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

Ziyun Wang[1] Jinyuan Guo[1] Kostas Daniilidis[1,2]

[1] University of Pennsylvania [2] 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. Although event cameras have recently shown competitive performance in unsupervised optical flow estimation, 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 have been heavily dependent on labeled data. However, biological vision systems have developed the ability to avoid moving objects through daily tasks without being given 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 handle an arbitrary number of not predetermined objects and is easily scalable to datasets where expensive IMO labels are not readily available. We evaluate our approach on the EVIMO dataset and show that it performs competitively with supervised methods, both quantitatively and qualitatively. Please see our project website for more details:* https://www.cis.upenn.edu/~ziyunw/un_evmoseg/.

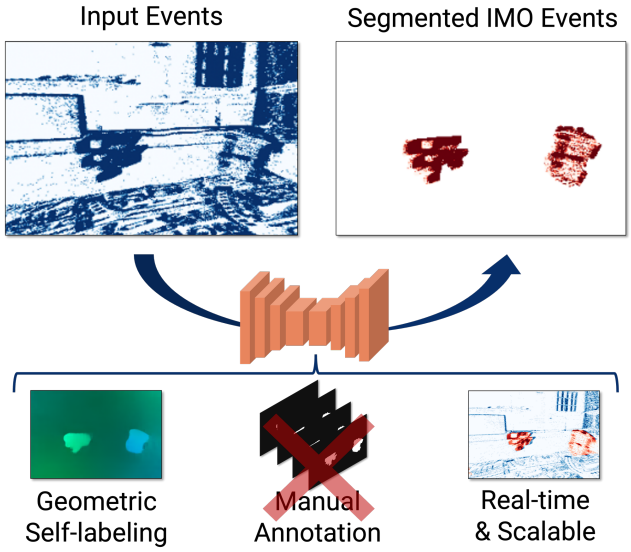



Figure 1. We propose a novel framework for training event-based motion segmentation networks. Our method does not require any manually labeled IMOs. Instead, a geometric labeling approach is used to enable scalable training. The event IMO segmentation network runs real-time and can handle various conditions without extensive parameter tuning.

## 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 [24]. Cross-species studies have found that biological systems have neurons that specialize in detecting looming motion, a special case of independent motion [38]. Scientists have found that certain parts of the visual field are involved in subtracting out self-motion to help identify moving objects [29]. In cognitive science, the ability to model or segment independently moving objects has been extensively studied [16, 31–33]. 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 [17]. High-accuracy depth sensors, e.g. LIDAR, are able to map rigid scenes but have to apply semantic segmentation in order to detect IMOs.

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.

Recently, CNN-based approaches have shown success in

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

Table 1. Feature comparisons. Un-EvMoSeg 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.

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 [24, 38]. 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-EvMoSeg 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-EvMoSeg 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 [21] is a motion segmentation data set that contains more 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. Evdodgenet [34] 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. [36] propose an Expectation-Maximization framework that assigns events to different motion clusters by optimizing the event-based contrast maximization. EMSGC [44] 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. [22] use a graph neural network to learn the segmentation masks directly in the event point space. ConvGC [22] use a graph neural network to learn event-based segmentation on graphs constructed on downsampled events. SpikeMs [27] 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 [30]. In classical computer vision, motion segmentation is solved by optimization that simultaneously estimates parametric flow and motion labels. Early layered flow models [7, 14, 15] 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, 23].

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 ego-motion is to directly predict flow,

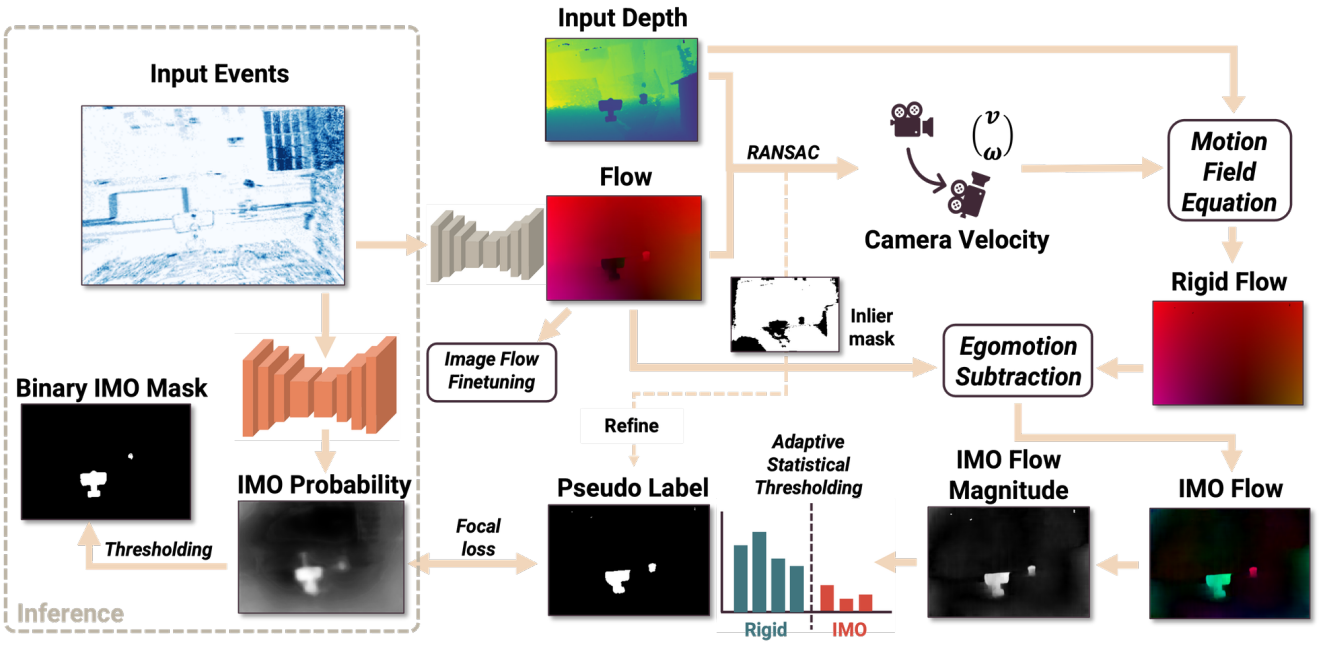



Figure 2. Pipeline of Un-EvMoSeg. **Left Dotted Box**: we train a network to directly predict IMO masks from events. **Rest of Figure**: we use a geometric self-labeling method to generate binary IMO pseudo-labels that supervise the IMO segmentation network. Our framework uses off-the-shelf optical flow (fine-tuned on image-based flow) and input depth. The camera motion fitted from flow and depth through RANSAC is used to compute rigid flow from the camera only. Pseudo-labels are generated through adaptive thresholding techniques based on the magnitude of estimated IMO motion field. Running inference Un-EvMoSeg is simple without parameter turning because while the training process requires geometry-based labels, only events are used for prediction. 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.

depth, and egomotion [6, 30, 42, 48]. 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. [45] inserted a nondifferentiable RANSAC layer to allow explicit handling of nonrigid and/or independently moving objects in the scene. Casser et al. [5] model 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. [30] propose an adversarial collaboration framework to explain and assign pixels to IMO or rigid backgrounds. Furthermore, informatic-theoretic approaches are proposed to supervise segmentation networks by training an inpainter and a segmenter [40]. 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 approach that is related to our work is geometric self-labeling. Yang and Ramanan [39] train a network to segment objects based on the error in the flow of the predicted scene. Zheng and Yang [43] refine pseudo-labels by looking at the uncertainty of semantic segmentation.

## 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. [19]. 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:

$$\dot{P} = -v - \omega \times P \tag{1}$$

$$= -\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} - \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2}$$

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:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} - \frac{\dot{Z}}{Z^2} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{3}$$

Plugging Equation 2 into Equation 3, we obtain the 2D motion field generated from point $P$:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \frac{1}{Z}\begin{bmatrix}-1 & 0 & x\\ 0 & -1 & y\end{bmatrix}\begin{bmatrix}v_X\\ v_Y\\ v_Z\end{bmatrix} + \tag{4}$$

$$\begin{bmatrix}xy & -(1+x^2) & y\\ 1+y^2 & -xy & -x\end{bmatrix}\begin{bmatrix}\omega_X\\ \omega_Y\\ \omega_Z\end{bmatrix} \tag{5}$$

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:

$$\Psi(x) = \Psi_{cam}(x) + \sum_i \Psi_{O_i}(x)\mathbb{1}[x \in O_i] \tag{6}$$

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. Since the objects are assumed to be non-transparent, for each point observed by the camera, only one object contains this point:

$$\cap_{i=1}^{n} O_i = \emptyset \tag{7}$$

From Equation 6, it can be seen that the objects and the camera have independent motion patterns. It it 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. [36], Mitrokhin et al. [21], Zhou et al. [44] 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.

| | Table | Box | Floor | Wall | Fast |
|---|---|---|---|---|---|
| E-RAFT [11] | 11.150 | 14.902 | 4.983 | 8.036 | 20.471 |
| Ours | **1.550** | **3.432** | **1.036** | **2.062** | **5.331** |

Table 2. Optical flow comparison. E-RAFT underperforms when there is independent motion. We report EPE metric as described in E-RAFT [11].

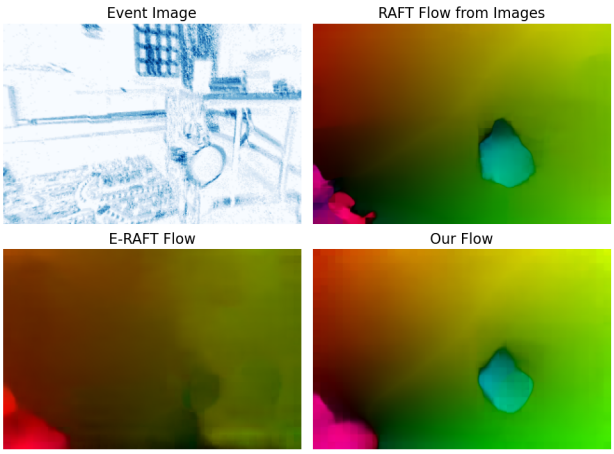



Figure 3. **Top Left**: Events projected onto x-y space. **Top Right**: RAFT flow from images. **Bottom Left**: E-RAFT flow. **Bottom Right**: our optical flow containing independent motion. Independent motions are clearly missing from E-RAFT. Flow fields are predicted on the wall test sequence of EVIMO. The color indicated direction. Best viewed in color.

EMSGC [44] uses 4 to 12 parameter models on different scenes. EMMC [36] uses linear, rotational, 4-DOF and 8-DOF models. The most general model is EVIMO [21], 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 6, 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 2, we show the pipeline of Un-EvMoSeg. Generating motion labels on a large scale has a been a challenging problem. The most scalable solution is collecting data in simulation [9, 20]. In video datasets such as DAVIS16 [28], the motion masks of objects are usually labeled by humans.

| | Table | Box | Floor | Plain Wall | Fast Motion |
|---|---|---|---|---|---|
| | **Supervised** | | | | |
| Baseline CNN | 66±23 | 50±23 | **74±13** | 60±20 | 52±24 |
| EVIMO [21] | **79±6** | **70±5** | 59±9 | 78±5 | 67±3 |
| EVDodgeNet [34] | 70±8 | 67±8 | 61±6 | 72±9 | **60±10** |
| SpikeMS [27] | 50±8 | 65±8 | 53±16 | 63±6 | 38±10 |
| GConv [22] | 51±16 | 60±18 | 55±19 | **80±7** | 39±19 |
| | **Unsupervised** | | | | |
| EMSGC [44] Top 30% | **55±17** | 24±28 | 18±29 | 24±33 | 43±27 |
| EMSGC [44] Top 50% | 36±27 | 14±25 | 11±24 | 15±28 | 26±29 |
| Un-EvMoSeg (Ours) | 50±21 | **45±24** | **56±15** | **53±19** | **44±21** |

Table 3. Quantitative Evaluation on EVIMO. Event-masked IoU on predicted masks and gt masks is calculated as described in 12. Our method compares favorably with EMSGC, which is the only one other than Un-EvMoSeg 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.

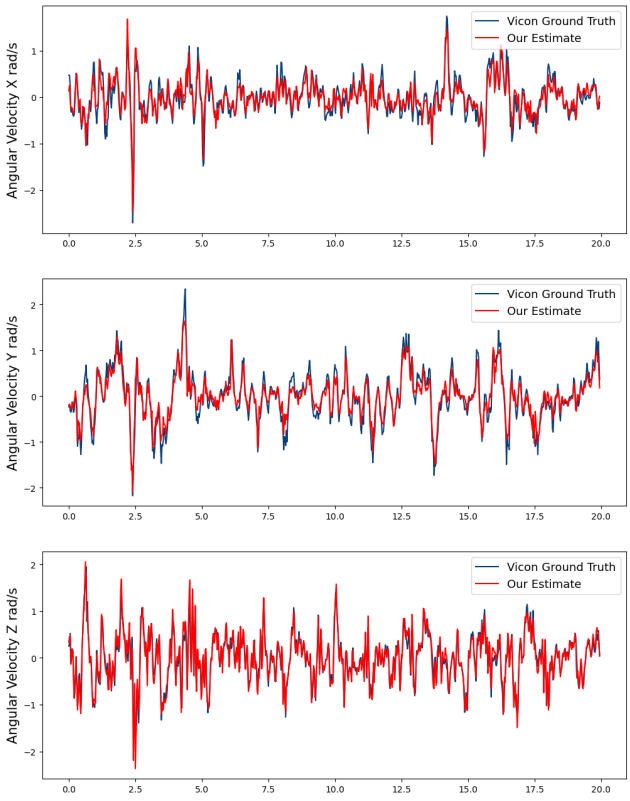



Figure 4. Estimated camera rotation from estimated optical flow. The results are shown for the whole evaluation sequence $wall\_00$. Translational velocity shown in Supplementary Material due to space limitations. Best viewed in color.

In driving datasets that have high accuracy depth sensros, such as KITTI [12], 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 [21], 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 [11, 37, 41, 46, 47] estimation. 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 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 [46] and E-RAFT [11] 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 [37]. In Figure 3, we show examples of three types of optical flow. RAFT [37] 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 [35].

**Optical Flow with Independent Motion** Flow networks trained on driving data cannot be easily used for IMO detection. To show this, we compare our optical flow results with the state-of-the-art E-RAFT models pre-trained on DSEC [10]. For this evaluation, we use 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 supervise on the high-quality optical flow computed using RAFT [37] with photometric matching and refinement. We evaluated unseen test sequences in EVIMO using RAFT output as ground truth. In our experiments, we observe that the performance gap between our Un-EvMoSeg flow network and E-RAFT is tightly correlated with how dynamic the scene is. 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 IMOs 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. [47] jointly learn emotion, depth, and flow assuming rigid scenes, which is dependent on a network to ignore independent motions. E-RAFT [11], although it does not learn ego-motion directly, has been shown to underperform in the independent motion regions [35]. 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:

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

where $\epsilon$ is an error function, $\rho$ is a robust likelihood function, 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 6. A naive optimization without outlier rejection will bias the motion estimation towards the motion of near and fast-moving objects. Based on Equation 3, the camera motion $(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$ can be solved by the linear equation:

$$\begin{bmatrix} 1/z_1 & 0 & x_1/z_1 & x_1y_1 & -(1+x_1^2) & y_1 \\ \vdots & & & & & \\ 1/z_n & 0 & x_n/z_n & x_ny_n & -(1+x_n^2) & y_n \\ 0 & -z_1^{-1} & y_1/z_1 & 1+y_1^2 & -x_1y_1 & -x_1 \\ \vdots & & & & & \\ 0 & -z_n^{-1} & y_n/z_n & n+y_n^2 & -x_ny_n & -x_n \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{x_1} \\ \vdots \\ \dot{x_n} \\ \dot{y_1} \\ \vdots \\ \dot{y_n} \end{bmatrix} \tag{8}$$

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.

### 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 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 [25].

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

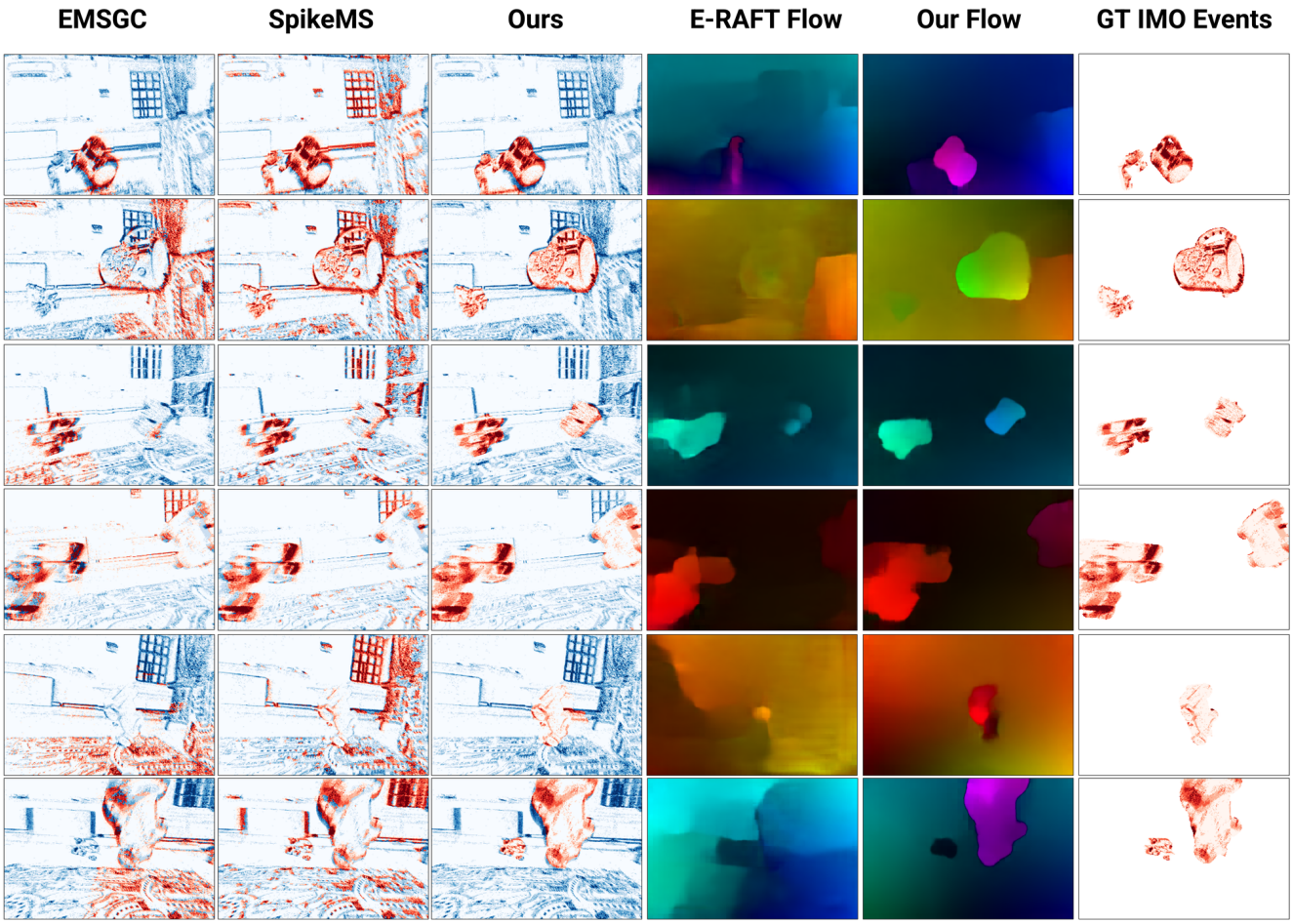



Figure 5. **Columns 1 to 3**: Segmentation Results of EMSGC, SpikeMS, Un-EvMoSeg and EVIMO-Supervised. **Columns 4 to 5**: Flow output of E-RAFT (trained on DSEC) and our fine-tuned flow network. **Column 6**: Segmented IMO event using ground truth. It can be seen that Un-EvMoSeg produces sharper and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

solutions. The objective of the search is

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

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

$P_k$ 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. 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 [11, 46], 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 event volume as described in [47]:

$$E(x, y, t) = \sum_i p_i k_b(x - x_i) k_b(y - y_i) k_b(t - t_i^*). \quad (9)$$

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-EvMoSeg. 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 [13] encoder with pre-trained weights on ImageNet [8]. Although events and images have different appearances, the kernels learned in an image-based encoder can be re-used in event-based prediction. Since objects usually occupy much less space than the rigid background, we use a Focal Loss [18] to handle the class imbalance problem. Let us denote ground truth class for a pixel as $y$: $y$ is 1 for an IMO pixel and $-1$ for a rigid background pixel, and our predicted probability for this pixel is $p$.

$$\mathcal{L}_{\text{focal}}(p_t) = -(1-p_t)^{\gamma} \log(p_t) \tag{10}$$

$$p_{\text{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise.} \end{cases} \tag{11}$$

Here, $\gamma$ is a hyperparameter that determines how much we diminish the loss for well-classified examples. We use $0.25$ for this value. 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

## 5.1. Quantitative Evaluation

In Table 3, we report the IoU our Un-EvMoSeg 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

$$\text{IoU}(O_t, P_t, E_t) = \frac{|(E_t \cap P_t) \cap (E_t \cap O_t)|}{|(E_t \cap P_t) \cup (E_t \cap O_t)|} \tag{12}$$

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. It is worth noting that the EMSGC method is very sensitive to parameters such as the class of the parametric model and the number of objects. In our evaluation, we had to devote considerable effort to tuning the parameters to get the best performance. Our model outperforms the supervised spiking method and unsupervised ESMGC (with per-sequence tuning). It can be seen that our method is comparable to supervised methods on Tables, Floor, Wall and Fast Motion. Compared to supervised methods, the main disadvantage of our approach is the lack of sharp boundaries in prediction because the network is trained with noisy labels. To better evaluate this, we computed the "detection" rate of IMOs on the floor sequence by thresholding the predicted IOU at 0.3. Our detection rate is 0.912, which indicates that we can find the object $91\%$ of the time. It is possible that the lack of sharp mask boundaries contributes to low IoU values. In terms of inference speed, we are able to run inference with neural network-based methods well over 40Hz on an RTX 3090 GPU. EMSGC performs worse in this category due to iterative optimization steps. EMSGC takes 7 to 10 seconds to process 25 ms of events on average.

## 5.2. Qualitative Evaluation

In Figure 5, we provide qualitative examples of competing methods on the Wall sequence of the evaluation set. We only show examples using methods whose source code is available. A supplementary video of our prediction on the full sequence is provided in the supplementary material. It can be seen that qualitatively, 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.

# 6. 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 IIS-RI 2212433, NSF TRIPODS 1934960, and ONR N00014-22-1-2677

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

## Supplementary Material

## 1. Summary of Items in Supplementary Files

In the compressed zip file submitted, we include the following items:

- One video of IMO segmentation on EVIMO [21]
- The supplementary PDF file (this file) with additional results

## 2. Additional Results

**Consecutive Segmentation Results** In **Figure 5** of the main manuscript, we show samples of the test sequences. These images only show how individual predictions perform. In this supplementary material, we include more consecutive predictions to show that the network prediction is consistent, although the prediction at each time is independent. In Figure 6, we show clips of continuous IMO segmentation to demonstrate temporal consistency. Similarly to our evaluation procedure, each image uses 0.025s of events. In each clip, we show six consecutive event slices in ascending order by time from left to right. We see in these figures that the boundaries of objects are sometimes misclassified as background events. There are two main reasons for this issue. First, the pseudo-masks are computed on a specific time rather than over a duration, which causes the network to predict the mask at a given time. Thus, the motion of objects during the time of the event slice can cause the network to underestimate the size of the IMO regions. In our experiments, networks trained with ground-truth labels also experience the same problem. Second, the sharp boundaries in the ground truth masks help the network learn better decision boundaries on binary classification. The baseline CNN we trained was able to keep slowly improving performance even after many epochs, whereas our method stopped improving after the first few epochs.

**Egomotion Estimation Results** In the main manuscript, we assume that the camera pose can be accurately estimated from the flow prediction. Although we do not train a network to estimate the pose, accurate optical flow and depth can be combined to estimate egomotion robustly. In Figure 9, we show the complete velocity estimate (linear and rotational) computed on unseen wall and floor sequences. Due to the high-frequency movements in EVIMO, our flow at 40Hz acts as a filter that smooths the velocities. On the other hand, the VICON ground truth is captured at 200Hz, which allows one to see the high-frequency vibrations. Overall, the robust RANSAC algorithm is able to estimate egomotion accuarately for idenfying the IMO regions.

| | Wall | Table | Floor | Box | Fast |
|---|---|---|---|---|---|
| Detection Rate | 0.853 | 0.817 | 0.912 | 0.703 | 0.694 |

Table 4. Detection rate using IoU of 0.3 on all evaluation sequences on EVIMO [21].

## 3. Additional Detection Rate Results

In Section 5.1 in the main manuscript, we explain the lower performance of our approach compared to supervised baseline methods. Sharp boundaries in ground-truth masks provide stronger discriminative signals to the network. However, in fast motion estimation, an essential task is to locate the IMOs. In Table 4, we computed the detection rate using the IoU threshold at $0.3$.

## 4. Failure Cases and Limitations

In Figure 7, 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 a limitation of our approach, which is the lack of temporal consistency in the prediction. A possible solution is to explicitly add constraints between the current IMO mask and the immediate past IMO masks during training. This could be applied during the pseudo-label generation phase as well for better ground truth.

In Figure 8, we demonstrate why adding temporal consistency can be helpful. The network lost track of the IMO at time, but it should know that an IMO is close-by 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.

## 5. Implementation Details

### 5.1. Data Preparation

As described in the main manuscript, we perform motion segmentation on events projected on $x, y$ space, allowing us to use existing image-based segmentation architectures. However, this does not imply that we discard time information from the input of the network. Instead, we use an event volume [47] to encode the spatiotemporal information in the events. The input volume has a dimension of $(N, H, W)$, where $H$ and $W$ are the spatial dimensions of the event

Figure 6. In each row, we show motion segmentation of a clip. Each clip shows temporally consistent segmentation results while each slice is predicted independently. Each row progresses temporally from left to right. Blue are background events and red are segmented IMO events. Best viewed in color.

| Sequence \ Percentile | K=0.3 | K=0.4 | K=0.5 | K=0.6 | K=0.7 | K=0.8 | K=0.9 | K=1.0 |
|---|---|---|---|---|---|---|---|---|
| Table | 55±17 | 45±23 | 36±27 | 30±28 | 26±28 | 23±27 | 20±27 | 18±26 |
| Wall | 24±33 | 18±31 | 15±28 | 12±26 | 11±25 | 9±23 | 8±22 | 7±21 |
| Floor | 18±29 | 14±26 | 11±24 | 9±22 | 8±21 | 7±20 | 6±19 | 5±18 |
| Fast | 43±27 | 33±29 | 26±29 | 22±28 | 19±27 | 16±26 | 15±25 | 13±24 |
| Box | 24±28 | 18±26 | 14±25 | 12±23 | 10±22 | 9±21 | 8±20 | 7±19 |

Table 5. Full EMSGC evaluation results on all sequences. Each column corresponds to the top $K$ performance of EMSGC.

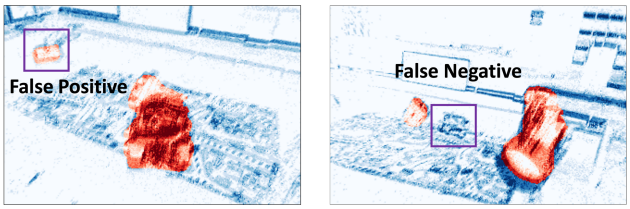



Figure 7. Failure cases of our method. On the left, the network incorrectly classifies a static square pattern on the ground as IMO. On the right, the network fails to find the apparent IMO in the scene.

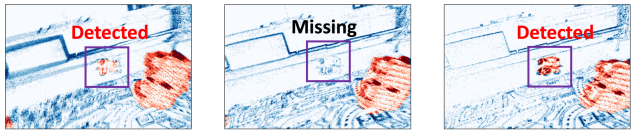



Figure 8. IMO predictions at three consecutive event slices. Our IMO detection runs on single slices of events. Occasional erroneous predictions do not have temporal consistency with the previous and next predictions.

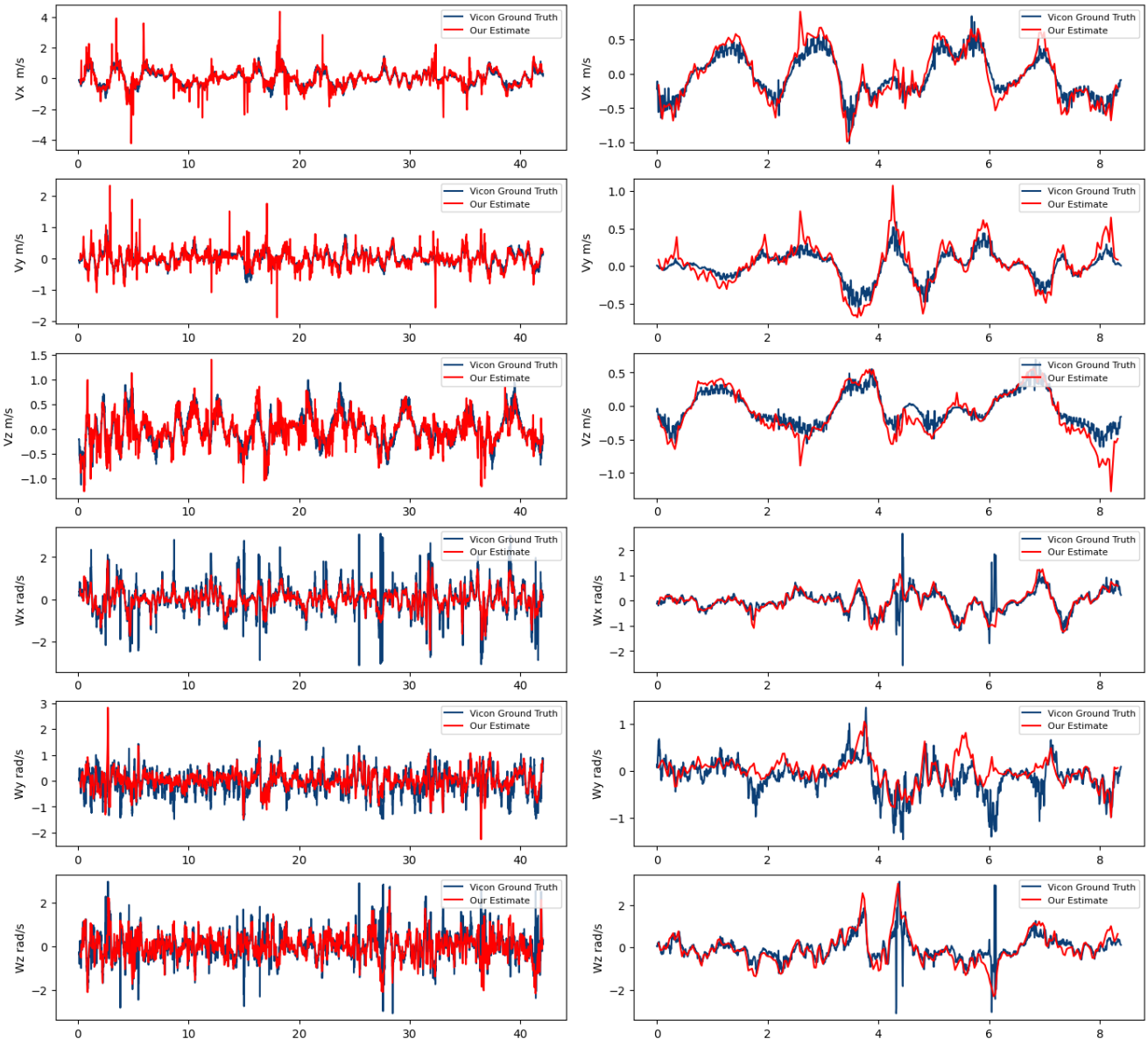
camera, and $N$ is the number of temporal bins used to discretize time. We use a relatively large number 15 for $N$ to balance between the amount of temporal information and the usage of gpu. In EV-IMO [21], an DAVIS 346 is used for data collection, the sensor resolution is $260 \times 346$. In this dataset, a rather wide lens was used, which caused distortion. Our method assumes calibrated cameras, and thus we undistort the events and input depth and crop the images to $215 \times 320$. We use the raw resolution for training and inference. In addition, we clarify the training and test split of our network. Table, Wall, Floor and Box training sequences are used during training. We performed the test on all evaluation sequences from the same four classes. We perform an evaluation on all slices where at least a single object is present, when IoU is meaningful.

We notice that multiple modalities of the provided ground truth have built-in noise. For example, the depth maps are provided with holes and the scans have discontinuities on flat surfaces. Therefore, we only use depth maps up to 3 meters of the camera during training. In our pseudo-label generation, the holes in the depth map created discontinuous masks, which we use mathematical morphology to fill these holes. However, we find the network relatively robust to these changes because the pseudo-masks are themselves noisy. We report these engineering choices to ensure that the experiments are completely reproducible. Details can be seen in the code files submitted.

## 5.2. EMSGC Comparison

EMSGC [44] is an optimization-based method. We choose to compare with this method because it similarly does not use labeled training data. In this method, the authors propose to build a spatiotemporal graph and cut the graph based on contrast loss with respect to a predetermined number of motion models (2-parameter, 4-parameter, etc.). Like many optimization methods, EMSGC suffers from high sensitivity to hyperparameters. The exact hyperparameters for each sequence are not released with the code. These parameters include various motion models for the background and foreground, the weight $\lambda$ that balances local consistency versus spatial coherence, and MDL weight that determines how much we want to regulate the number of clusters. The details are in Section VI-C of the EMSGC paper [44], which states that the parameters are obtained based on properties of the data set and empirical tuning. However, in practice, it is difficult to know these parameters in advance, which weakens the method's ability to perform real-time inference.

In our initial tests, we used their open-source code and configuration files to run prediction on all evaluation sequences. However, this approach does not produce meaningful results in most of the event slices. Then, we tried tuning the parameters on each sequence separately, but found that per-sequence tuning was not sufficient for good performance. 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 used the highest $K$ percent of all IoU to compute the mean performance and then reported the results. The performance with low K value can be seen as an approximation of the upper-bound performance of the




(a) Results on Test Sequence Wall 00

(b) Results on Test Sequence Floor 01

Figure 9. Estimated linear and angular velocity in EVIMO evaluation sequences. Red is our estimated velocities from flow and RANSAC, and blue is the ground-truth velocity captured by VICON. It can be seen that the VICON estimates are at 200Hz, which is able to capture high-frequency motion more effectively, whereas our estimate is based on flow at 40Hz.

method. In Table 5, we report the full results for selecting different $K$.

### 5.3. SpikeMS Comparison

For SpikeMS [26], we take quantitative results directly from their paper. However, there is a hyperparameter that specifies the maximum background-to-foreground ratio during evaluation. Therefore, the numbers reported in their paper can be seen as the upper bound of their performance. We used the pre-trained model released by the authors to generate the qualitative results. We notice that the network prefers to remove events in both IMO and background areas, which induces high recall, which works well in low background-to-foreground ratio scenarios. In our experiments with SpikeMS, the performance is significantly worse for general cases when the objects are smaller.

### 5.4. Supervised CNN Baseline Comparison

The original EVIMO network [21] has a few auxilliary losses to assist segmentation. GConv [22] uses a graph neural network on subsampled events where per-event labels are available. Comparing these methods does not give us a direct understanding of the effectiveness of the self-labeling mechanism. Therefore, we train a baseline network using the same architecture and ground truth labels. We report the results in Table 3 of the main article, labeled "Baseline CNN". The average performance gap between this method and Un-EvMoSeg is smaller than that between other listed methods. This simple baseline supports our hypothesis that our pseudo-labels are good approximation of the ground-truth labels, given that other factors have been controlled. We train the network using the same setting as the EVIMO network [21].

### 5.5. Optical Flow Fine-tuning

In EVIMO, only the flow of the foreground is given. We instead used RAFT [37] to compute the optical flow from low-quality DAVIS images and use these as a good reference flow. We then fine-tuned the E-RAFT [11] network for 10 epochs to allow E-RAFT to learn the IMO flow. In our experiments, we find that our flow network is able to overcome the missing IMO problem from this fine-tuning. In certain cases, it actually produces sharper flow than the RAFT flow labels. Since the ground-truth flow was missing from the general scene, we leave the full flow evaluation to future work. The fine-tuned network is forozen and is directly used as a fixed predictor in our pseudo-label generation module. We would like to emphasize that we do not claim new flow methods. Instead, we corrected the flow based on our need for accurate IMO motion estimation.

### 5.6. Network Details

In our experience with event data, pre-trained backbone usually gives the network better gradients for quicker convergence. We use a ResNet18 pre-trained on ImageNet as our encoder backbone. The event volumes are reshaped as $(15, 256, 256)$ via nearest neighbor interpolation and then fed into the network. The decoder is trained from scratch with $(256, 128, 64, 32, 16)$ channels with increasing resolution from the bottleneck. Standard skip connections between the encoder output and the decoder output are used. The final output has one channel, which is passed through the sigmoid function to get the IMO probability. We trained our network when a small validation set loss curve flattens. We do not apply special gradient clipping or decay techniques. We used a learning rate of 2e-4 with an ADAM optimizer. The batch size of our training experiments is 32. On an Nvidia RTX 3090 GPU, the training speed is about 1 iteration per second. For the supervised baseline CNN, the network is trained in the exact setting. The only difference is that the ground truth IMO masks are given and the network can train longer because the ground truth masks can force the network to learn sharp boundaries as training progresses.

## References


[1] Serge Ayer and Harpreet S Sawhney. 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*, pages 777–784. IEEE, 1995. 4

[2] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25 (2):407–417, 2013. 5

[3] Pia Bideau and Erik Learned-Miller. 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*, pages 433–449. Springer, 2016. 2

[4] Levi Burner, Anton Mitrokhin, Cornelia Fermüller, and Yiannis Aloimonos. Evimo2: An event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. *arXiv preprint arXiv:2205.03467*, 2022. 2

[5] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8001–8008, 2019. 3

[6] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. 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*, pages 7063–7072, 2019. 3

[7] Trevor Darrell and Alexander Pentland. Robust estimation of a multi-layered motion representation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 173–174. IEEE Computer Society, 1991. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 4

[10] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3): 4947–4954, 2021. 6

[11] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cam-

eras. In *2021 International Conference on 3D Vision (3DV)*, pages 197–206. IEEE, 2021. 4, 5, 6, 7

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[14] Shanon X Ju, Michael J Black, and Allan D Jepson. 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*, pages 307–314. IEEE, 1996. 2

[15] M Pawan Kumar, Philip HS Torr, and Andrew Zisserman. Learning layered motion segmentations of video. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 33–40. IEEE, 2005. 2

[16] Oliver W Layton and Brett R Fajen. A neural model of mst and mt explains perceived object motion during self-motion. *Journal of Neuroscience*, 36(31):8093–8102, 2016. 1

[17] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 473–488. Springer, 2020. 1

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 8

[19] Hugh Christopher Longuet-Higgins and Kvetoslav Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397, 1980. 3

[20] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 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*, pages 4040–4048, 2016. 4

[21] Anton Mitrokhin, Chengxi Ye, Cornelia Fermüller, Yiannis Aloimonos, and Tobi Delbruck. Ev-imo: Motion segmentation dataset and learning pipeline for event cameras. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6105–6112. IEEE, 2019. 2, 4, 5, 1, 3

[22] Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermuller, and Yiannis Aloimonos. Learning visual motion segmentation using event surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14414–14423, 2020. 2, 5

[23] Manjunath Narayana, Allen Hanson, and Erik Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1577–1584, 2013. 2

[24] Karin Nordström, Paul D Barnett, and David C O'Carroll. Insect detection of small targets moving in visual clutter. *PLoS biology*, 4(3):e54, 2006. 1, 2

[25] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. 6

[26] Chethan M Parameshwara, Simin Li, Cornelia Fermüller, Nitin J Sanket, Matthew S Evanusa, and Yiannis Aloimonos. Spikems: Deep spiking neural network for motion segmentation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3414–3420. IEEE, 2021. 4

[27] Chethan M Parameshwara, Nitin J Sanket, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. 0-mms: Zero-shot multi-motion segmentation with a monocular event camera. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9594–9600. IEEE, 2021. 2, 5

[28] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 4

[29] Sabrina Pitzalis, Patrizia Fattori, and Claudio Galletti. The functional role of the medial motion area v6. *Frontiers in behavioral neuroscience*, 6:91, 2013. 1

[30] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. 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*, pages 12240–12249, 2019. 2, 3

[31] Florian Raudies and Heiko Neumann. Modeling heading and path perception from optic flow in the case of independently moving objects. *Frontiers in behavioral neuroscience*, 7:23, 2013. 1

[32] Constance S Royden and Erin M Connors. The detection of moving objects by moving observers. *Vision research*, 50 (11):1014–1024, 2010.

[33] Simon K Rushton and Paul A Warren. Moving observers, relative retinal motion and the detection of object movement. *Current Biology*, 15(14):R542–R543, 2005. 1

[34] Nitin J Sanket, Chethan M Parameshwara, Chahat Deep Singh, Ashwin V Kuruttukulam, Cornelia Fermüller, Davide Scaramuzza, and Yiannis Aloimonos. Evdodgenet: Deep dynamic obstacle dodging with event cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10651–10657. IEEE, 2020. 2, 5

[35] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 628–645. Springer, 2022. 6

[36] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7244–7253, 2019. 2, 4

[37] Zachary Teed and Jia Deng. 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*, pages 402–419. Springer, 2020. 5, 6

[38] Qiwen Wu and Yifeng Zhang. Neural circuit mechanisms involved in animals' detection of and response to visual threats. *Neuroscience Bulletin*, pages 1–15, 2023. 1, 2

[39] Gengshan Yang and Deva Ramanan. Learning to segment rigid motions from two frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1266–1275, 2021. 3

[40] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2019. 3

[41] Chengxi Ye, Anton Mitrokhin, Cornelia Fermüller, James A Yorke, and Yiannis Aloimonos. 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)*, pages 5831–5838. IEEE, 2020. 5

[42] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018. 3

[43] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision*, 129(4):1106–1120, 2021. 3

[44] Yi Zhou, Guillermo Gallego, Xiuyuan Lu, Siqi Liu, and Shaojie Shen. Event-based motion segmentation with spatio-temporal graph cuts. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 2, 4, 5, 3

[45] Alex Zihao Zhu, Wenxin Liu, Ziyun Wang, Vijay Kumar, and Kostas Daniilidis. Robustness meets deep learning: An end-to-end hybrid pipeline for unsupervised learning of egomotion. *arXiv preprint arXiv:1812.08351*, 2018. 3

[46] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018. 5, 7

[47] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019. 5, 6, 7, 1

[48] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 36–53, 2018. 3