sparser nature of our segmentation outputs (Fig. 5), which introduces errors during mask generation. The difference in EMSGC’s results compared to those in its manuscript is attributed to the different way we used to generate the masks.

Finally, we conduct a qualitative comparison with EMSGC [8] on the outdoor sequence recorded by its authors, as shown in Fig. 6. EMSGC’s initialization strategy causes fragmentation when segmenting non-rigid IMOs, erroneously assigning events from a single IMO to multiple motion models. In contrast, our method achieves more accurate and consistent segmentation. The visually sparser appearance of our results is due to downsampling and a shorter interval.

C. Computational Efficiency

As shown in Tab. IV, we compare the computational efficiency of EMSGC [8] and the proposed system using

a desktop with an Intel Core i7-14700k CPU. Both systems, implemented in C++ within the ROS framework, are tested on the EVIMO dataset and the self-recorded sequences. Compared to the extremely high computational cost of EMSGC, our system, benefiting from the characteristics of normal flow, significantly reduces computational complexity in two aspects, enabling the entire system to operate in real time at 30 Hz or an even higher rate, which is especially crucial for time-sensitive motion segmentation tasks.

This achievement is attributed to the following two features: in terms of initialization, as mentioned in Sec. III-D, EMSGC uses the CMax framework [16] for motion model fitting, which is highly sensitive to the initial values of the motion models. This necessity to initialize a large number of candidate models to avoid failure, combined with the high computational complexity of CMax’s operation on raw event data, results in an initialization time on the order of seconds for EMSGC. In contrast, the motion information contained in the normal flow allows the proposed system to initialize the motion models simply by traversing the normal flow. The high tolerance of the normal flow-based model fitting framework to initial values, along with the motion prediction strategy, significantly reduces the number of required candidate motion models, allowing our system to compress the initialization time to sub-millisecond levels. Second, the local representation ability of the normal flow significantly reduces the amount of data our system needs, thereby lowering the computational complexity of labeling and motion model fitting. The reduction stems from two aspects: spatially, our system utilizes only one normal flow per neighborhood instead of all events; temporally, our system achieves accurate motion segmentation using data over shorter time intervals. For instance, on the EVIMO dataset, our system requires only 10 ms or even shorter normal flow for segmentation, whereas EMSGC relies on 50 ms of events. With further improvements, there is still significant potential for accelerating our system.

V. CONCLUSION

We propose a real-time motion segmentation system with event-based normal flow. The proposed system is built upon EMSGC [8], an event-based motion segmentation framework, which formulates the problem as an energy minimization optimization problem and solves it using graph cuts. To address the computational limitation in EMSGC, we introduce event-based normal flow and, based on its characteristics, propose a novel motion model initialization and fitting method, which enables the proposed system to efficiently estimate the motion models of IMOs with only a limited number of candidate models. This significantly reduces the system’s computational complexity, enabling real-time operation and representing a significant step forward in event-based motion segmentation for practical applications. Extensive evaluations on three publicly available datasets demonstrate the efficiency of our system and its better segmentation performance compared to EMSGC. Despite these strengths, the current framework’s reliance on high-

quality normal flow may limit its robustness under extreme conditions. We believe future research could address this by integrating multi-scale flow features or learning-based priors to improve system reliability. Furthermore, extending the system to incorporate more complex motion models could improve its performance on non-rigid or deformable objects.

REFERENCES

- [1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor," vol. 43, no. 2, pp. 566–576, 2008.
- [2] H. Akolkar, S.-H. Ieng, and R. Benosman, "Real-time high speed motion prediction using fast aperture-robust event-driven visual flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 361–372, 2020.
- [3] D. R. Kepple, D. Lee, C. Prepsius, V. Isler, I. M. Park, and D. D. Lee, "Jointly learning visual motion and confidence from local patches in event cameras," in *European Conference on Computer Vision*. Springer, 2020, pp. 500–516.
- [4] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios," vol. 3, no. 2, pp. 994–1001, Apr. 2018.
- [5] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," vol. 37, no. 5, pp. 1433–1450, 2021.
- [6] S. Klenk, M. Motzet, L. Koestler, and D. Cremers, "Deep event visual odometry," in *International Conference on 3D Vision, 3DV 2024, Davos, Switzerland, March 18-21, 2024*. IEEE, 2024, pp. 739–749.
- [7] J. Niu, S. Zhong, X. Lu, S. Shen, G. Gallego, and Y. Zhou, "Esvo2: Direct visual-inertial odometry with stereo event cameras," *IEEE Transactions on Robotics*, 2025.
- [8] Y. Zhou, G. Gallego, X. Lu, S. Liu, and S. Shen, "Event-based motion segmentation with spatio-temporal graph cuts," 2021.
- [9] H. Zhou, Z. Shi, H. Dong, S. Peng, Y. Chang, and L. Yan, "Jstr: Joint spatio-temporal reasoning for event-based moving object detection," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 650–10 656.
- [10] Z. Wan, B. Fan, L. Hui, Y. Dai, and G. H. Lee, "Instance-level moving object segmentation from a single image with events," *International Journal of Computer Vision*, pp. 1–22, 2025.
- [11] D. Yuan, L. Burner, J. Wu, M. Liu, J. Chen, Y. Aloimonos, and C. Fermüller, "Learning normal flow directly from event neighborhoods," *arXiv preprint arXiv:2412.11284*, 2024.
- [12] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conrad, K. Daniilidis, et al., "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.
- [13] A. Glover and C. Bartolozzi, "Event-driven ball detection and gaze fixation in clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2203–2208.
- [14] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.
- [15] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, "Event-based motion segmentation by motion compensation," 2019, pp. 7243–7252.
- [16] G. Gallego, H. Rebecq, and D. 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*, 2018, pp. 3867–3876.
- [17] G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 280–12 289.
- [18] A. Mitrokhin, C. Ye, C. Fermüller, Y. Aloimonos, and T. Delbruck, "Ev-imo: Motion segmentation dataset and learning pipeline for event cameras," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6105–6112.
- [19] C. M. Parameshwara, N. J. Sanket, A. Gupta, C. Fermuller, and Y. Aloimonos, "MOMS with Events: Multi-object motion segmentation with monocular event cameras," *arXiv:2006.06158*, 2020.
- [20] C. M. Parameshwara, N. J. Sanket, C. D. Singh, C. Fermüller, and Y. Aloimonos, "0-mms: Zero-shot multi-motion segmentation with a monocular event camera," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9594–9600.
- [21] A. Mitrokhin, Z. Hua, C. Fermuller, and Y. Aloimonos, "Learning visual motion segmentation using event surfaces," 2020, pp. 14 414–14 423.
- [22] X.-L. Meng and D. Van Dyk, "The em algorithm—an old folk-song sung to a fast new tune," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 59, no. 3, pp. 511–567, 1997.
- [23] X. Lu, Y. Zhou, and S. Shen, "Event-based motion segmentation by cascaded two-level multi-model fitting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4445–4452.
- [24] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981.
- [25] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," vol. 25, no. 2, pp. 407–417, 2013.
- [26] C. M. Parameshwara, G. Hari, C. Fermüller, N. J. Sanket, and Y. Aloimonos, "Diffposenet: direct differentiable camera pose estimation," 2022, pp. 6845–6854.
- [27] R. Dinaux, N. Wessendorp, J. Dupeyroux, and G. C. De Croon, "Faith: Fast iterative half-plane focus of expansion estimation using optic flow," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7627–7634, 2021.
- [28] X. Clady, C. Clercq, S.-H. Ieng, F. Houseini, M. Randazzo, L. Natale, C. Bartolozzi, and R. Benosman, "Asynchronous visual event-based time-to-contact," vol. 8, no. 9, 2014.
- [29] Z. Ren, B. Liao, D. Kong, J. Li, P. Liu, L. Kneip, G. Gallego, and Y. Zhou, "Motion and structure from event-based normal flow," in *European Conference on Computer Vision*. Springer, 2024, pp. 108–125.
- [30] X. Lu, Y. Zhou, J. Niu, S. Zhong, and S. Shen, "Event-based visual inertial velometer," 2024.
- [31] X. Lu, Y. Zhou, J. Mai, K. Dai, Y. Xu, and S. Shen, "Event-based visual-inertial state estimation for high-speed maneuvers," *IEEE Transactions on Robotics*, 2025.
- [32] J. Li, B. Liao, X. Lu, P. Liu, S. Shen, and Y. Zhou, "Event-aided time-to-collision estimation for autonomous driving," in *European Conference on Computer Vision*. Springer, 2024, pp. 57–73.
- [33] J. Nagata, Y. Sekikawa, and Y. Aoki, "Optical flow estimation by matching time surface with event-based cameras," *Sensors*, vol. 21, no. 4, p. 1150, 2021.
- [34] Z. Wan, Y. Dai, and Y. Mao, "Learning dense and continuous optical flow from an event camera," *IEEE Transactions on Image Processing*, vol. 31, pp. 7237–7251, 2022.
- [35] S. Shiba, Y. Aoki, and G. Gallego, "Secrets of event-based optical flow," 2022, pp. 628–645.
- [36] D. Yuan and C. Fermüller, "A real-time event-based normal flow estimator," *arXiv preprint arXiv:2504.19417*, 2025.
- [37] D. Yuan, C. Fermuller, T. Rabbani, F. Huang, and Y. 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*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 57 871–57 886. [Online]. Available: <https://proceedings.mlr.press/v235/yuan24b.html>
- [38] Z. Hua, D. Yuan, and C. Fermüller, "Motion segmentation and egomotion estimation from event-based normal flow," *arXiv preprint arXiv:2507.14500*, 2025.
- [39] J. R. Shewchuk, "General-dimensional constrained delaunay and constrained regular triangulations, i: Combinatorial properties," in *Twentieth Anniversary Volume*. Springer, 2009, pp. 1–58.
- [40] R. B. Potts, "Some generalized order-disorder transformations," in *Mathematical proceedings of the cambridge philosophical society*, vol. 48, no. 1. Cambridge University Press, 1952, pp. 106–109.
- [41] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," vol. 96, no. 1, pp. 1–27, 2012.
- [42] C. Russell, J. Fayad, and L. Agapito, "Energy based multiple model fitting for non-rigid structure from motion," in *CVPR 2011*. IEEE, 2011, pp. 3009–3016.
- [43] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.