

# Un-EVIMO: Unsupervised Event-based Independent Motion Segmentation

Ziyun Wang<sup>1</sup>, Jinyuan Guo<sup>1</sup>, and Kostas Daniilidis<sup>1,2</sup>

<sup>1</sup> University of Pennsylvania

<sup>2</sup> Archimedes, Athena RC

**Abstract.** Event cameras are a novel type of biologically inspired vision sensor known for their high temporal resolution, high dynamic range, and low power consumption. Because of these properties, they are well-suited for processing fast motions that require rapid reactions. Event cameras have shown competitive performance in unsupervised optical flow estimation. However, performance in detecting independently moving objects (IMOs) is lacking behind, although event-based methods would be suited for this task based on their low latency and HDR properties. Previous approaches to event-based IMO segmentation heavily depended on labeled data. However, biological vision systems have developed the ability to avoid moving objects through daily tasks without using explicit labels. In this work, we propose the first event framework that generates IMO pseudo-labels using geometric constraints. Due to its unsupervised nature, our method can flexibly handle a non-predetermined arbitrary number of objects and is easily scalable to datasets where expensive IMO labels are not readily available. Our approach shows competitive performance on the EVIMO dataset compared with supervised methods, both quantitatively and qualitatively. See the project website for details: [https://www.cis.upenn.edu/~ziyunw/un\\_evimo/](https://www.cis.upenn.edu/~ziyunw/un_evimo/).

**Keywords:** Event Cameras · Motion Segmentation

## 1 Introduction

Biological visual systems show remarkable performance in identifying independently moving objects when the viewer is undergoing self-motion. Basketball players can catch a ball flying at high speed while running across the court. Insects have neurons optimized for detecting independent motion to search for prey or avoid threats [26]. Cross-species studies have found that biological systems have neurons that specialize in detecting looming motion, a special case of independent motion [42]. Scientists have found that certain parts of the visual field are involved in subtracting out self-motion to help identify moving objects [31]. In cognitive science, the ability to model or segment independently moving objects has been extensively studied [17, 33–35]. Human drivers have the ability to identify moving pedestrians and avoid them even when the car is traveling at high speed. Another consideration is the speed of camera and depth sensors, which has become the bottleneck of autonomous vision [18]. High-accuracy depth sensors, e.g. LIDAR, are able to map rigid scenes but have to apply semantic segmentation in order to detect Independently Motion Objects (IMOs).

**Table 1:** Feature comparisons. Un-EVIMO does not simplify the geometry by following the complete motion field model; it does not require manual labeling of IMO objects; it trains a network that performs inference on scenes without extensive tuning; and it runs inference at real-time without heavy optimization.

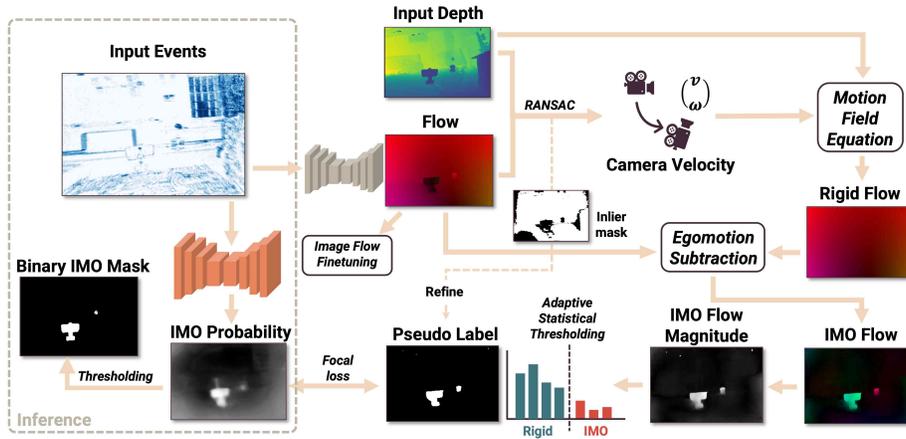
	Fast (Real-Time)	Scalable (No IMO Labels)	Minimal Tuning	Full Motion Models
EMSGC	✗	✓	✗	✗
EVIMO Network	✓	✗	✓	✗
SpikeMS	✓	✗	✓	-
ESMS	✗	✗	✓	-
Un-EVIMO	✓	✓	✓	✓

The recent development of event-based cameras has brought hope to these issues. Event cameras are able to record the log change of brightness of individual pixels asynchronously. These low-latency cameras allow for continuous monitoring of motion patterns of the scene. In this work, inspired by biological vision systems, we use an event camera as a silicon “eye” and tackle the IMO segmentation problem given a stream of events. CNN-based approaches have shown success in dense segmentation tasks. In this work, we use neural networks as our predictor to take advantage of their generalizability. The bottleneck of event-based algorithms is the need for a tremendous amount of labeled training data. However, if we examine how species acquired the ability to handle IMOs, the labels do not need to come from annotated binary masks. Actually, many studies have shown that the motion field itself contains enough information to differentiate between self-motion and independent motion [26, 42]. An important question is: *Can we learn motion segmentation with event cameras without manual labels by looking at the motion pattern in the scene?* In this work, we propose a novel framework for training IMO segmentation networks in an unlabeled dataset. Un-EVIMO is the first event-based learning framework for IMO detection without being trained with manual labels. We use a geometric self-labeling method to generate binary IMO pseudo-labels that supervise the IMO segmentation network. Our framework uses off-the-shelf optical flow prediction and input depth to fit 3D camera motion using RANSAC for excluding IMO as outliers. IMO flow field is obtained by subtracting the camera motion-induced flow field from the combined flow field. Pseudo-labels are generated through adaptive thresholding techniques based on the magnitude of estimated IMO motion field. Running inference Un-EVIMO is simple without parameter turning because while the training process requires geometry-based labels, only events are used for prediction. Unlike many previous works, we do not assume simplified motion models or a known number of objects.

## 2 Related Work

### 2.1 Event-based Motion Segmentation

Recent advances in event-based motion segmentation research are driven by several event-based datasets. EVIMO [24] is a motion segmentation data set that contains more



**Fig. 1:** Proposed pipeline. **Left Dotted Box:** we train a network to directly predict IMO masks from events. **Rest of Figure:** we use a geometric self-labeling method to generate IMO pseudo-labels for supervision. Our framework uses off-the-shelf optical flow (fine-tuned on image-based flow) and input depth. The camera motion fitted from flow and depth through RANSAC is used to compute egomotion flow. Pseudo-labels are generated through adaptive thresholding techniques based on the magnitude of estimated IMO motion field. We take the best of both worlds of deep learning and optimization: **1)** simple and robust inference with a simple feed-forward pass, and **2)** scalable with no expensive annotations required to train the network.

than 30 minutes of various motions of scanned objects with a moving camera. Objects are geometrically tracked with a multi-camera tracking system (Vicon) and then projected onto a tracked camera. In the EVIMO paper, a baseline approach has been proposed to learn the mixture of unsupervised 3D velocities, depth, and flow from events. Motion segmentation is trained using the motion masks provided in the datasets on top of the learned mixture weights. Recently, Burner et al. released EVIMO2 [4], which uses VGA resolution cameras. Evidodgenet [36] predict camera velocity by deblurring ground events using a downward-facing event camera and a motion segmentation network to identify objects that need to be dodged. Stoffregen et al. [38] proposed an Expectation-Maximization framework that assigns events to different motion clusters by optimizing the event-based contrast maximization. EMSGC [49] is an optimization method that uses a graph cut method to cluster events in the  $x$ - $y$ - $t$  event space based on parametric flow. Mitrokhin et al. [23] use a graph neural network to learn the segmentation masks directly in the event point space. GConv [23] uses a graph neural network to learn event-based segmentation on graphs constructed on down-sampled events. SpikeMs [29] apply a spiking neural network (SNN) architecture that allows incremental updates of the prediction over a longer time horizon. We compare the features of these methods with our work in Table 1.

## 2.2 Unsupervised Motion Segmentation

Motion estimation and segmentation are coupled problems [32]. In classical computer vision, motion segmentation is solved by optimization that simultaneously estimates

parametric flow and motion labels. Early layered flow models [8, 15, 16] model the flow field as multiple motion layers, each representing a parametric motion field. To robustly optimize the different flow patterns, mixture flow models are proposed to compose the overall optical flow field with multiple simpler parametric flow fields. These methods usually assume a fixed number of clusters and simplified parametric forms of the individual flow component. Later, several works have found that clustering the orientation of the flow field leads to good segmentation results [3, 25].

These problems have been significantly improved with the advancement of neural networks, which provide the ability to learn motion and structure prior from a large amount of data. The most common way to approach the problem of estimating egomotion is to directly predict flow, depth, and egomotion [7, 32, 47, 54]. These quantities are related by the rigid motion field equation, and thus, geometric constraints can be used for joint optimization to improve overall performance. Zhu et al. [50] inserted a nondifferentiable RANSAC layer to allow explicit handling of nonrigid and/or independently moving objects in the scene. Casser et al. [5] modeled both camera ego-motion and objects motion model in 3D space; however, the 3D object motion estimator requires precomputed semantic segmentation masks as input, which are unavailable in most settings.

The incompatibility between independent motion and camera motion also creates opportunities for segmentation. Ranjan et al. [32] proposed an adversarial collaboration framework to explain and assign pixels to IMO or rigid backgrounds. Furthermore, informatic-theoretic approaches have been proposed to supervise segmentation networks by training an inpainter and a segmenter [45]. The motion segmenter predicts a foreground mask so that the inpainter cannot recover the masked foreground region from the background. On the other hand, the inpainter tries to inpaint the flow field using a background flow pattern. These works tend to work better on datasets with relatively simple camera motion and a single IMO. Another line of approaches related to our work is geometric self-labeling. Yang and Ramanan [44] trained a network to segment objects based on the error in the flow of the predicted scene. Zheng and Yang [48] refined pseudo labels by examining the uncertainty of semantic segmentation. Xie et al. [43] uses the Segment Anything Model (SAM) to assist with flow-based motion grouping.

### 3 Preliminaries

In this section, we geometrically define Independently Moving Objects (IMOs) in a 2D motion field. We consider the first-order instantaneous optical flow derived by Longuet-Higgins et al. [20]. For a point  $P = (X, Y, Z)$  that is observed by a camera  $C$  that moves instantaneously with linear velocity  $v$  and angular velocity  $\omega$ , its 3D motion field is written as:

$$\dot{P} = -\mathbf{v} - \omega \times \mathbf{P} = - \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (1)$$

Assuming a pinhole camera model, the point  $(X, Y, Z)$  is projected to  $(\frac{X}{Z}, \frac{Y}{Z})$ , whose derivative with respect to time is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} - \frac{\dot{Z}}{Z^2} \begin{bmatrix} X \\ Y \end{bmatrix}. \quad (2)$$

Plugging Equation 1 into Equation 2, we obtain the 2D motion field generated from point  $P$ :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ v_Z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}. \quad (3)$$

It can be seen that for an object moving in the camera frame with linear and angular velocity  $v_o$  and  $\omega_o$ , the combined motion field can be written as the sum of two motion fields  $\Psi(v_c, \omega_c, X, Y, Z)$  and  $\Psi(-v_o, -\omega_o, X, Y, Z)$ , as object velocity can be thought as the opposite of camera velocity. In the following sections, we slightly abuse the notation to write  $\Psi(x)$  to indicate the motion field of a 2D point  $x$  which inversely projects to point  $[X, Y, Z]$  in the camera frame. More generally, with multiple IMOs, the motion field can be written as:

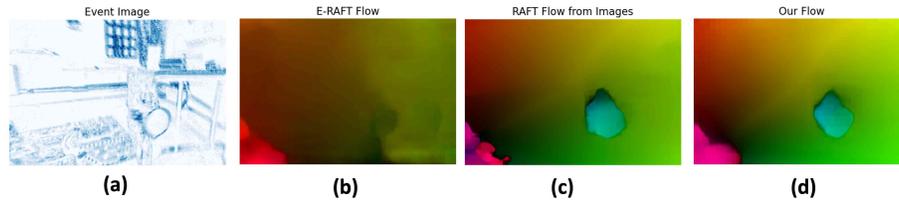
$$\Psi(x) = \Psi_{cam}(x) + \sum_i \Psi_{O_i}(x) \mathbb{1}[x \in O_i], \quad (4)$$

where  $O_i$  represents the  $i$ th object in the scene, where  $\cup_{i=1}^n O_i$  represents all independently moving points in the scene that can be observed in the camera.  $n$  is the total number of objects. Since the objects are assumed to be non-transparent, for each point observed by the camera, only one object contains this point:

$$\cap_{i=1}^n O_i = \emptyset. \quad (5)$$

From Equation 4, it can be seen that the objects and the camera have independent motion patterns. It is worth noting that previous literature usually models this as a mixture model [1] where the indicator function  $\mathbb{1}[x \in O_i]$  is replaced with a weight  $w_i$  and the camera motion field is weighted by  $w_{cam}$  such that  $w_{cam} + \sum_i w_i = 1$ . The weight  $w_i$  is a soft weight that indicates the likelihood that a point belongs to an object  $O_i$  or the camera. Similarly, Stoffregen et al. [38], Mitrokhin et al. [24], Zhou et al. [49] all employed this mixture formulation to enable segmentation among several candidate motion models. Either an Expectation-Maximization frame is used to optimize the weights directly, or a network is used to learn the mixture weights.

However, several underlying assumptions are made here to reduce the generalization ability of such approaches. First, such mixture models assume a fixed number of candidate models to initialize. These values cannot be easily tuned and depend heavily on the scene. In our experiments, we find the number of clusters cannot easily be selected without knowing beforehand the number of objects in the test sequence. Second, the mixture model makes strong assumptions about the parametric motion model. EMSGC [49] uses 4 to 12 parameter models on different scenes. EMMC [38] uses linear, rotational, 4-DOF and 8-DOF models. The most general model is EVIMO [24],



**Fig. 2:** (a): Events projected onto x-y space. (b): E-RAFT flow. (c): RAFT flow from Images. (d): our optical flow containing independent motion. Independent motions are clearly missing from E-RAFT. Flow fields are predicted on the wall test sequence of EVIMO. The color indicated direction. Best viewed in color.

which uses translational-only models for the object and a full rigid motion field for the camera.

In comparison, we deploy the exact formulation in Equation 4, and estimate the IMO motion weights directly through a per-pixel classification network, utilizing a discriminative power of a neural network over a large amount of data. This choice leads to a major challenge in event-based research, which is the lack of labeled data. In the next sections, we explain how we train the network without labeled motion masks.

## 4 Unsupervised Motion Segmentation

In Figure 1, we show the pipeline of Un-EVIMO. Generating motion labels on a large scale has been a challenging problem. The most scalable solution is collecting data in simulation [10, 22]. In video datasets such as DAVIS16 [30], the motion masks of objects are usually labeled by humans. In driving datasets that have high accuracy depth sensors, such as KITTI [13], IMOs are mostly cars. These objects are removed and inserted back using fitted car CAD models. In certain constrained cases, the labels can be generated by projecting known objects into the current camera frame. In EVIMO [24], the authors scanned the environment and objects before collecting dynamic motion. During data collection, VICON markers are attached to objects and cameras so that the relative poses between the camera, objects, and room are known. The object masks are then subsequently obtained by projecting the 3D model of the object onto the current camera. Despite this automatic labeling scheme, the amount of work required to calibrate the system and provide high-quality object scans makes this supervising method not transferable to general scenes.

In this section, we propose a framework for automatically obtaining labels taking advantage of the results of the CNN-based optical flow [12, 39, 46, 52, 53] estimation. The event-based optical flow networks are usually trained with large-scale event-based dataset [6, 11, 51]. Our method is based on geometric error rather than on the semantics of the objects, which allows it to be applied on a large scale. We explain how roughly accurate labels can be generated only using depth and camera data. In addition, we describe how we train a robust event-based motion segmentation network completely without human annotation. Our pipeline is mainly composed of two parts: a robust

**Table 2:** Quantitative Evaluation on EVIMO. Event-masked IoU on predicted masks and gt masks is calculated as described in Eqn. 13. Our method compares favorably with EMSGC, which is the only one other than Un-EVIMO that does not need labels. Our method performs competitively with other supervised methods. “Baseline CNN” is our network-trained ground truth masks. EMSGC requires per-scene parameter tuning. For fair comparisons, we take the top 30 and 50 percent of EMSGC IoU.

	Table	Box	Floor	Plain Wall	Fast Motion
<b>Supervised Methods</b>					
Baseline CNN	66±23	50±23	<b>74±13</b>	60±20	52±24
Motion-blurred Video	24±25	28±30	40±25	30±26	14±18
EVIMO [24]	<b>79±6</b>	<b>70±5</b>	59±9	78±5	<b>67±3</b>
EVDodgeNet [36]	70±8	67±8	61±6	72±9	60±10
SpikeMS [29]	50±8	65±8	53±16	63±6	38±10
GConv [23]	51±16	60±18	55±19	<b>80±7</b>	39±19
<b>Unsupervised Methods</b>					
EMSGC [49] Top 30%	<b>55±17</b>	24±28	18±29	24±33	43±27
EMSGC [49] Top 50%	36±27	14±25	11±24	15±28	26±29
Un-EVIMO (Ours)	50±21	<b>45±24</b>	<b>56±15</b>	<b>53±19</b>	<b>44±21</b>

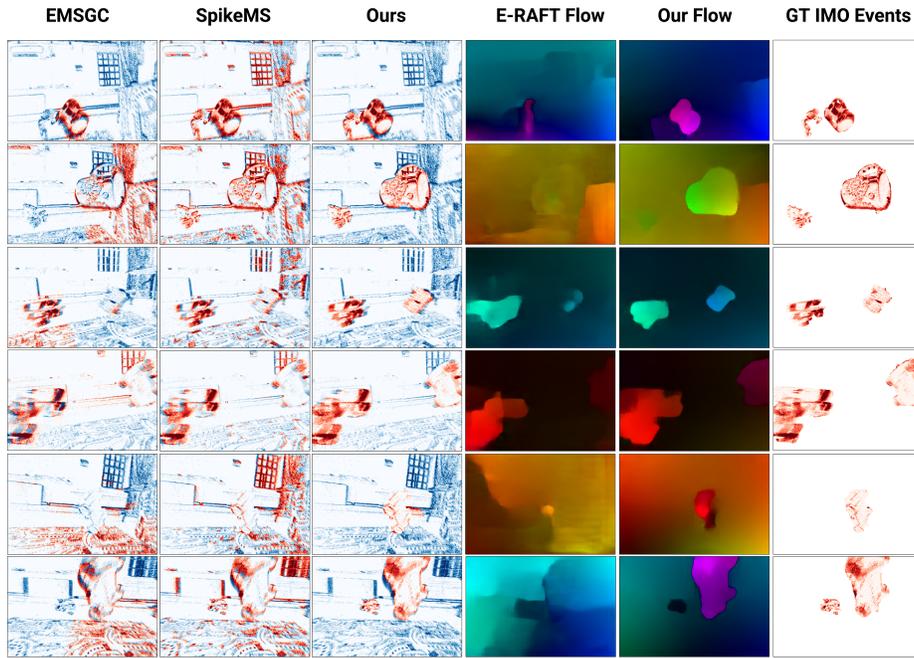
**Table 3:** Optical flow comparison. E-RAFT underperforms when there is independent motion. We report EPE metric as described in E-RAFT [12].

	Table	Box	Floor	Wall	Fast
E-RAFT [12]	11.150	14.902	4.983	8.036	20.471
Ours	<b>1.550</b>	<b>3.432</b>	<b>1.036</b>	<b>2.062</b>	<b>5.331</b>

pseudo-label generation module and an event motion segmentation network. The data required for training is only the depth map in the camera frame. The depth information is only used during training in our geometry-based pseudo-label generation module. Such data are not required during inference. Instead, we train a per-pixel classifier that takes in events and produces a binary segmentation mask.

#### 4.1 Optical Flow with Independent Motion

The high temporal resolution of the events preserves rich temporal information in x-y-t space, which allows robust estimation of optical flow under various challenging conditions. Early work achieves this estimation by plane fitting [2], which produces an event-based optical flow only on regions with events. EV-FlowNet [52] and E-RAFT [12] are trained neural networks that learn the dense optical flow from events. In our formulation, it is critical to have dense flow predictions in order to compute the residual error between camera motion and the observed flow field. In this work, we used the E-RAFT flow network pretrained on DSEC. We fine-tuned the flow on the predicted flow from grayscale images using RAFT [39]. In Figure 2, we show examples of three types of



**Fig. 3: Columns 1 to 3:** Segmentation results of EMSGC, SpikeMS, Un-EVIMO and EVIMO-Supervised. **Columns 4 to 5:** E-RAFT flow output (trained in DSEC) and our fine-tuned flow network. **Column 6:** Segmented IMO event using ground truth. It can be seen that Un-EVIMO produces sharper and more consistent masks than the baseline methods.

optical flow. RAFT [39] is the state-of-the-art optical flow method for images. E-RAFT extends the RAFT framework to events. It can be seen that our fine-tuned flow correctly estimates the flow for IMO objects. This is consistent with the discovery of Shiba et al. that E-RAFT performs poorly on independently moving objects [37].

*Optical Flow with Independent Motion* Flow networks trained on driving data cannot be easily used for IMO detection. To show this, we compared our optical flow results with the state-of-the-art E-RAFT models pre-trained on DSEC [11]. For this evaluation, we used the architecture of E-RAFT as is and only fine-tune the flow based on image-based flow. Since the ground-truth optical flow of EVIMO is not provided, we supervised the high-quality optical flow computed using RAFT [39] with photometric matching and refinement. In Table 3, we compare our fine-tuned flow with pre-trained E-RAFT flow on unseen test sequences in EVIMO using RAFT flow as ground truth. In our experiments, we observe that the performance gap between our Un-EVIMO flow network and E-RAFT is tightly correlated with the dynamic of the scene. In our experiments, due to the missing IMOs, the E-RAFT baseline cannot provide good pseudo-labels for training the downstream network.

## 4.2 Robust Camera Motion Estimation

Traditionally, the motion segmentation problem can be seen as a chicken-and-egg problem because IMO can significantly bias camera motion estimation if they are not properly filtered. Several self-supervised methods for joint motion estimation approaches are susceptible to this problem. For example, Zhu et al. [53] jointly learned egomotion, depth, and flow assuming rigid scenes, which is dependent on a network to ignore independent motions. E-RAFT [12], although it does not learn ego-motion directly, has been shown to underperform in independent motion regions [37]. Thus, a robust camera motion module needs to be designed to avoid further blurring of the decision boundary between IMO motion and camera motion. To this end, we take advantage of the classical outlier rejection techniques and use Random Sample Consensus (RANSAC) to estimate camera motion. In general, RANSAC is used to solve the following problem:

$$\theta = \arg \min_{\theta} \sum_{i=1}^N \rho(\epsilon(u_i; \theta)), \quad (6)$$

where  $\epsilon$  is an error function,  $\rho$  is a robust likelihood function,  $N$  is the total number of observations, and  $u_i$  is the observed motion field at pixel  $i$  with respect to the camera motion given the velocity  $\theta$ . We notice that the error term  $\epsilon(u_i; \theta)$  corresponds exactly to  $\sum_i \Psi_{O_i}(x) \mathbb{1}[x \in O_i]$ , the second term in Equation 4. A naive optimization without outlier rejection will bias the motion estimation towards the motion of near and fast-moving objects. Based on Equation 2, the camera motion  $(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$  can be solved by the linear equation:

$$\begin{bmatrix} -1/z_1 & 0 & x_1/z_1 & x_1 y_1 & -(1+x_1^2) & y_1 \\ & & & \vdots & & \\ 1/z_n & 0 & x_n/z_n & x_n y_n & -(1+x_n^2) & y_n \\ 0 & -z_1^{-1} & y_1/z_1 & 1+y_1^2 & -x_1 y_1 & -x_1 \\ & & & \vdots & & \\ 0 & -z_n^{-1} & y_n/z_n & n+y_n^2 & -x_n y_n & -x_n \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \\ \dot{y}_1 \\ \vdots \\ \dot{y}_n \end{bmatrix}, \quad (7)$$

where  $z_i, x_i, y_i$  are the depth values (input) and the pixel coordinates of the  $i$ th pixel and  $(\dot{x}_i, \dot{y}_i)$  is the calibrated optical flow from events. We sample 3 points every time to solve the equation for a maximum of 300 iterations, or a stop probability of 0.999 is reached. Then we use all inlier pixels to solve the over-constrained least square problem using SVD. We present the quantitative pose estimation results in Table 4. Our average translational error in relative pose estimation is sub-centimeter in Table and Floor sequences. The error is 4 for the extremely challenging Fast sequence. This shows the robustness of our pose tracking method that fuses an accurate flow method with robust geometric estimation method.

## 4.3 Adaptive Geometry-based Thresholding

We combine accurate flow estimation from events and robust motion estimation to produce a residual flow field. In contrast to model-based approaches in previous event-based motion segmentation works, we do not assume a fixed number of parametric

**Table 4:** Relative camera pose estimation using flow displacement. The translational error is defined as the mean squared error between the estimated and ground truth camera positions. The rotation error is defined as  $\log_m(R_{gt}^T R_{pred})$ .

	Table	Box	Floor	Wall	Fast
Trans. (m)	0.0082	0.0251	0.0075	0.0141	0.0416
Rot. (rad)	0.0348	0.0412	0.0261	0.0296	0.1110

flow models. In Section 5, we show failure cases of parametric flow due to the high variation of motion and depth in real data. Since no competing models are learned or optimized, selecting an appropriate threshold for the magnitude of the residual flow becomes a crucial step. In analyzing the data, we find that the error usually demonstrates a bimodal distribution, where one peak corresponds to the correct rigid motion, and the other model concentrates at a much higher mean. Since there is usually no fixed threshold value due to the variation of noise and depth, we adopt a statistically robust thresholding method based on Otsu’s method [27].

Given a set of pixels  $\Lambda = \{q_i\}$ , the residual flow function for each pixel is predicted by computing the  $l^2$  norm of the residual flow:  $r(q_i) = \|\Psi(q_i) - \Psi_{cam}(q_i)\|_2$ . Modeling the residual  $r(q_i)$  as a bimodal distribution, choosing a threshold  $\hat{r}$  is treated as the problem of maximizing the variance between the two classes. The two classes, by definition, are rigid areas and IMO areas. IMO areas have higher residual flow because they have different velocities than the camera. The problem can be solved efficiently with a simple 1D search if we define  $R = \{r_j\}$  as the set of candidate solutions. The objective of the search is

$$\arg \max_{r_j \in R} \sum_{k=0}^{r_j} P_k (\mu_{bg}(r_j) - \mu)^2 + \sum_{k=r_j}^{K_{max}} P_k (\mu_{imo}(r_j) - \mu)^2 \quad (8)$$

$$\mu = \sum_{k=0}^{K_{max}} P_k k \quad \mu_{bg}(r_j) = \sum_{k=0}^{r_j} P_k k \quad \mu_{imo}(r_j) = \sum_{k=r_j}^{K_{max}} P_k k. \quad (9)$$

$P_k$  is the probability that a pixel  $q_i$  falls into the bin  $k$ . We use 256 bins for this problem, and the histogram is clipped at 10 pixels. In our search, we applied a two-stage filter on Otsu’s thresholding results. First, we examine the total variance of the histogram of errors; If the variance is greater than some threshold  $\epsilon_{var}$ , we do not look at this slice of events, since the flow prediction does not provide clear boundaries of the objects. Similarly, we compute the variance between IMO pixels and BG pixels, based on the selected threshold  $r_j$  and remove the training example if this value is too small. These two calculated variance values can be seen as a measure of confidence in the labels. Selecting confident labels is a crucial step in pseudo-label selection.

#### 4.4 Optional Depth Input

To compute the optical flow, we take an optional depth map as input in Equation 2. In practice, the depth map can be acquired with a paired sensor or monocular depth

network. This depth map is **only used in training** for generating the pseudo labels and **never used during actual inference** after the network has been trained. Alternatively, we can use parametric flow independent of the depth. In [21], several parametric depth models are proposed. We take the 12-DOF biquadratic flow as an example. The flow for each pixel is defined as:

$$x' = q_{x'x^2}x^2 + q_{x'xy}xy + q_{x'y^2}y^2 + q_{x'xx}x + q_{x'yy}y + q_{x'} \quad (10)$$

$$y' = q_{y'x^2}x^2 + q_{y'xy}xy + q_{y'y^2}y^2 + q_{y'xx}x + q_{y'yy}y + q_{y'}. \quad (11)$$

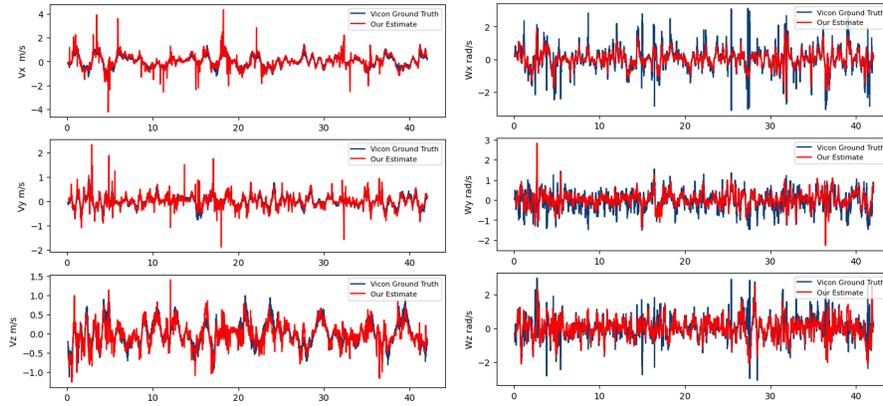
The estimation of the camera-induced camera motion can be modeled as follows:

$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \\ x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ & & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} q_{x'x^2} \\ q_{x'xy} \\ q_{x'y^2} \\ q_{x'xx} \\ q_{x'yy} \\ q_{x'} \\ q_{y'x^2} \\ q_{y'xy} \\ q_{y'y^2} \\ q_{y'xx} \\ q_{y'yy} \\ q_{y'} \end{bmatrix} = \begin{bmatrix} x'_1 \\ \vdots \\ x'_n \\ y'_1 \\ \vdots \\ y'_n \end{bmatrix}. \quad (12)$$

The estimation problem can be solved with RANSAC with six points. We present these results in Table 6 as ablation studies to provide how much the choice of flow modeling affects the segmentation performance.

#### 4.5 Event-based Motion Segmentation Network

It can be seen from our pseudo-label generation framework that the task of independent motion segmentation can be seen as a combination of global and local motion estimation. As previously studied in the event-based flow literature [12, 52], it is preferred to preserve motion information in events. For this purpose, we use the event volume representation, which encodes the temporal domain as discretized channels of a 3D tensor. A bi-linear interpolation kernel( $k_b$ ) is used to distribute events to discretized bins based on their spatio-temporal proximity with these bins. We use the volume of events, which has been shown to be effective in understanding motion, as described in [40, 41, 53]:  $E(x, y, t) = \sum_i p_i k_b(x - x_i) k_b(y - y_i) k_b(t - t_i^*)$ . We use 15 channels for the event volume to allow the network to extract fine temporal information from events. We provide details on the implementation of the network and the loss functions of Un-EVIMO. Our trained prediction module is a UNet-like convolutional neural network. The bottleneck layers facilitate the aggregation of global features, since the segmentation problem relies not only on the local flow pattern of events but also on the global motion pattern caused by the camera. We use a pre-trained ResNet34 [14] encoder with pre-trained weights on ImageNet [9]. Since objects usually occupy much less space than the rigid



**Fig. 4:** Estimated camera rotation from estimated optical flow. The results are shown for the whole evaluation sequence *wall\_00*. Best viewed in color. The left and right columns show translational and rotational error respectively.

background, we use a Focal Loss [19] to handle the class imbalance problem. The network is trained with an Adam optimizer using a learning rate of  $2e - 4$  on EVIMO Table, Wall, Floor, Box, and Fast training sequences.

## 5 Experiments

**Quantitative Evaluation** In Table 2, we report the IoU our Un-EVIMO against competing methods on different classes of EVIMO. The IoU is computed on masked events directly in order to compare with single-event labeling approaches. The IoU score is computed as:

$$\text{IoU}(O_t, P_t, E_t) = \frac{|(E_t \cap P_t) \cap (E_t \cap O_t)|}{|(E_t \cap P_t) \cup (E_t \cap O_t)|}, \quad (13)$$

where  $E_t$  is the set of projected events surrounding time  $t$ .  $P_t$  and  $O_t$  are the projected mask and ground truth in 2D.  $E_t$ ,  $P_t$ , and  $O_t$  are all subsets of all pixels. The comparison is evaluated at 40Hz, which is the default evaluation frequency for the dataset. Comparison methods can be divided into two classes: supervised and optimization-based. In supervised methods, a mask of a moving object is provided at each time. On the other hand, EMSGC in the table is an optimization-based method, which does not use mask labels. Instead, multiple motion models are fitted to the events by alternating between contrast maximization and flow fitting. We note that EMSGC is sensitive to parameters such as the class of the parametric model and the number of objects. Due to the large amount of evaluation data (thousands of frames per sequence), we were unable to tune the parameters for each slice. Instead, we tuned for each sequence and generously reported the top 30 performance to give it a fair comparison. This further emphasizes the advantages of our pseudo-label-based method over per-event-slice optimization. See supplemental for details. Our model outperforms the supervised spiking method and

**Table 5:** Analysis of data processing (Pre), algorithm running (Run), and total time.

	<b>Ours</b>	<b>SpikeMS [28]</b>	<b>GConv [23]</b>	<b>EVIMO [24]</b>	<b>EMSGC [49]</b>
Pre (ms)	<b>3.35</b>	10.56	698.62	16.74	33.38
Run (ms)	<b>3.22</b>	110.01	16.00	4.29	9496.04
Total (ms)	<b>6.57</b>	120.57	715.62	26.85	9529.42

unsupervised EMSGC (with per-sequence tuning). Un-EVIMO is comparable to supervised methods on tables, floor, wall, and fast motion. Compared to supervised methods, Un-EVIMO predicts less sharp boundaries because the network is trained with noisy labels. To demonstrate the difficulty of the task when using frame cameras in low-light conditions, we include experiments with synthetic motion-blurred videos. We used SuperSlowMo to upsample videos to 640 fps and averaged frames to synthesize motion-blurred videos with a 0.125-second shutter time. We trained a supervised network on regular videos and tested it with motion-blurred videos.

**Qualitative Evaluation** In Figure 3, we provide qualitative examples of competing methods on the Wall sequence of the evaluation set. We show examples using methods whose source code is available. Our results are very similar in quality compared with supervised CNN methods, largely outperform optimization-based methods, and even outperforms supervised SNNs. SpikeMS tends to sparsify the events and keep edges. EMSGC needs extensive tuning to get reasonable results. However, it still misclassifies IMO as rigid areas. With these noise predictions across the image from SpikeMS and EMSGC, IMO cannot be easily detected and handled, while our network produces spatially consistent segmentations.

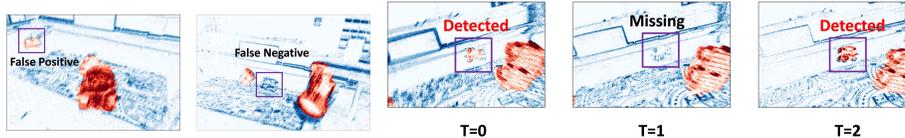
**Computational Speed** Un-EVIMO trains a single feed-forward U-Net for IMO segmentation. Therefore, no heavy optimization is needed. In Table 5, we show the computational time comparison between our method and the baseline methods. GConv [23] uses a fast network, but the graph building operation takes a significant amount of time. EMSGC [49] is faster in building the graph, but the per-event-slice optimization is extremely slow and is not guaranteed to converge. Our method shifts the computational burden to computing the pseudo-labels and enable single forward pass during inference. Our compute platform is a single RTX 3080 mobile GPU with an 8-core cpu.

## 6 Ablation Studies

We provide two ablation results, as shown in Table 6. First, we compare the IMO labels computed with the un-refined flow directly predicted from E-RAFT. The off-the-shelf flow network is trained on DSEC, which has limited independent motion. Second, we show two results using parametric flow models (**b** and **c**) based on the definitions defined in previous work [21]. The parametric flow models provide the possibility without the input depth. With depth models with fewer degrees of freedom, the IMO labeling scheme still outperforms EMSGC. However, the best performance so far is the full model with input depth, which maximizes the IMO pseudo labeling quality. We leave the elimination of these dependencies to our future work.

**Table 6: Ablation Studies.** (a) refers to using unrefined pre-trained E-RAFT flow network. (b,c) shows results using parametric flow models described in [21].

	Table	Box	Floor	Wall	Fast
(a)ERAFT	32±23	28±21	35±19	42±22	27±23
(b)6-DOF	43±26	42±25	51±21	47±23	37±24
(c)12-DOF	47±24	40±25	56±18	49±22	37±25
Ours	<b>50±21</b>	<b>45 ± 24</b>	<b>56±15</b>	<b>53±19</b>	<b>44±21</b>



(a) Failure cases of our method. On the left, the network incorrectly classifies a static square pattern on the ground as IMO. On the right, the network fails to find the apparent IMO in the scene.

(b) IMO predictions at three consecutive event slices. Our IMO detection runs on single slices of events. Occasional erroneous predictions do not have temporal consistency with the previous and next predictions.

## 7 Failure Cases and Limitations

In Figure 5a, we show one false positive and one false negative output from our approach. Due to the extreme dynamic nature of the dataset, the residual between the background flow and the IMO flow is small. In particular, there are cases where the objects have near-zero velocities. These objects should be segmented if we consider its past motion, but should be excluded if we only look at current motion. This leads to the lack of temporal consistency in the prediction. A possible solution is adding constraints between the current IMO mask and the immediate past IMO masks during training. This could be applied during the pseudo-label generation phase too for better ground truth. In Figure 5b, we demonstrate that adding temporal consistency can be helpful. The network lost track of the IMO at time, but it should know that an IMO is nearby by looking at the previous several mask predictions. A discontinuity in prediction should be penalized because the motion of an object can be seen as continuous in the events.

## 8 Conclusion

In this work, we tackle the problem of event-based segmentation from a geometric point of view. We focus on the major problem of event-based motion segmentation, which is the lack of labeled segmentation masks. Instead of using clustering techniques that require a fixed number of clusters and simplified parametric flow, our approach is purely geometric and robust to unseen semantic classes. Using the accurate event-based optical flow, we generated pseudo-labels based on the residual flow field defined by the difference between the estimated ego-motion field and the general motion field. Ego-motion field was predicted using depth and a pre-trained flow network. With experiments on the EVIMO dataset, we show that our framework can be used to train downstream motion segmentation to perform competitively with supervised methods.

**Acknowledgment:** We gratefully acknowledge the support by the following grants: NSF FRR 2220868, NSF IISRI 2212433, NSF TRIPODS 1934960, and ONR N0001422-1-2677

## References

1. Ayer, S., Sawhney, H.S.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In: Proceedings of IEEE International Conference on Computer Vision. pp. 777–784. IEEE (1995) [5](#)
2. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.H., Bartolozzi, C.: Event-based visual flow. *IEEE transactions on neural networks and learning systems* **25**(2), 407–417 (2013) [7](#)
3. Bideau, P., Learned-Miller, E.: It’s moving! a probabilistic model for causal motion segmentation in moving camera videos. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. pp. 433–449. Springer (2016) [4](#)
4. Burner, L., Mitrokhin, A., Fermüller, C., Aloimonos, Y.: 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) [3](#)
5. Casser, V., Pirk, S., Mahjourian, R., Angelova, A.: Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 8001–8008 (2019) [4](#)
6. Chaney, K., Cladera, F., Wang, Z., Bisulco, A., Hsieh, M.A., Korpela, C., Kumar, V., Taylor, C.J., Daniilidis, K.: M3ed: Multi-robot, multi-sensor, multi-environment event dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4016–4023 (2023) [6](#)
7. Chen, Y., Schmid, C., Sminchisescu, C.: Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7063–7072 (2019) [4](#)
8. Darrell, T., Pentland, A.: Robust estimation of a multi-layered motion representation. In: Proceedings of the IEEE Workshop on Visual Motion. pp. 173–174. IEEE Computer Society (1991) [4](#)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) [11](#)
10. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV) (2015), <http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15> [6](#)
11. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters* **6**(3), 4947–4954 (2021) [6](#), [8](#)
12. Gehrig, M., Millhäusler, M., Gehrig, D., Scaramuzza, D.: E-raft: Dense optical flow from event cameras. In: 2021 International Conference on 3D Vision (3DV). pp. 197–206. IEEE (2021) [6](#), [7](#), [9](#), [11](#)
13. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* **32**(11), 1231–1237 (2013) [6](#)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) [11](#)

15. Ju, S.X., Black, M.J., Jepson, A.D.: Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In: Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 307–314. IEEE (1996) [4](#)
16. Kumar, M.P., Torr, P.H., Zisserman, A.: Learning layered motion segmentations of video. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 1, pp. 33–40. IEEE (2005) [4](#)
17. Layton, O.W., Fajen, B.R.: A neural model of mst and mt explains perceived object motion during self-motion. *Journal of Neuroscience* **36**(31), 8093–8102 (2016) [1](#)
18. Li, M., Wang, Y.X., Ramanan, D.: Towards streaming perception. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 473–488. Springer (2020) [1](#)
19. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017) [12](#)
20. Longuet-Higgins, H.C., Prazdny, K.: The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences* **208**(1173), 385–397 (1980) [4](#)
21. Mann, S., Picard, R.W.: Video orbits of the projective group a simple approach to featureless estimation of parameters. *IEEE Transactions on Image Processing* **6**(9), 1281–1295 (1997) [11](#), [13](#), [14](#)
22. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016) [6](#)
23. Mitrokhin, A., Hua, Z., Fermuller, C., Aloimonos, Y.: Learning visual motion segmentation using event surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14414–14423 (2020) [3](#), [7](#), [13](#)
24. Mitrokhin, A., Ye, C., Fermüller, C., Aloimonos, Y., Delbruck, T.: Ev-imo: Motion segmentation dataset and learning pipeline for event cameras. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6105–6112. IEEE (2019) [2](#), [5](#), [6](#), [7](#), [13](#)
25. Narayana, M., Hanson, A., Learned-Miller, E.: Coherent motion segmentation in moving camera videos using optical flow orientations. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1577–1584 (2013) [4](#)
26. Nordström, K., Barnett, P.D., O’Carroll, D.C.: Insect detection of small targets moving in visual clutter. *PLoS biology* **4**(3), e54 (2006) [1](#), [2](#)
27. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* **9**(1), 62–66 (1979) [10](#)
28. Parameshwara, C.M., Li, S., Fermüller, C., Sanket, N.J., Evanusa, M.S., Aloimonos, Y.: Spikems: Deep spiking neural network for motion segmentation. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3414–3420. IEEE (2021) [13](#)
29. Parameshwara, C.M., Sanket, N.J., Singh, C.D., Fermüller, C., Aloimonos, Y.: 0-mms: Zero-shot multi-motion segmentation with a monocular event camera. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 9594–9600. IEEE (2021) [3](#), [7](#)
30. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: *Computer Vision and Pattern Recognition* (2016) [6](#)
31. Pitzalis, S., Fattori, P., Galletti, C.: The functional role of the medial motion area v6. *Frontiers in behavioral neuroscience* **6**, 91 (2013) [1](#)

32. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12240–12249 (2019) [3](#), [4](#)
33. Raudies, F., Neumann, H.: Modeling heading and path perception from optic flow in the case of independently moving objects. *Frontiers in behavioral neuroscience* **7**, 23 (2013) [1](#)
34. Royden, C.S., Connors, E.M.: The detection of moving objects by moving observers. *Vision research* **50**(11), 1014–1024 (2010) [1](#)
35. Rushton, S.K., Warren, P.A.: Moving observers, relative retinal motion and the detection of object movement. *Current Biology* **15**(14), R542–R543 (2005) [1](#)
36. Sanket, N.J., Parameshwara, C.M., Singh, C.D., Kuruttukulam, A.V., Fermüller, C., Scaramuzza, D., Aloimonos, Y.: Evidodgenet: Deep dynamic obstacle dodging with event cameras. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 10651–10657. IEEE (2020) [3](#), [7](#)
37. Shiba, S., Aoki, Y., Gallego, G.: Secrets of event-based optical flow. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII. pp. 628–645. Springer (2022) [8](#), [9](#)
38. Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., Scaramuzza, D.: Event-based motion segmentation by motion compensation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7244–7253 (2019) [3](#), [5](#)
39. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020) [6](#), [7](#), [8](#)
40. Wang, Z., Chaney, K., Daniilidis, K.: Evac3d: From event-based apparent contours to 3d models via continuous visual hulls. In: European conference on computer vision. pp. 284–299. Springer (2022) [11](#)
41. Wang, Z., Ojeda, F.C., Bisulco, A., Lee, D., Taylor, C.J., Daniilidis, K., Hsieh, M.A., Lee, D.D., Isler, V.: Ev-catcher: High-speed object catching using low-latency event-based neural networks. *IEEE Robotics and Automation Letters* **7**(4), 8737–8744 (2022) [11](#)
42. Wu, Q., Zhang, Y.: Neural circuit mechanisms involved in animals’ detection of and response to visual threats. *Neuroscience Bulletin* pp. 1–15 (2023) [1](#), [2](#)
43. Xie, J., Yang, C., Xie, W., Zisserman, A.: Moving object segmentation: All you need is sam (and flow). *arXiv preprint arXiv:2404.12389* (2024) [4](#)
44. Yang, G., Ramanan, D.: Learning to segment rigid motions from two frames. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1266–1275 (June 2021) [4](#)
45. Yang, Y., Loquercio, A., Scaramuzza, D., Soatto, S.: Unsupervised moving object detection via contextual information separation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 879–888 (2019) [4](#)
46. Ye, C., Mitrokhin, A., Fermüller, C., Yorke, J.A., Aloimonos, Y.: Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5831–5838. IEEE (2020) [6](#)
47. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1983–1992 (2018) [4](#)
48. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision* **129**(4), 1106–1120 (2021) [4](#)

49. Zhou, Y., Gallego, G., Lu, X., Liu, S., Shen, S.: Event-based motion segmentation with spatio-temporal graph cuts. *IEEE Transactions on Neural Networks and Learning Systems* (2021) [3](#), [5](#), [7](#), [13](#)
50. Zhu, A.Z., Liu, W., Wang, Z., Kumar, V., Daniilidis, K.: Robustness meets deep learning: An end-to-end hybrid pipeline for unsupervised learning of egomotion. *arXiv preprint arXiv:1812.08351* (2018) [4](#)
51. Zhu, A.Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., Daniilidis, K.: The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters* **3**(3), 2032–2039 (2018) [6](#)
52. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898* (2018) [6](#), [7](#), [11](#)
53. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 989–997 (2019) [6](#), [9](#), [11](#)
54. Zou, Y., Luo, Z., Huang, J.B.: Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 36–53 (2018) [4](#)