IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 June 2023, accepted 21 July 2023, date of publication 28 July 2023, date of current version 4 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299597

RESEARCH ARTICLE

A Multi-Scale Recurrent Framework for Motion Segmentation With Event Camera

SHAOBO ZHANG[1], LEI SUN[1], AND KAIWEI WANG[1,2], (Member, IEEE)
[1]State Key Laboratory of Modern Optical Instrumentation, Zhejiang University, Hangzhou 310027, China
[2]National Engineering Research Center of Optical Instrumentation, Zhejiang University, Hangzhou 310027, China

Corresponding author: Kaiwei Wang (wangkaiwei@zju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 12174341, and in part by the National Key Research and Development Program of China under Grant 2022YFF0705500 and Grant 2022YFB3206000.

ABSTRACT Motion segmentation is a formidable computer vision task, aiming to segment moving targets from a dynamic scene. In this paper, we choose to introduce an additional modality to bolster the robustness. The event camera is a bio-inspired sensor that accurately detects and captures intensity changes with exceptional temporal resolution and dynamic range, which is an optimal choice for motion segmentation. Therefore, we present a novel framework for event-based motion segmentation and propose Multi-Scale Recurrent Neural Network (MSRNN) to fuse temporal information efficiently. To our best knowledge, it is the first time that a multi-scale recurrent architecture is implemented in event-based motion segmentation. The proposed framework is evaluated through experiments conducted on the EV-IMO dataset. Our method achieves a mean Intersection-over-Union (mIoU) of 82.0%, which sets a new state-of-the-art in motion segmentation. To further validate our approach in arduous real-world scenarios, we introduce the Event Challenging Motion dataset, consisting of 350 images and corresponding events, in which our method outperforms the other methods by 1.5% in Intersection-over-Union (IoU).

INDEX TERMS Motion segmentation, event cameras.

I. INTRODUCTION

Motion segmentation aims to predict motion masks to understand scene dynamics. Motion segmentation allows applications such as robotics to focus on moving objects or ignore them based on task requirements. While image-based motion segmentation has advanced rapidly in recent years [1], [2], motion segmentation is still hindered by the drawbacks of frame-based cameras, which inevitably introduce motion blur and image degradation, as well as image information defects resulting from high dynamic range.

Event cameras [3], [4], [5], offer high temporal resolution and dynamic range, are widely applied in optical flow estimation [6], [7], [8], [9], [10], [11], [12], image deblurring [13], [14], [15], [16], [17], [18], frame reconstruction [19], video frame interpolation [20], human pose estimation [21], fast auto-focus [22], and other computer vision tasks. Distinct from traditional image sensors such as CMOS and CCD, event cameras function as bionic technology by capturing asynchronous temporal intensity changes of a scene as a continuous events stream. Thus, event cameras can detect both moving objects in a dynamic scene along with the background's motion caused by the inevitable movement of cameras. Moreover, they exhibit exceptional temporal resolution and dynamic range, making them suitable for handling complex and challenging scenarios in motion segmentation.

Motion segmentation is to distinguish between moving objects and the background. Given a dynamic scene, the target of motion segmentation is to tell the moving objects from the whole scene. It can be implemented in a drone or a walking robot. These machines need to accurately perceive and respond to rapidly moving objects in the scene, especially in extreme conditions. Thus, we desire to design a more robust framework for motion segmentation that can be adapted to a fast or low-illumination scene. Temporal information plays an important role in this task. Most learning-based methods take multiple image frames or additional data for motion segmentation [1], [23], [24], [25]. Meunier et al. utilize only optical

The associate editor coordinating the review of this manuscript and approving it for publication was Angel F. García-Fernández.

flow information for motion segmentation [26]. However, the drawbacks of images and low-quality additional information will reduce the accuracy. Instead, we utilize events for motion segmentation due to the extraordinary property of event cameras.

Recent learning-based motion segmentation networks like [25], [26], and [27] are simply based on UNet [28]. It is first implemented in biomedical segmentation and is found to be useful in current universal segmentation. In the UNet, the encoder extracts the overall and local information from the event frames to estimate the pose of the moving object, and the decoder aims to fuse the features from the encoder and reconstruct the contour and position of the moving object. The multi-scaled features from each stage of the encoder contain information of the image from different levels. Large-scale features provide more edge and contour information, while small-scale features contain richer semantic information about the moving object and background. However, This framework does not utilize long-range temporal information for motion segmentation, which causes less temporal consistency.

Based on the previous work, our method, Multi-Scale Recurrent Neural Network (MSRNN), focuses on the temporal consistency of a dynamic scene. MSRNN fuses multi-scaled features from the previous time step to get more accurate motion estimation. Besides, the iterative recurrent architecture provides no extra learnable parameters and is easy to train.

In this work, we explore the potential of events for motion segmentation and propose Multi-Scale Recurrent Neural Network (MSRNN) that effectively fuses long-range temporal information from the previous time steps. To our best knowledge, it is the first time that a recurrent architecture is implemented in event-based motion segmentation. To improve the robustness of motion prediction on both large and small scales, we propose a multi-scale recurrent architecture that incorporates a recurrent block at every encoder stage. Specifically, the spatial size of the feature maps is halved after each block, which helps improve the prediction of large motion and small motion, respectively. We conduct experiments and compare our method with state-of-the-art motion segmentation methods on the EV-IMO dataset [27], and prove its effectiveness through a detailed ablation study. Next, we collect a new event motion segmentation dataset named Event Challenging Motion (ECMotion) in a laboratory setting with a SEEM1 event camera. Our dataset contains 350 frames in total, under varying light conditions, and 150 frames are annotated with ground-truth. Furthermore, we perform extensive comparisons against an image-based framework and other competitive methods on the ECMotion dataset, demonstrating the superiority of our event-based motion segmentation framework.

In summary, our contributions are as followings:

1) We utilize events to improve motion segmentation and propose Multi-Scale Recurrent Neural Network (MSRNN) for event-based motion segmentation.
2) Our motion segmentation model achieves the new state-of-the-art for motion segmentation on the EV-IMO dataset.
3) A novel dataset for evaluation on real-world high-speed motion segmentation is proposed. Several methods are evaluated on the proposed dataset.

Both the code and ECMotion dataset are available at https://github.com/shaobo007/msrnn.

II. RELATED WORK

A. IMAGE-BASED MOTION SEGMENTATION

Motion segmentation is a fundamental computer vision task. Hand-crafted algorithms such as [29], [30], and [31] separate the optical flow into 'layers' modeled by an affine motion, in which the robustness depends on the performance of the optical flow algorithms. Following researches utilize Bayesian treatment [32] to enhance multi-body factorization [33]. Brox et.al propose to integrate the motion segmentation into the variation formulation of the optical flow estimation with level sets [34]. Before the advent of deep learning, several more notable methods appear. And most of them build upon previous ideas, including advanced trajectory-based methods [35], and a Conditional Random Field (CRF) based approach [36]. Nevertheless, the speed, robustness, and performance of all the above algorithms cannot compete with modern learning approaches.

In recent years, motion segmentation has made significant progress through the utilization of the Convolution Neural Network (CNN). Tokmakov et al. extract the feature map of each frame and then incorporate the features of adjacent frames to establish motion masks [23]. Shen et al. predict motion masks using a lightweight UNet [28] in their pipline [25]. The community has witnessed several innovative components and methods, including multi-fusion architecture [1], [2], [24], [37], partially supervised networks [38], fully unsupervised network [26], and Recurrent Neural Network [39]. Despite their extraordinary effect on motion segmentation, image-based methods still struggle when facing real-world scenarios, particularly in extreme conditions such as high-speed motion, and low-illumination conditions.

B. EVENT-BASED MOTION SEGMENTATION

Recently, a number of event-based motion segmentation algorithms appears. Lagoree et al. [40] propose a kernel function-based method for segmenting moving objects. In another work, Mitrokhin et al. [41] propose a motion detection and tracking algorithm using time images, compensating for camera motion while tracking moving objects. Moreover, they propose a challenging event-based dataset called EED, which contains five different scenes and bounding boxes of moving objects. Zhou et al. create a space-time event graph and pass it to an iterative clustering algorithm to predict scene motion [42]. Chen et al. propose a mutually reinforced framework both for motion estimation and event denoising [43]. However, these conventional approaches

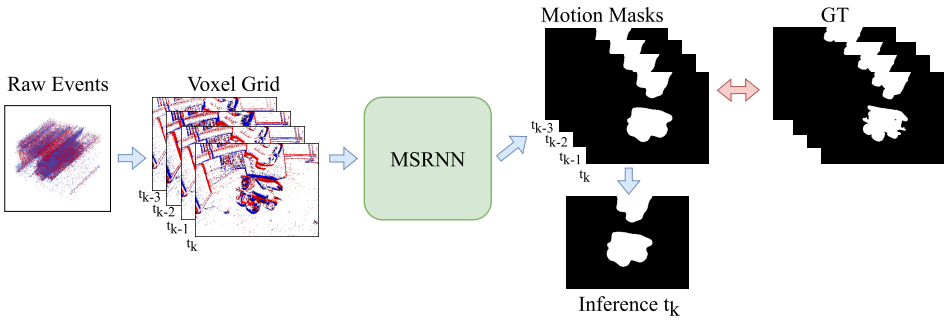


FIGURE 1. The proposed framework for motion segmentation using the representation of voxel grid. The raw events are first converted into a voxel grid. A pack of adjacent frames is then input into the MSRNN to generate motion masks for each time step. We compare the masks with their ground truths respectively. Notably, the current step t_k is used for inference.

perform in event space and can be extremely impacted by the event noise.

Mitrokhin et al. [27] introduce the first event-based motion segmentation dataset called EV-IMO which contains depth maps, motion masks, camera, and object motion information. They also present a deep convolution neural network based on UNet [28] to predict motion masks for applications with limited scenes, such as robotics. This method utilizes early fusion by concatenating the input feature maps of adjacent frames, however, it doesn't take full advantage of long-range temporal information because the events utilized in their work are near the timestamp of the target time, which discards previous and future events for long-range temporal information. In another work, they propose a method using event surface and a Graph Neural Network (GNN) [44]. Though this approach treats each event as a node in the GNN, resulting in improved training and inference times. However, GNN-based methods still struggle with training instability.

Most recent image-based methods like multi-fusion [1], [23], [24], [25] take additional information like optical flow or depth as input, thus, it takes more effort to accomplish the motion segmentation pipeline and low-quality optical flow can deteriorate the segmentation result. The RNN-based method [39] does not utilize multi-scaled information and still takes additional optical flow as input. Our method is most similar to UNet [28] or SfM-Net [38], which are image-based networks. However, we utilize voxel grid [6] as input which is perfectly compatible with these image-based methods. Moreover, we design a novel multi-scaled recurrent architecture to fuse long-range temporal information from adjacent time steps for analyzing the relative pose change to learn the motion mask. The multi-scale architecture extracts local and global features, helping alleviate the noise influence of the event camera.

III. METHOD

A. FRAMEWORK

Event cameras [5] respond only to changes in brightness in the log domain of the photocurrent intensity, i.e. $L = \log(I)$. If the brightness change $\Delta L(\mathbf{x}_i, t_i)$ comparing the previous event at pixel $\mathbf{x}_i = (x_i, y_i)^\top$ exceeds the threshold C, the i-th event $e_i = (\mathbf{x}_i, t_i, p_i)$ is triggered at time t_i by the brightness increase (polarity $p_i = 1$) or decrease (polarity $p_i = -1$)

$$\Delta L(\mathbf{x}_i, t_i) = L(\mathbf{x}_i, t_i) - L(\mathbf{x}_i, t_i - \Delta t_i), \tag{1}$$

$$p_i = \begin{cases} +1, \text{ if } \Delta L(\mathbf{x}_i, t_i) > C, \\ -1, \text{ if } \Delta L(\mathbf{x}_i, t_i) < -C, \end{cases} \tag{2}$$

where Δt_i denotes the time gap since the previous event at the same pixel $\mathbf{x}_i$, and p_i indicates the polarity of brightness change. The inherent characteristics of event cameras make them suitable for capturing dynamic and fast scenes, particularly under challenging light conditions.

In our work, we explore the potential of events for motion segmentation. Due to their high temporal resolution without motion blur, event cameras are ideal for motion segmentation in dynamic scenes. Fig. 1 shows the proposed event-based motion segmentation framework. We first convert an event stream with a time interval of T into a voxel grid [6] with channel dimension. Each channel consists of the accumulated events within a T/C time frame, thus partially maintaining the raw data's temporal information. Subsequently, we process this voxel grid using a CNN-based network to predict motion masks. Each prediction can be used to predict the consecutive frame.

Our pipeline can be expressed as (3). Here $\sum_{t=t_k}^{T} e_t$ indicates the events for time t_k, V represents the transformation of voxel grid, and C_{k-1} and H_{k-1} represents the multi-scale features from the previous frames. Furthermore, F_e represents the encoder's transformation, with Θ_1 representing its learnable parameters, and F_d is the decoder's transformation, with Θ_2 representing its learnable parameters.

$$f_o = F_d(F_e(V(\sum_{t=t_k}^{T} e_t), C_{k-1}, H_{k-1}, \Theta_1), \Theta_2) \tag{3}$$

In our proposed framework, we utilize two paradigms to encode the time information: voxel grid representation and recurrent architecture. The raw events of a specific time

period $\sum(x, y, t, p)$ is converted into voxel grid represented as $R^{H\times W\times C}$, where C represents the different time periods. Raw events are not compatible with CNN-based methods because of their asynchronous nature. Voxel grid is a dense representation for events that can be applied to the CNN-based framework and is widely used in event-based optical flow estimation task [6], [10], [11], [12] to learn the scene motion. Compared to other dense event representations like event frame [45], motion-compensated event image [46] and time surface [47], voxel grid has variant channels with finer time information and it sustains the polarity information within the period of time. Compared to raw events, the time information is discretized, leading to the loss of accurate temporal information compared to the raw event stream. The other paradigm is recurrent architecture. We meticulously design a novel recurrent network based on UNet [28] that efficiently fuses previous adjacent frame features to predict the motion masks. Based on these two paradigms, our proposed recurrent network for event cameras achieves an 82.0% mIoU score on motion segmentation, demonstrating our pipeline's feasibility.

B. MODEL ARCHITECTURE

To efficiently feed the features of previous frames to our network, we employed the UNet as the foundation and designed a Muti-Scale Recurrent Neural Network (MSRNN) to improve the accuracy of motion segmentation. Each UNet unit of our architecture mainly composes an encoder and a decoder, depicted in Fig. 2. The decoder contains four successive decoder blocks and culminates with a Sigmoid output layer. Table 1 presents comprehensive details of the network layer's input and output size and the number of channels. The encoder block is composed of two consecutive convolutional layers, a Channel-Wise Attention (CA) block [48], and a Long Short-Term Memory (LSTM) [49] block, as shown in Fig. 3.

As illustrated in Fig. 4 (a), the CA block includes a branch to learn the channel weight and a shortcut to connect the input feature. This module multiplies the weight with the output of the network layer. We unfold high-level features $f^h \in R^{W\times H\times C}$ as $f^h = [f_1^h, f_2^h, f_3^h, \cdots, f_C^h]$, where $f_i^h \in R^{W\times H\times 1}$ represents the feature of the i-th channel and C is the total channel number. CA weight $\omega^h \in R^c$ is extracted from f^h through the CA network layer. First, each f_i^h is transformed to a channel-wise feature vector $v^h \in R^C$ through average pooling. Then $\omega^h \in R^c$ is obtained through a Fully Connected layer (FC) followed by a ReLU activation layer, another FC layer, and then a Sigmoid activation layer. $\omega^h \in R^c$ is the weight of each channel which is mapped to [0, 1]. Finally, the module's output $\widetilde{f^h}$ is obtained by weighting the original input feature with $\omega^h \in R^c$. CA block composes a sequence of layers that assign a weight to each channel of the original feature. Through the CA block, the temporal information of the multi-scale feature of each encoder block can be enhanced, resulting in the improvement of the accuracy of predicting more fine-grained mask edges.

TABLE 1. Detailed structure of proposed network architecture. ("E": Encoder, "D": Decoder, "conv": Convolution layer).

Name	Input Size	Output size	Channel (m,n,k)
In conv	$256\times336\times3$	$256\times336\times64$	3, 64, 3
E block 1	$256\times336\times64$	$128\times168\times128$	64, 128, 3
E block 2	$128\times168\times128$	$64\times84\times256$	128, 256, 3
E block 3	$64\times84\times256$	$32\times42\times512$	256, 512,3
E block 4	$32\times42\times512$	$16\times21\times1024$	512, 1024, 3
D block 1	$16\times21\times1024$	$32\times42\times512$	1024, 512, 3
D block 2	$32\times42\times512$	$64\times84\times256$	512, 256, 3
D block 3	$64\times84\times256$	$128\times168\times128$	256, 128, 3
D block 4	$128\times168\times128$	$256\times336\times64$	128, 64, 3
Out conv	$256\times336\times64$	$256\times336\times1$	128, 64, 1, 2

In our MSRNN model, every encoder block consists of an LSTM block. Figure 4 (b) illustrates the fundamental components of LSTM, including a memory cell, an input gate, an output gate, and a forget gate. The memory cell stores the previous values of the cell and its states, while the three gates regulate how much of the previous cell state to "forget" or "remember" when processing a new input. Through LSTM, each stage of the encoder can transmit information from adjacent frames effectively. The LSTM block takes the previous frames' state (c_{k-1}^m, h_{k-1}^m) as input, generating an output state (c_k^m, h_k^m) to assist with predicting the following frame. This spatial design of the model accomplishes a multi-scale recurrent architecture.

C. MULTI-SCALE RECURRENT ARCHITECTURE

To utilize long-range temporal information from events, and make full use of multi-scale features, we design MSRNN with a multi-scale recurrent architecture, as illustrated in Fig. 2. Compared with a traditional RNN, MSRNN transmits multi-scaled features which offer more sufficient information from the consecutive time step and the RNN architecture achieved by LSTM blocks makes it more trainable. The encoder includes four stages that receive features (c_{k-1}, h_{k-1}) from the previous frame, and forward new features (c_k, h_k) of respective scales of [1/2, 1/4, 1/8, 1/16] to the subsequent frame. The scales are chosen based on the shape of features output from each of the encoder blocks based on UNet. The input size is 256 × 336 and 1/16 scale of it is enough for extracting low-level features. Thanks to the previous feature input of multiple scales, our network enables more accurate prediction of both large and small objects. We utilize LSTM units to incorporate long-range information to enhance the prediction of the present frame and improve the temporal consistency. As shown in Fig. 3, each stage of the encoder contains a LSTM block, as a role to interconnect adjacent time steps. Adjacent frames are highly correlated on motion segmentation, so the LSTM block can improve the accuracy of prediction and maintain temporal consistency. Our proposed multi-scale architecture can preserve multi-scale and multi-level features from preceding steps, serving as a source of prior knowledge for the ongoing

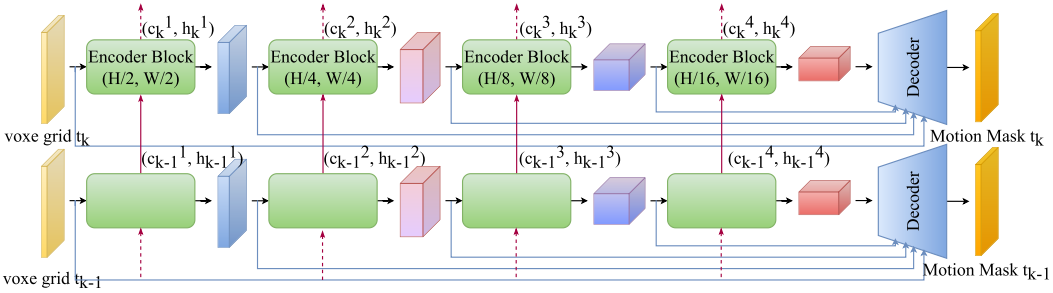


FIGURE 2. The architecture of our Multi-Scale Recurrent Neural Network (MSRNN) based on events. (c_k^m, h_k^m) represents the output state from the m-th encoder block at time step t_k.

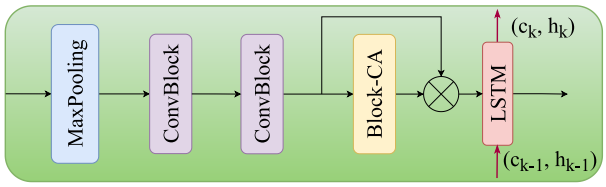


FIGURE 3. The structure of encoder block. MaxPooling: 2 × 2 MaxPool2d layer, ConvBlock: a 3 × 3 convolution layer, a batch norm layer, and a ReLu activation layer. The LSTM unit takes as input the previous state (c_{k-1}, h_{k-1}) and generates present state (c_k, h_k) for the next prediction.

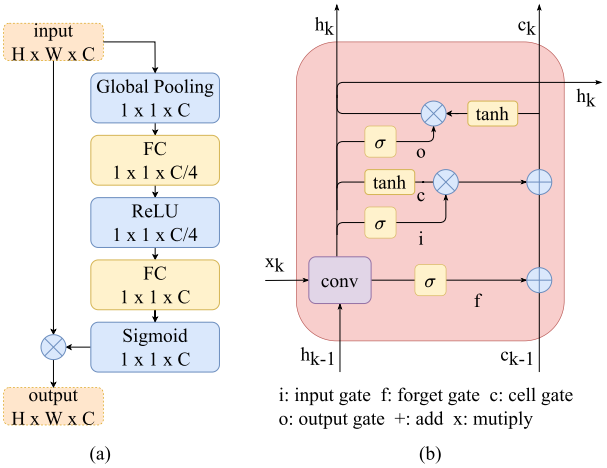


FIGURE 4. (a) : The structure of Block-CA. FC: fully-connected layer. (b): The detailed structure of LSTM block. The *conv* layer takes as input the current features x_k and previous state h_{k-1} and generates four outputs corresponding to the four gates, respectively.

stage with concentrated information from previous frames augmenting overall robustness.

IV. EXPERIMENTS

A. DATASET

The EV-IMO dataset [27] serves as the main dataset of our research, representing the first event-based dataset to encompass both camera motion and multiple moving objects. The data is collected from specific scenes captured by the DAVIS-346C event camera with a resolution of 260 × 346 and a 70° field of view, for around 30 minutes in total. Each recorded sequence presents no more than three objects, with a true mask provided for each object at a rate exceeding 200 frames per second.

EED dataset [41] contains limited samples with annotated bounding boxes but no motion masks, and it has no train set, causing it to be not appropriate for our learning framework. Compare to the synthetic dataset MOD++, EV-IMO contains only real-world data, which our work focuses on. Therefore, we only utilize EV-IMO dataset for our research.

The EV-IMO dataset includes 34 high-quality sequences for training, featuring main scenes of boxes, floor, table, tabletop, and wall. It also includes 21 sequences for validation, encompassing the main scenes of the boxes, fast, floor, table, tabletop, and wall. The original sequences of the EV-IMO dataset are recorded in seconds or minutes, rendering them inadequate for training. Consequently, we divide the data into multiple time slices with ground-truth corresponding to each slice, respectively serving as training samples. With image-based ground-truth being generated at 40 frames per second, the interval between any two adjacent ground-truth' timestamps is roughly 25 μs. The true mask (ground-truth) of each moving object is saved in image form, marked with its corresponding timestamp. We take each timestamp corresponding to the ground-truth image as the slice center, with a length of 0.03 s. Each slice is represented as an event matrix of $N \times 4$, where N indicates the number of events in 0.03 s. Then, the matrix is converted into a voxel grid format, creating an image-like matrix of $3 \times 260 \times 346$. We further adjust the shape to $3 \times 256 \times 336$ by center cutting. The 0-th dimension holds the total event integral within a given time interval. Given a time span of 0.03 s, we have a channel of 3, and thus, $t = T/3 = 0.01$ s represents each channel's timespan. By this representation of events, we can extract features from events and treat each slice as an image to train a convolutional neural network and learn the motion mask.

Eventually, we yield around 15,000 samples for training and 5,300 samples for validation.

B. IMPLEMENTATION DETAIL

To address the data imbalance in the dataset, we employ a hybrid loss function that combines Focal Loss [50]

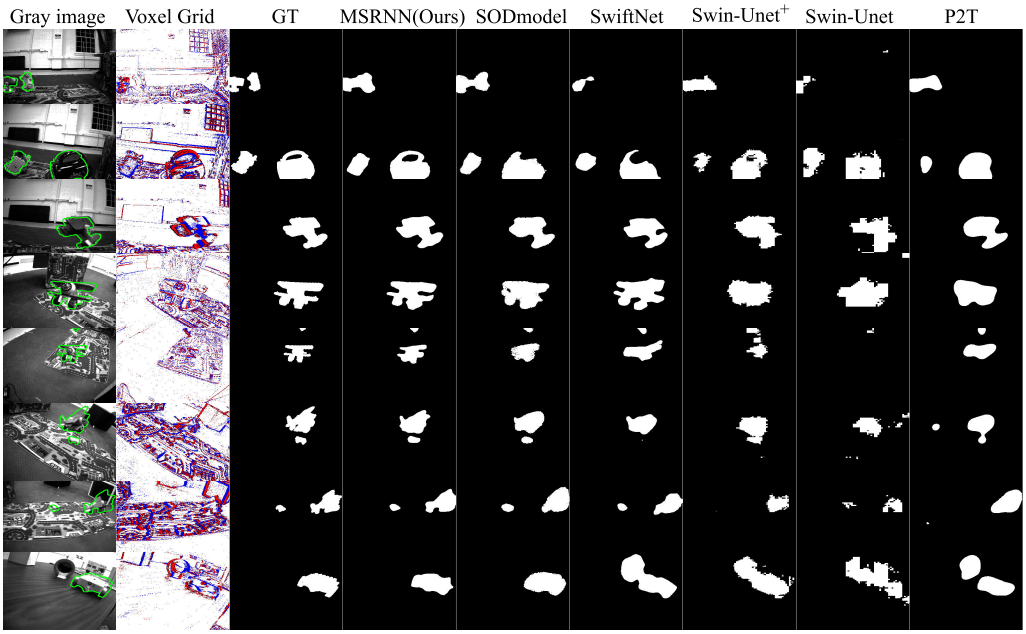


FIGURE 5. Qualitative results from our experiments. The contours marked in green represent the moving object. Compared to other methods, our method achieves better edge segmentation of moving objects.

TABLE 2. mIoU (%) results of motion segmentation methods on each scene of the EV-IMO dataset [27]. The results of EV-IMO method and GConv are from [44]. "mIoU": mean IoU value of moving objects and the background. "Swin-UNet+": an updated version of Swin-UNet with a multi-scale recurrent architecture.

Model	Boxes	Fast	Floor	Table	Tabletop	Wall
EV-IMO method [27]	70.0	67.0	59.0	79.0	n/a	78.0
GConv [44]	60.0	39.0	55.0	57.0	n/a	51.0
SwiftNet [52]	73.5	69.9	80.0	77.3	79.9	77.8
SODModel [48]	77.5	**75.9**	84.1	83.6	85.2	82.3
Swin-UNet [53]	59.5	57.9	66.0	61.6	57.0	65.6
Swin-UNet+ [53]	69.7	68.4	74.5	72.1	68.5	74.6
P2T [54]	74.8	71.6	79.0	79.3	78.6	78.8
MSRNN (Ours)	**79.2**	75.6	**85.1**	**84.5**	**87.1**	**82.3**

TABLE 3. Comparison of motion segmentation methods on EV-IMO dataset [27] (IoU>50% and IoU>60%). "IoU": the IoU value of moving objects.

Model	IoU > 50%	IoU > 60%	Runtime (ms)
SwiftNet [52]	0.739	0.579	8.0
SODModel [48]	0.883	0.772	7.4
Swin-UNet [53]	0.175	0.046	16.5
Swin-UNet+ [53]	0.571	0.342	17.0
P2T [54]	0.789	0.620	26.4
MSRNN (Ours)	**0.894**	**0.787**	**6.5**

and Dice Loss [51], which is defined as:

$$\ell = \ell_F + \ell_D = \alpha_t \, (1 - e^{-\hat{\ell}})^{\gamma} \cdot \hat{\ell} + 1 - \frac{2 * \sum x_i \cdot y_i}{\sum x_i + \sum y_i} \quad (4)$$

where $\hat{\ell} = \ell_b(\sum x_i, \sum y_i)$ is the BCEloss, and x_i and y_i denote the prediction and ground-truth respectively. Based on the spatial architecture design (Fig. 2), we propose a multi-frame loss that summarizes four adjacent frames' losses as a batch loss to supervise the training.

We train our network for 15 epochs using the NVIDIA Titan XP with a batch size of 4, for approximately 10 hours. We implement the Root Mean Square Propagation (RMSprop) optimizer with a Warm-Up scheduler. We set the initial learning rate to 1×10^{-5}, and implement a Warm-Up scheduler for 10 epochs.

C. RESULTS

For each validation dataset scenario, we evaluate the accuracy of the samples by measuring the Intersection over Union (IoU) and the mean IoU (mIoU) for segmentation.

In this subsection, we compare our MSRNN with state-of-the-art models or network architectures, namely the pyramid feature attention network (SODModel) [48] and SwiftNet [52], Swin-UNet [53] and P2T [54] on the EV-IMO dataset. We compare our method with the results of EV-IMO method [27] and GConv [44] which are given by [44]. We do not provide the results of [41] and [42]. They perform in event space and utilize the metrics in the form of success rate, which is different from the metrics we use in the paper. Thus, we can not obtain the same form of quantitative results as our pipeline. We present motion segmentation results in Table 2. It is evident from the results that our MSRNN outperforms the other methods in most of the scenarios set on the EV-IMO dataset, resulting in a mean improvement of 0.9% in mIoU compared to SODModel. In addition, we improve Swin-UNet and create a multi-scale recurrent version, named Swin-UNet+. The result shows that the multi-scale recurrent architecture can be applied to the transformer and make considerable improvement.

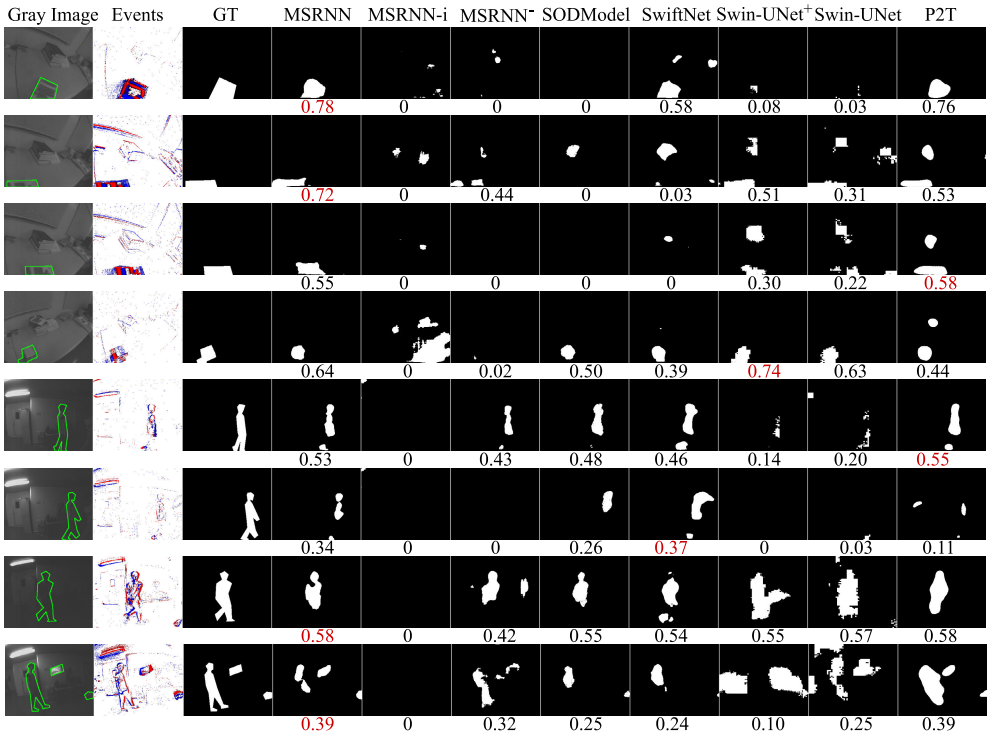


FIGURE 6. The visualization of results on the ECMotion dataset. The contours marked in green represent the moving object. The IoU score marked in red in each row represents the best result. MSRNN-i denotes the MSRNN model trained on gray images only on the EV-IMO dataset, while the others are trained with events. Each row is from different light conditions (Proper: 1, 2, 3, 4; Poor: row 7; HDR: Row 5, 6, 8).

TABLE 4. Ablation study of various factors of our method on the EV-IMO dataset [27]. We halve the channel numbers of each convolution layer to conduct the ablation study. "IoU": IoU value of moving objects.

Input	CA	Recurrent	Multi-scale Recurrent block		IoU (%)	mIoU (%)
Events	✗	✗	✓	LSTM	63.6	79.6
Events	✗	✓	✓	LSTM	65.8	81.0
Events	✗	✓	✗	LSTM	24.8	55.5
Events	✗	✓	✓	GRU	63.2	79.5
Images	✓	✓	✓	LSTM	70.8	83.8
Events	✓	✓	✓	LSTM	65.9	81.0

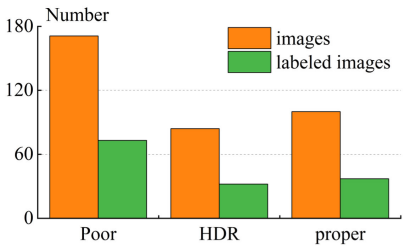


FIGURE 7. Data categories distribution of the proposed ECMotion dataset.

TABLE 5. IoU (%) results of motion segmentation methods on our proposed ECMotion dataset. "IoU": the IoU value of moving objects. MSRNN⁻ represents our MSRNN without the RNN architecture.

Model	Poor	HDR	Proper	All
SwiftNet(Events) [52]	40.7	27.5	27.7	35.6
SODModel(Events) [48]	37.1	25.1	7.8	30.8
Swin-UNet(Events) [53]	33.5	20.2	18.3	29.7
Swin-UNet⁺(Events) [53]	30.5	2.3	30.5	24.5
P2T(Events) [54]	39.2	**38.4**	**39.4**	**39.0**
MSRNN (Images)	0.0	0.2	0.1	0.1
MSRNN⁻ (Events)	37.6	15.9	11.2	30.0
MSRNN (Events)	**40.7**	28.2	34.8	37.1

TABLE 6. The results of the image-based and event-based MSRNN on ECMotion dataset after fine-tuning on the train set of ECMotion dataset.

Model	IoU	mIoU
MSRNN(Image)	87.1	93.1
MSRNN(Event)	**91.2**	**95.2**

Additionally, We evaluate the models based on the proportion of samples with IoU values greater than 50% and 60%. This metric denotes the accuracy of successful prediction, where the predicted mask overlaps the ground-truth by at least 50% and 60%, respectively. The quantitative results including runtime are illustrated in Table 3, indicating that our method performs better than the others. However, we are more concerned with accuracy than speed in this vision task.

We further provide a visual comparison of our MSRNN and the other models, illustrated in Fig. 5, demonstrating that MSRNN exhibits the best segmentation results. For instance, for the segmentation of a drone wing contour, our model yields astounding results, accurately segmenting the

intricate details of the object, while the other models struggle. Moreover, our proposed model shows better robustness than the other models when dealing with multiple objects.

D. ABLATION STUDY

We have conducted additional experiments to investigate the impact of various factors on our network's performance, including the model input, the CA module, the presence of recurrent architecture, multi-scale architecture, and the type of recurrent unit. As shown in table 4, by simply introducing a multi-scale recurrent architecture for fusing adjacent frames, an improvement of 2.2% in IoU over the base model was achieved, confirming the critical role of the structure. Without multi-scale features, which means we use the same scale for every layer of MSRNN, the results show the performance downgrades, as illustrated in Table 4. Furthermore, encoders with integrated CA modules improve the robustness of our model. Moreover, we find that the recurrent block of "LSTM" outperforms that of "GRU" [55].

The model trained on images in the dataset outperforms that trained on event data. The primary reason behind the improved performance is that frames contain more comprehensive texture information of objects. To address this issue, we propose a novel dataset for event-based motion segmentation of various moving objects under challenging conditions.

E. EXPERIMENTS ON ECMotion DATASET

We introduce a novel dataset with ground-truth annotations for motion segmentation, named Event Challenging Motion (ECMotion), to validate the effectiveness of our event-based framework. Specifically, we collect data in real-world settings with challenging indoor scenarios using a SEEM1 event camera with a resolution of 262×320.

These scenarios consisted of low light, high dynamic range (HDR), and proper light conditions, including moving objects different from EV-IMO, to broadly validate our method's robustness. Our dataset includes 350 frames with 150 annotated with ground-truth, and the distribution of different scene categories is detailed in Fig. 7.

Distinct from the EV-IMO dataset, the ECMotion dataset contains a limited amount of samples mainly captured under challenging scenarios. As our model has considerable parameters, the dataset is too small to train or fine-tune the model. If we pre-train a model on the EV-IMO dataset and fine-tune on our divided ECMotion dataset, the model including the image-based model can work or perform better on the ECMotion dataset because of overfitting. Thus, we prefer to use this dataset only for testing. All the models are trained on the EV-IMO dataset and evaluated on our ECMotion dataset.

The evaluation results of different methods on ECMotion are illustrated in Table 5. Notably, Our event-based MSRNN model outperforms the others with an IoU improvement of 1.5% over SwiftNet. Notably, the image-based MSRNN achieves an IoU of 0 and almost fails in every prediction, indicating that this image-based method is difficult to adapt to challenging scenes. The performance of MSRNN$^-$ drops by 7.1% in IoU compared to the original one, indicating the effectiveness of the RNN architecture in utilizing long-range information.

The visualized results obtained from various methods are shown in Fig. 6. Compared with the image-based approach, we note that event-based methods show greater robustness in challenging conditions, which shows that event cameras are more effective in motion segmentation, especially under challenging conditions. Moreover, our MSRNN shows better segmentation performance than MSRNN$^-$, indicating that the RNN component is crucial for utilizing long-range temporal information for the prediction of current frame, resulting in higher accuracy. Our model also achieves better segmentation results than SODmodel, SwiftNet, Swin-UNet+, and Swin-UNet, as evidenced by IoU scores. P2T model performs well on the ECMotion dataset by the IoU metric. However, it learns to segment wrong objects in most of the scenes, as illustrated in Figure 6, while MSRNN shows more robustness due to its multi-scale recurrent architecture which can sustain temporal consistency.

Furthermore, we split our ECMotion to 3:2 for training and validation respectively and we fine-tune our MSRNN model on the ECMotion dataset. Table 6 illustrates the comparison of image-based and event-based frameworks. The results indicate that event-based MSRNN still performs better than image-based MSRNN after fine-tuning, demonstrating that events perform better than images in extreme conditions. The image-based method is limited to the poor quality of images because in extreme scenes the object becomes blurry.

V. CONCLUSION

In summary, we investigate the potential of events for motion segmentation through comprehensive experiments. Specifically, we have introduced MSRNN, a novel motion segmentation network with a multi-scale recurrent architecture that effectively fuses features of adjacent frames, achieving considerable improvement. In addition, we introduce a real-scene dataset, ECMotion, which contains several instances of challenging conditions. Our method notably outperforms other existing methods for motion segmentation, both on the EV-IMO and our ECMotion datasets. We believe that our work will inspire further research into the intrinsic properties of events, and we intend to investigate the applicability of events for other visual tasks under dynamic scenes.

REFERENCES

[1] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting anything that moves," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1493–1502.
[2] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, "MODNet: Moving object detection network with motion and appearance for autonomous driving," 2017, *arXiv:1709.04821*.
[3] P. Lichtsteiner, "64 × 64 event-driven logarithmic temporal derivative silicon retina," in *Proc. IEEE Workshop CCD AIS*, vol. 2, 2003, pp. 202–205.
[4] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 dB 3 µs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.

[5] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[6] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based optical flow using motion compensation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 711–714.

[7] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3867–3876.

[8] T. Stoffregen and L. Kleeman, "Simultaneous optical flow and segmentation (SOFAS) using dynamic vision sensor," 2018, *arXiv:1805.12326*.

[9] T. Stoffregen and L. Kleeman, "Event cameras, contrast maximization and reward functions: An analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12292–12300.

[10] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 989–997.

[11] M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza, "E-RAFT: Dense optical flow from event cameras," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 197–206.

[12] Y. Wu, F. Paredes-Vallés, and G. C. H. E. de Croon, "Rethinking event-based optical flow: Iterative deblurring as an alternative to correlation volumes," 2022, *arXiv:2211.13726*.

[13] L. Sun, C. Sakaridis, J. Liang, Q. Jiang, K. Yang, P. Sun, Y. Ye, K. Wang, and L. Van Gool, "Event-based fusion for motion deblurring with cross-modal attention," in *Proc. Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, 2022, pp. 412–428.

[14] W. Shang, D. Ren, D. Zou, J. S. Ren, P. Luo, and W. Zuo, "Bringing events into video deblurring with non-consecutively blurry frames," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4511–4520.

[15] C. Haoyu, T. Minggui, S. Boxin, W. Yizhou, and H. Tiejun, "Learning to deblur and generate high frame rate video with an event camera," 2020, *arXiv:2003.00847*.

[16] Z. Jiang, Y. Zhang, D. Zou, J. Ren, J. Lv, and Y. Liu, "Learning event-based motion deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3317–3326.

[17] S. Lin, J. Zhang, J. Pan, Z. Jiang, D. Zou, Y. Wang, J. Chen, and J. Ren, "Learning event-driven video deblurring and interpolation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 695–710.

[18] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai, "Bringing a blurry frame alive at high frame-rate with an event camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6813–6822.

[19] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3852–3861.

[20] L. Sun, C. Sakaridis, J. Liang, P. Sun, J. Cao, K. Zhang, Q. Jiang, K. Wang, and L. Van Gool, "Event-based frame interpolation with ad-hoc deblurring," 2023, *arXiv:2301.05191*.

[21] J. Chen, H. Shi, Y. Ye, K. Yang, L. Sun, and K. Wang, "Efficient human pose estimation via 3D event point cloud," 2022, *arXiv:2206.04511*.

[22] Y. Bao, L. Sun, Y. Ma, D. Gu, and K. Wang, "Improving fast auto-focus with event polarity," 2023, *arXiv:2303.08611*.

[23] P. Tokmakov, K. Alahari, and C. Schmid, "Learning motion patterns in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 531–539.

[24] G. Yang and D. Ramanan, "Learning to segment rigid motions from two frames," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1266–1275.

[25] S. Shen, Y. Cai, W. Wang, and S. Scherer, "DytanVO: Joint refinement of visual odometry and motion segmentation in dynamic environments," 2022, *arXiv:2209.08430*.

[26] E. Meunier, A. Badoual, and P. Bouthemy, "EM-driven unsupervised learning for efficient motion segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4462–4473, Apr. 2023.

[27] A. Mitrokhin, C. Ye, C. Fermüller, Y. Aloimonos, and T. Delbruck, "EV-IMO: Motion segmentation dataset and learning pipeline for event cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6105–6112.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Munich, Germany, 2015, pp. 234–241.

[29] T. Darrell and A. Pentland, "Robust estimation of a multi-layered motion representation," in *Proc. IEEE Workshop Vis. Motion*, Jan. 1991, pp. 173–174.

[30] M. Irani, B. Rousso, and S. Peleg, "Detecting and tracking multiple moving objects using temporal integration," in *Proc. Eur. Conf. Comput. Vis.*, 1992, pp. 282–287.

[31] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 625–638, Sep. 1994.

[32] A. G. Bors and I. Pitas, "Prediction and tracking of moving objects in image sequences," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1441–1445, Aug. 2000.

[33] R. Vidal and R. Hartley, "Motion segmentation with missing data using powerfactorization and GPCA," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun./Jul. 2004, p. 2.

[34] T. Brox, A. Bruhn, and J. Weickert, "Variational motion segmentation with level sets," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 471–483.

[35] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, Jun. 2014.

[36] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3899–3908.

[37] S. D. Jain, B. Xiong, and K. Grauman, "FusionSeg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2126.

[38] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "SfM-Net: Learning of structure and motion from video," 2017, *arXiv:1704.07804*.

[39] C. Xie, Y. Xiang, Z. Harchaoui, and D. Fox, "Object discovery in videos as foreground motion clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9986–9995.

[40] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1710–1720, Aug. 2015.

[41] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[42] Y. Zhou, G. Gallego, X. Lu, S. Liu, and S. Shen, "Event-based motion segmentation with spatio-temporal graph cuts," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 12, 2021, doi: 10.1109/TNNLS.2021.3124580.

[43] J. Chen, Y. Wang, Y. Cao, F. Wu, and Z. Zha, "ProgressiveMotionSeg: Mutually reinforced framework for event-based motion segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 303–311.

[44] A. Mitrokhin, Z. Hua, C. Fermüller, and Y. Aloimonos, "Learning visual motion segmentation using event surfaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14402–14411.

[45] M. Liu and T. Delbruck, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–12.

[46] G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12272–12281.

[47] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "HOTS: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.

[48] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3080–3089.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

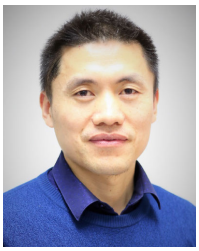
[50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[51] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

[52] M. Oršic, I. Krešo, P. Bevandic, and S. Šegvic, "In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12599–12608.
[53] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-Unet: Unet-like pure transformer for medical image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 205–218.
[54] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2T: Pyramid pooling transformer for scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 30, 2022, doi: 10.1109/TPAMI.2022.3202765.
[55] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

SHAOBO ZHANG received the B.S. degree in electronic science and technology from Tianjin University, in 2020. He is currently pursuing the degree with the State Key Laboratory of Modern Optical Instrumentation, Zhejiang University (ZJU). His research interests include optical sensors and computer vision, specifically, he does research on event-based motion segmentation.

LEI SUN received the B.S. degree in optical engineering from the Beijing Institute of Technology (BIT), in 2018. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Modern Optical Instrumentation, Zhejiang University (ZJU). He is a Visiting Ph.D. Student with the Computer Vision Laboratory, ETH Zürich. He has published more than ten refereed research articles. His research interests include semantic segmentation, event-based vision, and low-level vision. For more information please visit his website: ahupujr.github.io.



KAIWEI WANG (Member, IEEE) received the B.S. and Ph.D. degrees from Tsinghua University, in 2001 and 2005, respectively. In October 2005, he started his postdoctoral research with the Center of Precision Technologies (CPT), University of Huddersfield, funded by the Royal Society International Visiting Postdoctoral Fellowship and the British Engineering Physics Council. He joined Zhejiang University, in February 2009, and has been mainly researching on intelligent optical sensing technology and visual assisting technology for the visually impaired. He is currently a Full Professor with the State Key Laboratory of Modern Optical Instrumentation and the Deputy Director of the National Optical Instrument Engineering Research Center, Zhejiang University. Up to date, he owns 80 patents and has published more than 150 refereed research articles. For more information, visit his website: http://wangkaiwei.org/.