Supplementary Material: Un-EVIMO: Unsupervised Event-based Independent Motion Segmentation

1 Additional Results

Consecutive Segmentation Results In Figure **3** 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 1, 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 2, 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.

2 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 1, we computed the detection rate using the IoU threshold at 0.3.

Fig. 1: 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.

Table 1: Detection rate using IoU of 0.3 on all evaluation sequences on EVIMO [3].

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

Table 2: Full EMSGC evaluation results on all sequences. Each column corresponds to the top K performance of EMSGC.

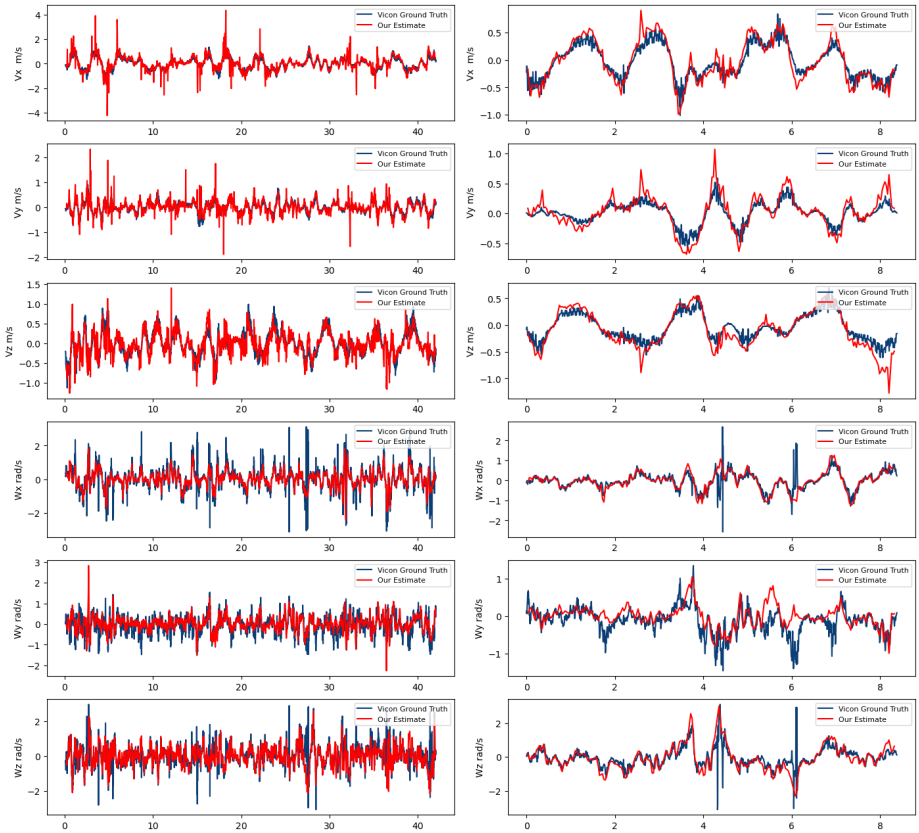
Sequence \ Percentile	K=0.3	K=0.4	K=0.5	K=0.6	K=0.7	K=0.8	K=0.9	K=1.0
Table	55±17	45±23	36±27	30±28	26±28	23±27	20±27	18±26
Wall	24±33	18±31	15±28	12±26	11±25	9±23	8±22	7±21
Floor	18±29	14±26	11±24	9±22	8±21	7±20	6±19	5±18
Fast	43±27	33±29	26±29	22±28	19±27	16±26	15±25	13±24
Box	24±28	18±26	14±25	12±23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

3 Implementation Details

3.1 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 [7] 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 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 [3], an DAVIS 346 is used for data collection, the sensor resolution is 260×346. 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 215×320. 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.



(a) Results on Test Sequence Wall 00 **(b)** Results on Test Sequence Floor 01

Fig. 2: 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.

3.2 EMSGC Comparison

EMSGC [6] 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 λ 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 [6], 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 method. In Table 2, we report the full results for selecting different K.

3.3 SpikeMS Comparison

For SpikeMS [4], 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.

3.4 Supervised CNN Baseline Comparison

The original EVIMO network [3] has a few auxilliary losses to assist segmentation. GConv [2] 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-EVIMO 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 [3].

3.5 Optical Flow Fine-tuning

In EVIMO, only the flow of the foreground is given. We instead used RAFT [5] 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 [1] 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.

3.6 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. 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
2. 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) 5
3. 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) 3, 5, 6

4. 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) 5
5. 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
6. 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) 5
7. 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) 3