Event-based Semantic-aided Motion Segmentation

Chenao Jiang, Julien Moreau, Franck Davoine

Event-based Semantic-aided Motion Segmentation

Chenao Jiang[1][a], Julien Moreau[1][b] and Franck Davoine[1][c]
[1]*Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60319 - 60203 Compiègne Cedex, France.*
{*chenao.jiang, julien.moreau, franck.davoine*}*@hds.utc.fr*

Keywords: Event cameras, Unconventional vision, Semantic and Motion segmentation

Abstract: Event cameras are emerging visual sensors inspired by biological systems. They capture intensity changes asynchronously with a temporal precision of up to μs, in contrast to traditional frame imaging techniques running at a fixed frequency of tens of Hz. However, effectively utilizing the data generated by these sensors requires the development of new algorithms and processing.
In light of event cameras' significant advantages in capturing high-speed motion, researchers have turned their attention to event-based motion segmentation. Building upon (Mitrokhin et al., 2019) framework, we propose leveraging semantic segmentation enable the end-to-end network not only to segment moving objects from background motion, but also to achieve semantic segmentation of distinct moving objects. Remarkably, these capabilities are achieved while maintaining the network's low parameter count of 2.5M. To validate the effectiveness of our approach, we conduct experiments using the EVIMO dataset and the new and more challenging EVIMO2 dataset (Burner et al., 2022). The results demonstrate improvements attained by our method, showcasing its potential in event-based multi-objects motion segmentation.

1 INTRODUCTION

Motion segmentation plays a vital role in enabling autonomous robots to navigate dynamic scenes. However, this has always been a challenging problem due to the presence of dual motion originating from both the camera and moving objects.

Traditional imaging cameras often struggle in dynamic scenarios with moving objects due to motion blur and low-light conditions. Inspired by the spiking nature of biological visual pathways, neuromorphic engineers have developed a sensor called event camera, or Dynamic Vision Sensor (DVS) (Lichtsteiner et al., 2008). Unlike conventional image frames, the DVS captures asynchronous temporal changes in the scene as a stream of events. When a change in log light intensity is detected in a pixel, the camera immediately returns an event, $e = \{x, y, t, p\}$, consisting of the position of the pixel (x, y), timestamp of the change t, accurate to microseconds, and the polarity of the change p, corresponding to whether the pixel became brighter or darker.

Event cameras provide benefits in terms of temporal resolution, low latency, and low-bandwidth signals. However, due to the unconventional output and underlying principles of operation, algorithms designed for traditional cameras cannot be directly applicable. To fully harness the potential of event cameras, novel algorithms are required. A recent survey (Gallego et al., 2022) offers a overview of event cameras, algorithms, and their applications.

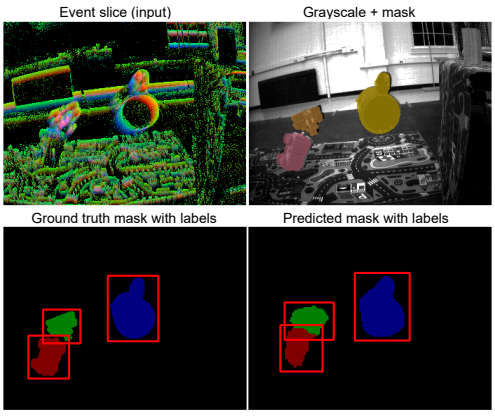

Figure 1: Motion and semantic segmentation with a monocular event camera on an EVIMO dataset sequence. Event slice will be explained in Sec. 3.1. Grayscale images are only provided for visualization, and bounding boxes on the images are only used for computing the evaluation metrics. Best viewed in color.

[a] https://orcid.org/0009-0005-6283-4064
[b] https://orcid.org/0000-0001-5008-9232
[c] https://orcid.org/0000-0002-8587-6997

Our work focuses on the motion segmentation estimation by leveraging the exceptional temporal resolution and high dynamic range of event cameras. We deal with dynamic scenes observed by moving event cameras, and containing Independently Moving Objects (IMOs). This scenario poses greater difficulties than with static cameras, as events are influenced not only by moving objects but also by the background.

To address this problem, we build upon the neural network framework proposed by (Mitrokhin et al., 2019), which estimates 3D motion field using event data. Expanding on this architecture, we introduce the task of supervised semantic segmentation to the network. By doing so, we empower the network to effectively segment IMOs in dynamic scenes captured by event cameras. Fig. 1 displays visual results, providing a sample of the output from our work.

Our contributions can be summarized as follows:

- A neural network inferring camera and object motion, and semantic segmentation, from event data.
- A comprehensive evaluation, qualitative and quantitative, conducted on available datasets, showing better performance compared to competing baseline methods.
- Introducing more complete metrics to assess performance in event-based IMOs segmentation and detection.

The rest of the paper is organized as follows: Sec. 2 discusses the related works on event-based motion segmentation and semantic segmentation problems. Sec. 3 provides a detailed presentation of our network, followed by a comprehensive evaluation on different datasets in Sec. 4. Finally, the study is summarized in Sec. 5.

2 RELATED WORK

2.1 Event-Based Motion Segmentation

Motion segmentation for a static event-based camera is relatively straightforward since events in this scenario are exclusively caused by moving objects (assuming no changes in illumination) (Litzenberger et al., 2006). For example, (Piatkowska et al., 2012) apply Gaussian mixture models to detect, describe and track objects in the case of static event cameras.

More challenges arise when dealing with a moving camera, as events are triggered across the entire image plane, originating from both moving objects and the apparent motion of the static scene induced by the camera's ego motion. In early works on event-based motion segmentation, it was necessary to have prior knowledge about the shape of IMOs: (Glover and Bartolozzi, 2016) detect and track circular objects (such as a ball) in the presence of clutter caused by camera ego-motion by extending the Hough-based circle detection algorithm using optical flow information from the spatio-temporal event space. Alternatively, prior knowledge about the correlation between tracked geometric primitives and the motion of the event camera was also used, e.g. (Vasco et al., 2017) detect and track corners in the event stream and learn the statistics of their motion as a function of the robot's joint velocities when no IMOs are present.

In more recent works, such prior knowledge is no longer necessary. There are some model-based methods like (Stoffregen and Kleeman, 2018), that works by collecting events up to a threshold and applying focus maximisation with a 2-Degree of Freedom (DoF) optic flow motion model. The events belonging to the dominant motion (e.g., background) were then removed to analyze the remaining events in a greedy manner with the same process. (Mitrokhin et al., 2018) use a similar scheme, whereby focus optimisation with a 4-DoF motion model is applied to a set of events to find the dominant motion, which is assumed to be the camera ego-motion. However it fails to achieve accurate segmentation in densely textured environments or in the presence of overlapping moving objects. (Stoffregen et al., 2019) improve on these results, which uses focus optimisation on multiple motion models together with a probabilistic model. The motion parameters and the event probabilities are then updated in a combined optimisation in an Expectation Maximisation (EM) approach. (Parameshwara et al., 2021) present a model based approach similar to (Stoffregen et al., 2019), apply a global motion-compensation, resulting in a sharp background and blurry object boundaries. They then apply motion tracking to the residual events and use K-means clustering to group the resulting track lets (K is set to a large value), the clusters are then merged using a contrast and distance function. Recently, (Zhou et al., 2023) propose to cast the motion segmentation problem as an energy minimization one involving the fitting of multiple motion models. They jointly solve two sub-problems, namely event cluster assignment (labeling) and motion model fitting, in an iterative manner by exploiting the structure of the input event data in the form of a spatio-temporal graph.

More related to our approach are the machine learning-based methods. (Mitrokhin et al., 2020) use a Graph Convolutional Neural Network (GCNN) architecture in which the nodes are the events. (Mitrokhin et al., 2019) propose a pipeline that incorporates a depth prediction and a pose prediction net-

works to estimate depth, per-pixel pose, and motion segmentation mask in supervised mode. The outputs are used to generate the optical flow, associated with a two-stage loss to evaluate the warping quality.

2.2 Event-Based Semantic Segmentation

Semantic segmentation is a visual recognition task that involves assigning a semantic label to each pixel in an image. Deep learning addresses this problem with state-of-the-art solutions predominantly relying on encoder-decoder CNN architectures using RGB images, such as (He et al., 2016), (Chollet, 2017), (Chen et al., 2018).

The pioneering work in event-based semantic segmentation was (Alonso and Murillo, 2019), with an Xception-type network (Chollet, 2017).

Novel sensors like event cameras often face a common challenge of limited labeled datasets for semantic segmentation. To address this issue, several approaches try to leverage labeled conventional images to train networks for event cameras. This transfer from a labeled source domain (images) to an unlabeled target domain (events) is generally defined as Unsupervised Domain Adaption (UDA). In this context, (Sun et al., 2022) explores the utilization of UDA for event-based semantic segmentation.

Current motion segmentation methods can detect and segment moving objects but lack semantic information. We believe that semantic information is valuable for motion segmentation, as it not only includes class labels but also relates to the dynamics and expected motion of objects. Moreover, both motion and semantic segmentation tasks can be achieved through encoder-decoder network. Therefore, in this work, we propose a multi-task network that estimate jointly motion and semantic segmentation, and demonstrating that these two tasks can mutually benefit each other.

3 PROPOSED ARCHITECTURE

3.1 Event Representation

The raw data of the Dynamic Vision Sensor (DVS) consists of a continuous stream of events, the representation of the event stream is the form of a sparse three-dimensional point cloud. For this unconventional data, various representation methods currently exist: event frame or 2D histogram, time surface, voxel grid, 3D point set, etc (Gallego et al., 2022). To enhance the efficiency of our convolutional neural network and maximize the utilization of the event

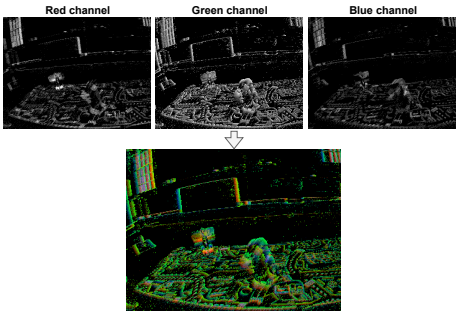

Figure 2: Example of the event slice represented in RGB mapping with a scene from EVIMO dataset. The R and B color channels represent the positive and negative event counts and the G color channel represents the aggregation of timestamps within the δt. Best viewed in color.

stream, we aim to represent it in a 2D form and partition it into continuous time slices of size δt (25ms in our network). The event information within each time slice is projected onto a generated frame referred to as *event slices*, akin to the approach in (Mitrokhin et al., 2018) and (Mitrokhin et al., 2019). An example of such event slice can be seen in Fig. 2. This frame-like representation encompasses three channel mappings: one is the aggregation of timestamps within the δt time slice, while the other two correspond to the count of positive and negative events.

3.2 Overview of the Architecture

Our pipeline, Fig. 3, inherits from (Mitrokhin et al., 2019) principle. It comprises a depth prediction network and a semantic-aided motion segmentation network, both designed as low-parameter encoder-decoder networks (Ronneberger et al., 2015). The depth prediction network is similar to that in (Mitrokhin et al., 2019) and is summarized in Sec. 3.3. Proposed semantic and motion segmentation network shares a common encoder and split into two decoding branches, for simultaneous semantic segmentation and motion segmentation. This way we introduce a lightweight semantic-aided motion segmentation network, which is detailed in Sec. 3.4.

Both networks are based on CascadeLayer and InvertedCascadeLayer, introduced in (Ye et al., 2018) and shown in Fig. 4. These two blocks aims to fuse multi-level features through concatenation operations. In our network, the output dimension of the CascadeLayer is half of the input, while the output dimension of the InvertedCascadeLayer is twice that of the input. Channel numbers of the features are controlled by the *growth rate* hyper-parameter. This configuration allows our network to correspond in the encoding and decoding layers and introduce skip connections to integrate features from different levels.

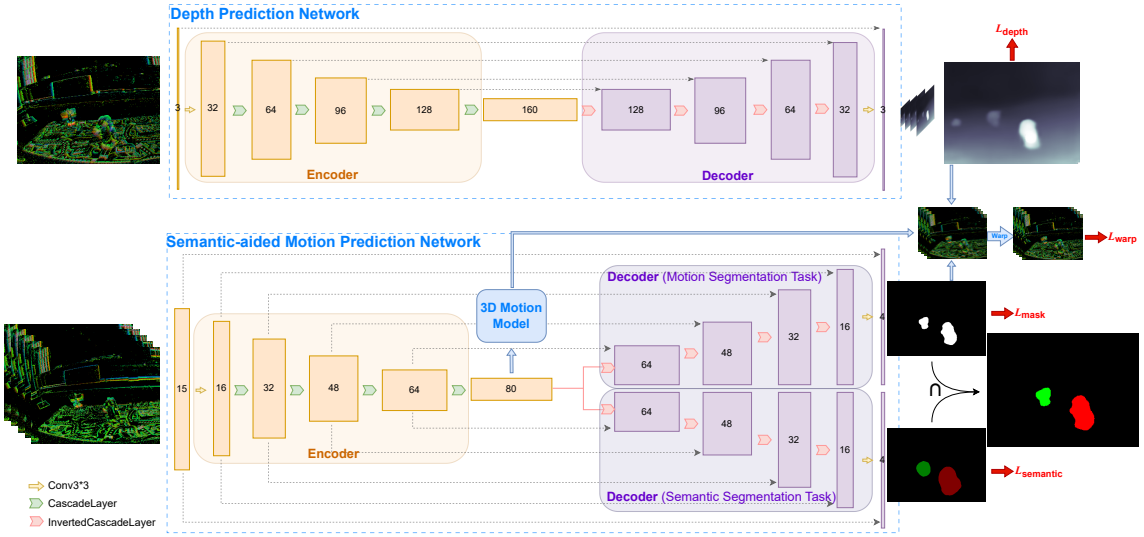

Figure 3: Network architecture for our depth and semantic-aided motion segmentation model. Top: the Depth Prediction Network uses an encoder-decoder architecture and is trained in supervised mode to estimate multi-scale depths. Bottom: the Semantic-aided Motion Segmentation Network and 3D Motion Model share the encoder and then branch out to predict 3D motion vector, multi-scale motion masks, and semantic segmentation. During training, the outputs (depth, 3D motion vector, motion mask) are combined to generate the optical flow and then to inversely warp the inputs and back-propagate the error. The final result is a motion mask with semantic labels. The number of channels before and after each CascadeLayer corresponds to C and C+gr in Fig. 4, and the InvertedCascadeLayer corresponds to C and C-gr.

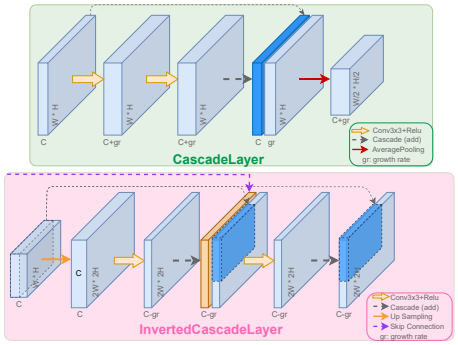

Figure 4: Design of the CascadeLayer and InvertedCascadeLayer.

3.3 Depth Prediction Network

The depth prediction network estimates scaled inverse depth from a single event slice (representation outlined in Sec. 3.1). As in (Mitrokhin et al., 2019), the network is supervised by ground-truth depth image, and its output is used as depth information in the subsequent process of calculating optical flow.

During the decoding phase, different levels of features are predicted, a bi-linear interpolation is applied to up-sample and refine the depth map, and introduce residuals to integrate different levels of features into the backbone prediction. For each level of predicted depth *pred*, the corresponding ground truth *gt* is generated through average pooling for supervision, with penalties applied to their deviations as follows, while also applying a smoothness constraint to the second-order gradient of depth prediction:

$$\mathcal{L}_{depth} = \max\left(\frac{gt}{pred}, \frac{pred}{gt}\right) + \frac{|pred - gt|}{gt} + ||\Delta pred||_1 \quad (1)$$

3.4 Semantic-aided Motion Segmentation Network

Semantic-aided Motion Segmentation Network shares an encoder and has a motion module and two decoders for motion segmentation and semantic segmentation tasks. The outputs' contributions to loss functions for training are shown in Fig. 3.

3.4.1 Motion Segmentation task

The Semantic-aided Motion Segmentation Network takes 5 consecutive event slices as input, so that it can explain the motion from the original input event data. Similar to (Mitrokhin et al., 2019), after an encoder composed of CascadeLayer (see Fig. 4), and under the assumption that both camera and objects are rigid motions, Semantic-aided Motion Segmentation Network uses the 3D motion model to estimate pixel-wise 3D motion (including camera ego-motion and object motion) from continuous event slices.

In the decoding phase, multi-scale prediction and residual fusion are performed under the supervision of ground truth, and finally the pixel-wise motion mask

weight is predicted for the background and moving objects. This weight is weighted to the camera ego-motion and object 3D motion to obtain the final pixel-wise 3D motion vector denoted as $\boldsymbol{p}$ which is the candidate velocities of objects relative to the camera. As show in the middle of Fig. 3.

Then, utilizing the Image Jacobian (Hutchinson et al., 1996) which relates image-plane velocity of a point to the relative velocity of the point with respect to the camera, we leverage the predicted 3D motion vector to predict the motion field, thereby effectively compensating for motion:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{x}{Z} & 1+y^2 & -xy & -x \end{bmatrix}}_{J_{img}} \underbrace{\begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}}_{\boldsymbol{p}} \tag{2}$$

where $\boldsymbol{p} = (\boldsymbol{v}, \boldsymbol{\omega})^T$ is 3D motion vector with a translational velocity $\boldsymbol{v} = (v_x, v_y, v_z)^T$ and a rotational velocity $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$, J_{img} denotes the 2x6 Image Jacobian matrix. $\boldsymbol{x} = (x, y)^T$ is the point image coordinates and its velocity $(\dot{x}, \dot{y})$. Thus, for each pixel, there is a linear relation between the optical flow and the 3D motion vector.

In the training process, we calculate the optical flow (Eq. 2) in order to inversely warp events to compensate for the motion, as show in the right of Fig. 3. This optical flow, along with depth (output of the Depth Prediction Network), is then utilized to calculate the warp loss (Sec. 3.4.3), which is employed for training optimization through back-propagation.

3.4.2 Semantic Segmentation task

In the previous section, we compute motion compensation to achieve the background and moving object segmentation (background and foreground). However, the segmentation between moving objects is also crucial in the field of autonomous robots, and this can be aided by semantic segmentation.

Currently, for traditional cameras, most of state-of-the-art on semantic segmentation solutions based on deep learning are based on different variants of the encoder-decoder CNN architecture (Ronneberger et al., 2015) (Chen et al., 2018). Similarly, for event-based cameras, they also adhere to an encoder-decoder framework (Alonso and Murillo, 2019). To minimize network parameters and complexity, we propose a method that shares the encoder with the motion mask prediction task, while introducing an auxiliary decoder for semantic segmentation task. Within this network, this decoder also is built upon the lightweight and trainable InvertedCascadeLayer (refer to Fig. 4) for multi-scale per-pixel classification. Additionally, our architecture employs skip connections (Ronneberger et al., 2015) to integrate shallow-level features, aiding in optimizing the deep structure and mitigating gradient vanishing issues. As shown below in Fig. 3.

Semantic-aided Motion Segmentation Network outputs are as follows. The output of the motion segmentation task is the pixel-wise motion mask weight. To get the binary motion mask, we apply a threshold to these weights. In the semantic segmentation task, we generate a mask containing the labels for each pixel. However, to reduce noise effects, we apply a median filter to the semantic mask before combining the outputs of these two tasks. Values of these filters are given in Sec. 4.2. The final network output is a semantically labeled motion mask, which is the intersection of the outputs from the two tasks.

3.4.3 Loss Functions

Here we describe the loss function used in Semantic-aided Motion Segmentation Network. Similarly to the Depth Prediction Network (Sec. 3.3), the output of our network is multi-scale. The loss functions described in this section are also computed at different scales and use residual structures to integrate features at different levels into backbone predictions. They are weighted by the number of pixels.

Warp Loss: From estimated optical flow, we perform an inverse warp of 4 adjacent event slices onto the central event slice. The warp loss is defined by the absolute difference in event counts after warping:

$$\mathcal{L}_{warp} = \sum_{-2 \le n \le 2, n \neq 0} |I_n^{warped} - I_0| \tag{3}$$

where I_n^{warped} and I_0 denote the warped adjacent event slices and the central event slice respectively.

Mask Loss: We apply a binary cross-entropy loss to constrain that our model applies the ego motion in the background region, while also applying a smoothing loss on their first-order gradients:

$$\mathcal{L}_{mask} = -\sum \log(mask_{bg}^i) \quad + \quad ||\nabla mask||_1 \tag{4}$$

where $mask_{bg}^i$ are the motion mask weights for i^{th} pixel and this pixel is the background.

Semantic Loss: With supervision form ground-truth labels at every scale, we employ the common softmax cross-entropy loss function to compute the cumulative pixel-wise loss:

$$\mathcal{L}_{semantic} = -\frac{1}{N} \sum_{j=1}^{N} \sum_{c=1}^{C} w_c y_{c,j} \log(\hat{y}_{c,j}) \tag{5}$$

where N is the number of labeled pixels and C is the number of classes. $y_{c,j}$ is a binary indicator of pixel j

belonging to class c (ground truth). $\hat{y}_{c,j}$ is the network predicted probability of pixel j belonging to class c.

In order to solve the problem of uneven number of category samples (generally speaking, the number of pixels in the background class far exceeds those in a specific moving object class), we add weights to each category in the cross-entropy loss:

$$w_c = \frac{\max(n_1, n_2, \cdots, n_C)}{n_c} \tag{6}$$

where w_c is the weight to c^{th} class, and n_c is the number pixels belonging to c^{th} class. This way, larger weights are assigned to classes with fewer pixel counts.

Total Loss: Finally, we aggregate the above loss functions through weighted summation to obtain our total loss (weights are defined by λ_i):

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{depth} + \lambda_2 \mathcal{L}_{warp} + \lambda_3 \mathcal{L}_{mask} + \lambda_4 \mathcal{L}_{semantic} \tag{7}$$

The Semantic-aided Motion Segmentation Network predicts both motion masks and semantic labels, which we take the intersection to obtain the final mask of different motion objects.

4 EVALUATION

4.1 Setup and training

We conducted experiments using the two networks explained in Sec. 3, along with the applicable datasets introduced in Sec. 4.2. We trained our networks from scratch using the following configuration: Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$; initial learning rate of 0.01 with a cosine annealing learning rate schedule. We trained for 50 epochs with a batch size of 32, applying data augmentation steps during training, including random zoom scales and crops as well as horizontal flips. We set *growth rate* (see Sec. 3.2) to 32 for the Depth Prediction Network and 16 for the Semantic-aided Motion Segmentation Network, and use the batch normalization. For the weights in Eq. 7, we set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$. Our baseline architecture consists of approximately 2.5 million parameters and each epoch takes about 25 minutes to train on an Nvidia Tesla V100-SXM2-32GB GPU. We save the model parameters of the epoch with the lowest validation loss as the best model for use.

4.2 Datasets

Literature in event-based motion detection provides following datasets: EED (Mitrokhin et al., 2018), MOD (Sanket et al., 2019) , MOD++ (Parameshwara et al., 2021), EVIMO (Mitrokhin et al., 2019), EVIMO2 (Burner et al., 2022). Since the training of our network is supervised, it needs depth ground truth frames and motion truth masks with semantic labels. Tab. 1 lists the characteristics of event-based motion detection datasets and whether they are applicable.

Therefore, during the experimental phase, we conducted quantitative evaluations on the EVIMO and EVIMO2 datasets in Sec. 4.4 and 4.5, applying motion weight thresholds of 0.7 and 0.8, and median filter with sizes of 7×7 and 15×15, respectively.

4.3 Evaluation Metrics

For the quantitative evaluation of our network, we employ two commonly used metrics in research on event-based motion segmentation: *Detection Rate* and *Intersection over Union (IoU)*. In addition, to conduct a comprehensive evaluation of IMOs segmentation and detection from events, we adopt the use of the standards *mIoU*, *Precision*, *Recall*, and *F1-score* to this application.

All these metrics, except IoU and mIoU, need bounding boxes representation. We compute them to fit the boundaries of the motion mask of each class.

4.3.1 Detection rate

We assess the performance of moving object detection using the detection rate. Detection rate was introduced in (Mitrokhin et al., 2018) and used ever since. Motion detection is considered successful when the estimated bounding box satisfies the following two conditions: 1) The overlapping area with the ground truth bounding box is greater than 50%; 2) The area intersected with the ground truth bounding box is greater than the area intersected with the outside world. We can express the metric as:

$$Success \text{ if } \mathcal{B}_p \cap \mathcal{B}_{gt} > 0.5 \text{ and } (\mathcal{B}_p \cap \mathcal{B}_{gt}) > (\mathcal{B}_p \cap \overline{\mathcal{B}_{gt}})$$

where $\mathcal{B}_p$ refers to the predicted bounding box, $\mathcal{B}_{gt}$ refers to the ground truth bounding box, and $\bar{\cdot}$ denotes the complement of a set. The detection rate is only used to evaluate the detection of moving objects, without checking their categories.

4.3.2 Intersection over Union (IoU) and mIoU

IoU: IoU is a standard metrics used to evaluate the performance of binary segmentation. It is used in this work and in related literature to show the accuracy of the segmented motion mask. IoU is expressed as:

$$IoU = \frac{\mathcal{S}_p \cap \mathcal{S}_{gt}}{\mathcal{S}_p \cup \mathcal{S}_{gt}}$$

where $\mathcal{S}_p$ refers to the predicted motion mask and $\mathcal{S}_{gt}$ the ground truth mask.

Table 1: Summary of characteristics of the event datasets for moving objects detection.

Dataset	EED (Mitrokhin et al., 2018)	MOD (Sanket et al., 2019)	MOD++ (Parameshwara et al., 2021)	EVIMO (Mitrokhin et al., 2019)	EVIMO2 (Burner et al., 2022)
Data-type	Real	Simulated	Simulated	Real	Real
Camera type	DAVIS 240B	Sim. DAVIS 346C ESIM (Rebecq et al., 2018)	Sim. DAVIS 346C ESIM (Rebecq et al., 2018)	DAVIS 346C	Samsung DVS Gen3
Resolution	240 x 180	346 x 240	346 x 240	346 x 240	640 x 480
Data	Events Grayscale Images IMO Bounding Boxes	Events RGB Images 6-DoF Camera + IMO Pose IMO Masks + Bounding Boxes	Events RGB Images 6-DoF Camera + IMO Pose IMO Masks + Bounding Boxes Optical Flow Depth	Events Grayscale Images 6-DoF Camera + IMO Pose IMO Masks Depth	Events RGB Images 6-DoF Camera + IMO Pose IMO Masks Depth
Scene Type	Indoor	Sim. Indoor + Outdoor	Sim. Indoor + Outdoor	Indoor	Indoor
Suitability for Our Network	No (Lack of Depth and IMO Mask)	No (Lack of Depth)	No (Without training sequences)	Yes	Yes

Similar to the detection rate, IoU also doesn't take into account the classes of moving objects.

mIoU: Since the above evaluation metrics are only for binary motion segmentation, we introduce mean Intersection over Union (mIoU) to evaluate joint semantic and motion segmentation results, which involves calculating IoU for each class of moving objects and then taking the average.

4.3.3 Precision, Recall and F1-score

Precision and Recall are the main metrics in object detection problem. We aim to employ them for a more comprehensive evaluation of IMOs detection. Therefore, slightly different from the detection rate which is blind to object classes, we extract bounding boxes and calculate precision and recall for each class of the moving objects, and take the average to assess the performance of our network. We consider a predicted bounding box to be a True Positive (TP) if its IoU with the ground truth bounding box is greater than 0.5. In following formulas, C is the number of classes, TP_i is the number of TP bounding box to i^{th} class. $Npredicted_i$ and Ngt_i refer to the total number of predicted and ground truth bounding boxes to i^{th} class.

Precision: Precision is the ratio between the number of correctly predicted bounding boxes to the total number of predicted bounding boxes:

$$Precision = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{Npredicted_i}$$

Recall: Recall is the ratio between the number of correctly predicted bounding boxes to the total number of ground truth bounding boxes:

$$Recall = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{Ngt_i}$$

F1-score: F1-score is the harmonic mean of precision and recall, while considering both wrong prediction and missed detection:

$$F1-score = \frac{2*(Precision*Recall)}{(Precision+Recall)}$$

All these metrics are inspired by classical detection metrics, using bounding boxes.

We propose to adopt these metrics for a more thorough evaluation of the moving objects detection from events. Precision is similar to the detection rate, but it takes into account IMO separation and association with the ground truth objects. For example, when a wrong class is predicted to a moving object, it is a positive sample for the detection rate, but a negative sample for the precision. Hence, precision criteria is more severe than the detection rate. Recall is complementary to precision, and together, they inform whether evaluated algorithm is conservative or loose in its predictions. F1-score gives a criteria to compare the methods as a trade-off between Precision and Recall.

4.4 Evaluation on the EVIMO Dataset

EVIMO (Mitrokhin et al., 2019) is a challenging dataset for event-based IMO segmentation. Sequences are recorded in an indoor real-world environment with five different backgrounds (*box, floor, table, tabletop, wall*). They include objects moving at high speed with random trajectories. It was collected using a fast-moving handheld DAVIS 346 camera.

We trained on the EVIMO dataset training set, and evaluated on sequences *box*, *table*, *wall* and *fast-motion* using detection rate (Tab. 2) and IoU (Tab. 3). In relative terms, *wall* and *fast-motion* sequences are more challenging due to their high number of objects and fast motion. As a result, their outcomes are not as favorable as those of other sequences. To compare

with state-of-the-art methods, EVIMO (Mitrokhin et al., 2019), 0-MMS (Parameshwara et al., 2021) and EMSGC (Zhou et al., 2023), our method has a slight improvement in detection rate and IoU score. Up to now, 0-MMS had the best scores. It uses an iterative model fitting and merging approach. However, it requires manual parameter selection for each sequence and cannot automatically select parameter.

For the IoU evaluation, our network achieved a 1% improvement over the original method we are based on, EVIMO, further demonstrating the value of semantic information.

Table 2: Comparison of proposed method on EVIMO using detection rate with available state-of-the-art data.

Method	Detection rate for sequence (%)				
	box	table	wall	fast motion	avg. of box+wall
0-MMS	-	-	-	-	81.06
Ours*	91.28	83.79	74.00	65.07	**82.64**

* Learning-based Method

Table 3: Comparison of proposed method on EVIMO using IoU with available state-of-the-art data.

Method	IoU for sequence (%)					
	box	table	wall	fast motion	avg. of box+wall	avg. of table+wall +fast motion
EMSGC	-	-	-	-	76.81	-
0-MMS	-	-	-	-	80.37	-
EVIMO*	-	83	75	73	-	77.00
Ours*	84.67	83.18	77.65	73.67	**81.16**	**78.17**

* Learning-based method.

Table 4: Evaluation of proposed method on EVIMO using mIoU for semantic segmentation, as well as Precision, Recall and F1-score for separated moving objects detection.

Metrics	Sequence				
	box	table	wall	fast motion	avg. of box+wall
mIoU (%)	77.97	71.50	58.98	57.00	68.47
Precision	0.87	0.80	0.55	0.36	0.71
Recall	0.83	0.81	0.68	0.58	0.76
F1-score	0.85	0.80	0.61	0.45	0.73

Meanwhile, we conducted quantitative evaluations of semantic segmentation using mIoU, Precision, Recall, and F1-score in Tab. 4. Average on *box* plus *wall* sequences is given for comparison with the metrics given in Tab. 2 and 3. As expected, reaching high scores with proposed metrics is harder than with the previous metrics used in event-based motion literature, giving more reliable measurements to compare best-performing methods. However, there is currently a lack of state-of-the-art methods that combine motion segmentation with semantic segmentation for comparison against our approach.

Furthermore, qualitative evaluation is also provided in Fig. 5. Our method is robust to scenes with multiple fast-moving IMOs as well as scenes with fast camera movements. In addition to showcasing successful cases, we have included instances of failure to better help future research work in Fig. 6.

4.5 Evaluation on the EVIMO2 Dataset

EVIMO2 (Burner et al., 2022) improves on the EVIMO dataset by providing data from cameras with higher resolution, in more complex scenarios, with more rotations and more objects.

In order to be able to conduct experiments on EVIMO2, specific processing have been mandatory:

IMO masks labelling: EVIMO2 provides per-pixel ground truth depth, semantic segmentation, as well as camera and object poses. EVIMO2 involves multiple independently moving objects, but the IMO masks are not provided separately in the ground truth mask. We calculate the object velocities in world coordinates using camera and object poses, then generate the IMOs ground truth masks from moving objects. For our networks supervision, generated IMOs ground truth masks are used with the motion segmentation output, while the provided ground truth labels are used with the semantic segmentation output (they are simplified, details in next paragraph). Since the objects classes in the EVIMO2 dataset are different from EVIMO, we cannot train them together.

Semantic classes restructuring: In EVIMO2, there are 26 instance labels (including the background), while there are only 4 labels in EVIMO. Since EVIMO2 dataset contain only 21 training sequences, having 26 labels is too many.

Therefore, we performed semantic classification on the 26 instance labels, and finally divided them into 7 classes (including background) for training, as shown in Tab. 5.

Figure 7 provides examples after reorganizing the labels, with used color scheme. The “others” class is composed of objects with different shapes, so the network cannot effectively treat them as a same semantic object. In fact, each object in “others” class has a small number of samples in the dataset and cannot be associated to a class in a way to imbalance well the restructured classes.

Despite us adding weights to address the sample imbalance issue in the semantic loss, training them as individual semantic labels remain challenging. For classes other than “others”, our network can learn relatively well, as shown in Fig. 8. However, for the “others” class, our network struggles to predict their labels accurately, as shown in first row in Fig. 9.

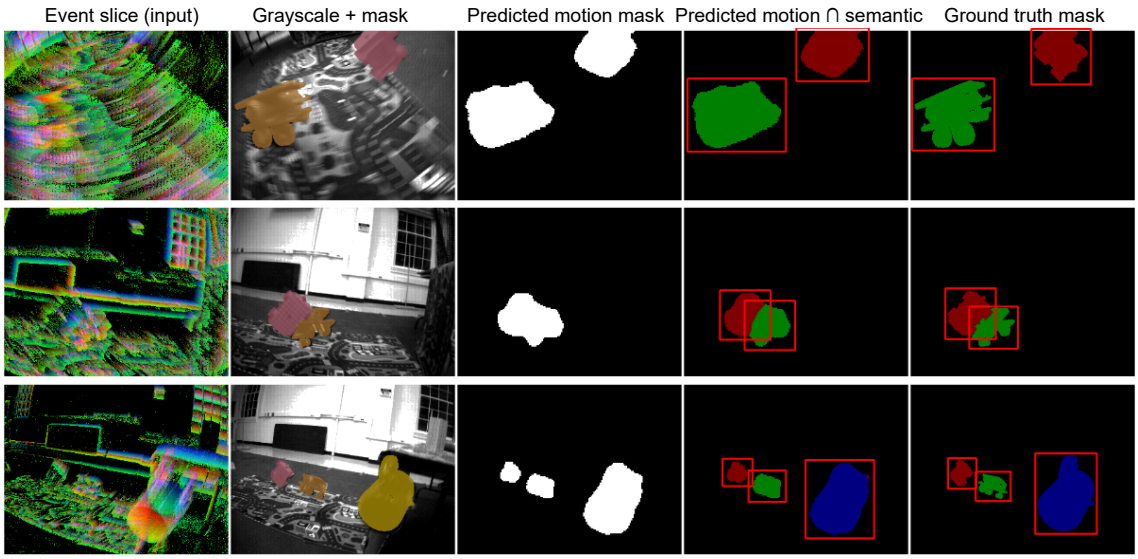

Figure 5: Qualitative Evaluation on EVIMO dataset. The table entries from left to right: event slice input, gray-scale image with ground truth mask, predicted motion mask, predicted motion mask with semantic labels, ground truth mask with semantic labels. There are three complex scenarios, from top to bottom: rapid camera rotations (row 1), overlapping IMOs (row 2), multiple objects at different distances (row 3).

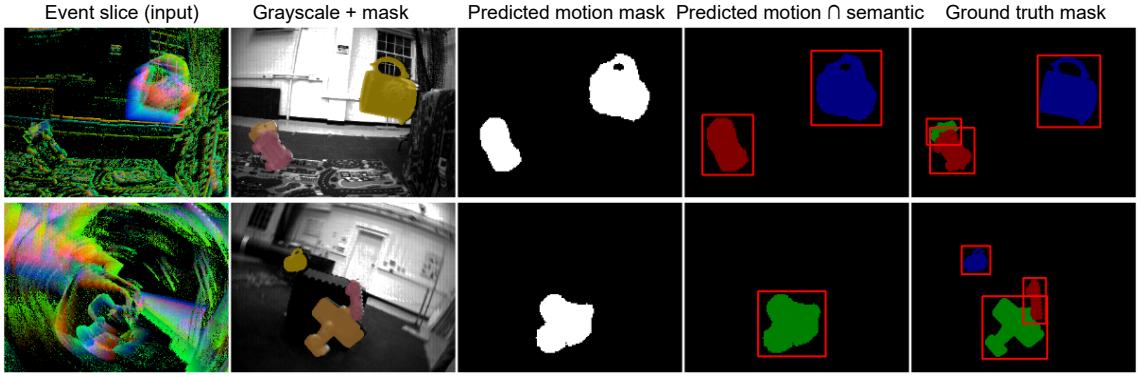

Figure 6: Two failure cases on EVIMO, from top to bottom: inability to separate overlapping objects with severe occlusion (row 1), missed detection of a small object with fast camera rotation (row 2).

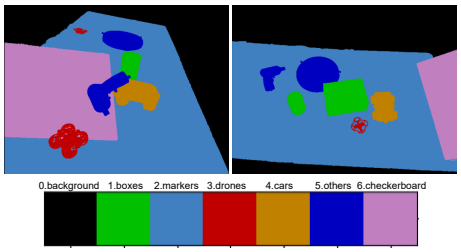

Figure 7: Reorganization of EVIMO2 labels. Segmentation examples and proposed labels and colors scheme.

Bounding box conversion: As explained in Sec. 4.3, we extract the boundary of the motion mask of each class to get the bounding box. The problem is that EVIMO2 has far more classes than EVIMO, some small errors in semantic labels generate a lot of wrong bounding boxes, which greatly affects the results of detection rate, precision and recall. We adopted the method of suppressing bounding boxes with an area smaller than the parameter *area threshold* to avoid it as much as possible. According to the size of the object in the EVIMO2 dataset, we define the *area threshold* as 3000 pixels. For example, in the last mask in the second row of Fig. 8, some pixels are mistakenly classified as drones (red), which would generate two small wrong bounding boxes if not applying this procedure.

The final quantitative evaluation results are shown in Tab. 6. We observe that in the sequences "test_13", most of moving objects fall under the "others" category. Consequently, the semantic segmentation results for these sequences are less satisfactory, leading

Table 5: Restructured semantic labels for EVIMO2.

Restructured labels	0.background	1.boxes (16228)	2.tabletop (12549)	3.drones (13658)	4.cars (19014)	5.others (14643)	6.checkerboard (8557)
Original labels*	background	box00 (267) box01 (5443) box02 (677) box03 (3012) can00 (6829)	tabletop (9841) marker00 (677) marker01 (677) marker02 (677) marker03 (677)	drone00 (6829) drone01(6829)	car00 (2437) car01 (6829) wheel00 (2437) wheel01 (2437) wheel02 (2437) wheel03 (2437)	knife00 (308) plane00 (677) plane01 (0) toy00 (6829) turntable (6829)	checkerboard (8557)

* From the objects.txt file in dataset EVIMO2.

() represents the number of images containing this label in EVIMO2 training set.

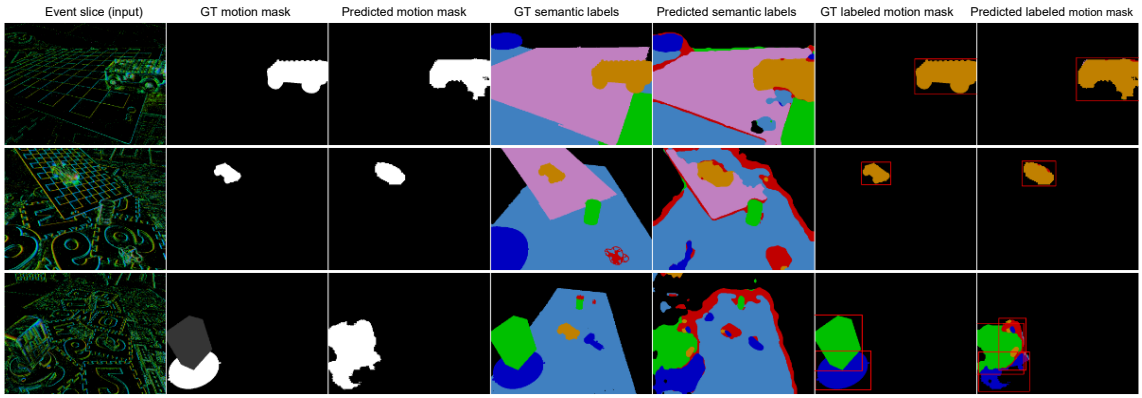

Figure 8: Qualitative Evaluation on EVIMO2 dataset. The table entries from left to right: event slice input, ground truth motion mask, predicted motion mask (output of motion task), ground truth semantic labels (7 classes introduced in Fig. 7), predicted semantic labels (output of semantic task), ground truth motion mask with labels, predicted motion mask with labels (intersection of two tasks). Three complex scenarios from top to bottom: object slowly accelerates from rest (row 1), distant IMO plus static objects (row 2), two objects suddenly accelerate (row 3).

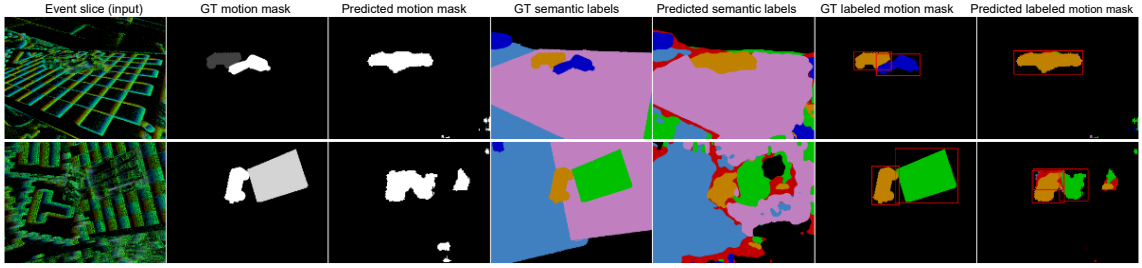

Figure 9: Two failure cases on EVIMO2, from top to bottom: wrong predicted semantic label for class "others" (row 1), less accurate prediction of motion at high changing motion speed (row 2).

to a relatively low mIoU. In the sequences "test_14", there are objects that exhibit rapid speed changes. As a result, the predictions tend to capture only the edges of these objects, which has a greater negative impact on detection rate, precision and recall, which require bounding box estimation.

EMSGC (Zhou et al., 2023) provides the IoU score of their method on the EVIMO2 dataset and this is the only score we found that was evaluated on the EVIMO2 dataset. Since there are more frequent 3D rotations than EVIMO, the 2D appearance of the object on the image plane is constantly changing, resulting the IoU score of their method compared to EVIMO not good enough. The IoU score of our method is almost the same as that of EVIMO, which proves that our method has strong generalization ability for different motions.

Biggest difficulty in EVIMO2 are the two states of IMOs: fast and slow motion. The more challenging scenario involves a significant change in speed, while the camera only moves slightly. In this case, when a moving object occupies a large portion of the frame, the network predict the object's outline. Events are not triggered inside the object's shape, leading to possible mis-prediction of multiple objects (as shown in the second row of Fig. 9). Although in EVIMO, IMOs are in a state of rapid motion with low speed change, and the camera is also moving at high speed. As a result, all objects generate a significant amount of event information as input to the network, enabling the net-

work to learn and predict more effectively.

In summary, the EVIMO2 dataset is more challenging than EVIMO. Through our qualitative and quantitative evaluations, we have demonstrated that our network is able to perform well when confronted with this highly challenging dataset.

Table 6: Comparison of proposed method on EVIMO2 using all proposed metrics with available state-of-the-art data.

Method	Sequence	Detection rate (%)	IoU (%)	mIoU (%)	Precision	Recall	F1-score
Ours	test13_00	80.71	77.94	41.93	0.85	0.76	0.80
	test13_05	46.38	62.81	37.02	0.67	0.32	0.43
	test14_03	54.88	87.53	84.25	0.96	0.51	0.67
	test14_05	34.67	85.97	72.82	0.45	0.32	0.37
	test14_05	46.40	83.97	74.13	0.50	0.36	0.42
	test15_01	83.60	81.84	52.96	0.46	0.48	0.47
	test15_02	73.04	81.21	59.51	0.89	0.59	0.71
	test15_05	71.75	78.04	70.43	0.40	0.51	0.45
	Average	61.43	**79.82**	61.63	0.65	0.48	0.55
EMSGC	*Average*	-	64.38	-	-	-	-

4.6 Ablation Study

In this section, we trained EVIMO and EVIMO2 separately using the same configuration, on different networks: "task 1" network with only motion segmentation, "task 2" network with only semantic segmentation, and proposed network with both tasks (all of these networks include depth prediction network). They are evaluated on all available metrics, in Tab. 7 and 8. For "task 1" network (motion seg.), only detection rate and IoU can be used to evaluate since there is no semantic information. For "task 2" network (semantic seg.), evaluation with proposed metrics is not straightforward as they aim at evaluating the semantic classification *of moving objects*. In EVIMO, all seen objects are moving, allowing to measure meaningful scores in Tab. 7. We also fuse object semantics to obtain a binary object masks generalized as a motion mask to measure detection rate and IoU. However, in the case of the EVIMO2, where multiple objects are stationary, we can not evaluate semantic segmentation task only with proposed metrics in Tab. 8.

The results show that the network trained for the two tasks together has the best effect, further verifying the beneficial contribution of semantic information to motion segmentation.

Table 7: Ablation study results of proposed motion and semantic segmentation network on EVIMO.

Depth Prediction Network	+Motion Seg. (task 1)	+ +Semantic Seg. (task 2)	+ Motion Seg. + Semantic Seg.
Params (M)	2.26	2.26	2.5
Detection rate (%)	81.75	70.71	82.64
IoU (%)	79.33	73.22	81.16
mIoU (%)	-	56.83	68.47
Precision	-	0.53	0.71
Recall	-	0.67	0.76
F1-score	-	0.59	0.73

Table 8: Ablation study results of proposed motion and semantic segmentation network on EVIMO2. All the scores of the network with both tasks can be retrieved in Tab. 6.

Depth Prediction Network	+Motion Seg. (task 1)	+ Motion Seg. + Semantic Seg.
Params (M)	2.26	2.5
Detection rate (%)	50.94	61.38
IoU (%)	77.92	79.88

5 CONCLUSIONS

We propose a network architecture for multi-motion and semantic segmentation using monocular event data. Our approach is built upon a binary motion segmentation state-of-the-art framework. Thanks to the help of added semantic segmentation task, it offers following improvements: better moving objects segmentation, ability to extract multiple IMOs of different classes. It learns in a supervised mode, can accurately predict motion masks and semantic information for multiple objects, and surpasses the state-of-the-art. We conducted qualitative and quantitative evaluations on two highly challenging datasets, EVIMO and EVIMO2, demonstrating the robustness of our method across various real scenes, multiple types of motion of IMOs and camera, and different semantics. However, further improvements are needed in predicting the shape of IMOs, such as enhancing the dataset and optimizing the motion prediction module.

The design of multi-task neural networks to share encoding layers is promising. Specifically, the network could be extended to include heads for depth and direct optical flow inference, while simplifying the dataset requirements by adding self-supervised losses such as in (Ye et al., 2020) and (Stoffregen et al., 2020). Some perspectives can be to extend the multi object segmentation to multi object tracking, to test adaptive time windows for event accumulation in order to adapt to different IMOs speeds, or to focus on richer event representation methods for the inputs to improve the network. Our next goal would be to study other types of datasets, such as the road scene DSEC dataset (Gehrig et al., 2021), by adding static and moving status to objects in the annotations, to explore broader applications of event cameras in the field of motion segmentation.

Acknowledgment

This work has been carried out within SIVALab, joint laboratory between Renault and Heudiasyc UMR UTC/CNRS.

REFERENCES

Alonso, I. and Murillo, A. C. (2019). Ev-segnet: Semantic segmentation for event-based cameras. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Burner, L., Mitrokhin, A., Fermuller, C., and Aloimonos, Y. (2022). 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*, abs/2205.03467.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., and Scaramuzza, D. (2022). Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gehrig, M., Aarents, W., Gehrig, D., and Scaramuzza, D. (2021). Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*.

Glover, A. and Bartolozzi, C. (2016). Event-driven ball detection and gaze fixation in clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*.

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*.

Litzenberger, M., Posch, C., Bauer, D., Belbachir, A., Schon, P., Kohn, B., and Garn, H. (2006). Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In *IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*.

Mitrokhin, A., Fermüller, C., Parameshwara, C., and Aloimonos, Y. (2018). Event-based moving object detection and tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Mitrokhin, A., Hua, Z., Fermüller, C., and Aloimonos, Y. (2020). Learning visual motion segmentation using event surfaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mitrokhin, A., Ye, C., Fermüller, C., Aloimonos, Y., and Delbruck, T. (2019). Ev-imo: Motion segmentation dataset and learning pipeline for event cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Parameshwara, C. M., Sanket, N. J., Singh, C. D., Fermüller, C., and Aloimonos, Y. (2021). 0-mms: Zero-shot multi-motion segmentation with a monocular event camera. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Piatkowska, E., Belbachir, A. N., Schraml, S., and Gelautz, M. (2012). Spatiotemporal multiple persons tracking using dynamic vision sensor. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.

Rebecq, H., Gehrig, D., and Scaramuzza, D. (2018). Esim: an open event camera simulator. In *Conference on robot learning*. PMLR.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer (MICCAI)*. Springer.

Sanket, N. J., Parameshwara, C. M., Singh, C. D., Kuruttukulam, A. V., Fermüller, C., Scaramuzza, D., and Aloimonos, Y. (2019). Evdodge: Embodied AI for high-speed dodging on A quadrotor using event cameras. *CoRR*, abs/1906.02919.

Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., and Scaramuzza, D. (2019). Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Stoffregen, T. and Kleeman, L. (2018). Simultaneous optical flow and segmentation (sofas) using dynamic vision sensor. *arXiv preprint arXiv:1805.12326*.

Stoffregen, T., Scheerlinck, C., Scaramuzza, D., Drummond, T., Barnes, N., Kleeman, L., and Mahony, R. (2020). Reducing the sim-to-real gap for event cameras. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*. Springer International Publishing.

Sun, Z., Messikommer, N., Gehrig, D., and Scaramuzza, D. (2022). Ess: Learning event-based semantic segmentation from still images. In *European Conference on Computer Vision*. Springer.

Vasco, V., Glover, A., Mueggler, E., Scaramuzza, D., Natale, L., and Bartolozzi, C. (2017). Independent motion detection with event-driven cameras. In *18th International Conference on Advanced Robotics (ICAR)*.

Ye, C., Devaraj, C., Maynord, M., Fermüller, C., and Aloimonos, Y. (2018). Evenly cascaded convolutional networks. In *IEEE International Conference on Big Data (Big Data)*.

Ye, C., Mitrokhin, A., Fermüller, C., Yorke, J. A., and Aloimonos, Y. (2020). Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Zhou, Y., Gallego, G., Lu, X., Liu, S., and Shen, S. (2023). Event-based motion segmentation with spatio-temporal graph cuts. *IEEE Transactions on Neural Networks and Learning Systems*.