Memory-boosting RNN with dynamic graph for event-based action recognition*

CHEN Guanzhou, LIU Sheng, and XU Jingting**

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

(Received 20 February 2023; Revised 2 April 2023)

Existing action recognition methods based on event cameras have not fully exploited the advantages of event cameras, such as compressing event streams into frames for subsequent calculation, which greatly sacrifices the time information of event streams. Meanwhile, the conventional PointCloud-based methods suffer from large computational complexity while processing event data, which make it difficult to handle long-term actions. To tackle the above problems, we propose a dynamic graph memory-boosting recurrent neural network (DG-MBRNN). The proposed DG-MBRNN splits the event stream into sequential graph data for preserving structural information, then uses the recurrent neural network (RNN) with boosting spatiotemporal memory to handle long-term sequences of actions. In addition, the proposed method introduces a dynamic reorganization mechanism for the graph based on the distances of features, which can effectively increase the ability to extract local features. In order to cope with the situation that the existing datasets have too simple actions and too limited categories, we propose a new event-based dataset containing 36 complex actions. This dataset will greatly promote the development of event-based action recognition research. Experimental results show the effectiveness of the proposed method in completing the event-based action recognition task.

Document code: A **Article ID:** 1673-1905(2023)10-0629-6

DOI https://doi.org/10.1007/s11801-023-3028-7

Event-based action recognition has practical applications in scenarios where fast actions may occur, such as sign language recognition, human-computer interaction, and gaming. Event cameras process each pixel independently and output an event that contains the pixel coordinate and timestamp whenever the light intensity change by more than a threshold. However, individual events do not contain any motion information, and the events composing the actions are not output all at once. Therefore, extracting action features from the event stream is a challenging task.

As the mainstream methods for processing event streams, both frame-based and PointCloud-based methods have significant drawbacks. Frame-based methods are further divided into image-based and representation-based methods for event streams. The image-based methods[1-3] accumulate events within a time period on the same time plane to generate a sparse data image, and use RGB-based image recognition methods to recognize the generated image. Therefore, the time information is missed during the time period, resulting in motion blur in the event frames and the low latency feature of the event camera is not fully utilized. The representation-based methods[4,5] obtain several frames containing non-visual information from event streams and then perform the action recognition task by using the conventional RGB-based methods. Although these methods have achieved good performance, the conventional RGB-based methods are not designed for such non-visual information, so they cannot fully extract representation information for the actions. The PointCloud-based methods[6-10] take the timestamp and coordinate of events as a three-dimensional space structure. However, the PointCloud-based methods cannot handle long-time actions because of the large amounts of data.

Although recurrent neural networks (RNNs) can handle long-term event streams, they lack the ability to extract spatial features from event streams[11]. The input of RNNs is divided into event stream segments. Due to the low latency of event cameras, there is rich temporal information between event stream segments, which means that the attention needs to be paid to spatial feature. Although the PredRNN method[12] has increased the ability to extract spatial features from data by adding the spatiotemporal memory, redundant or invalid information may be generated as memory accumulates, which affects the expression of spatial information. Therefore, the current RNNs demonstrate good capacity to extract temporal features, but they often overlook the significance of spatial features.

* This work has been supported by the National Key R & D Program of China (No.2018YFB1305200).

** E-mail: edliu@zjut.edu.cn

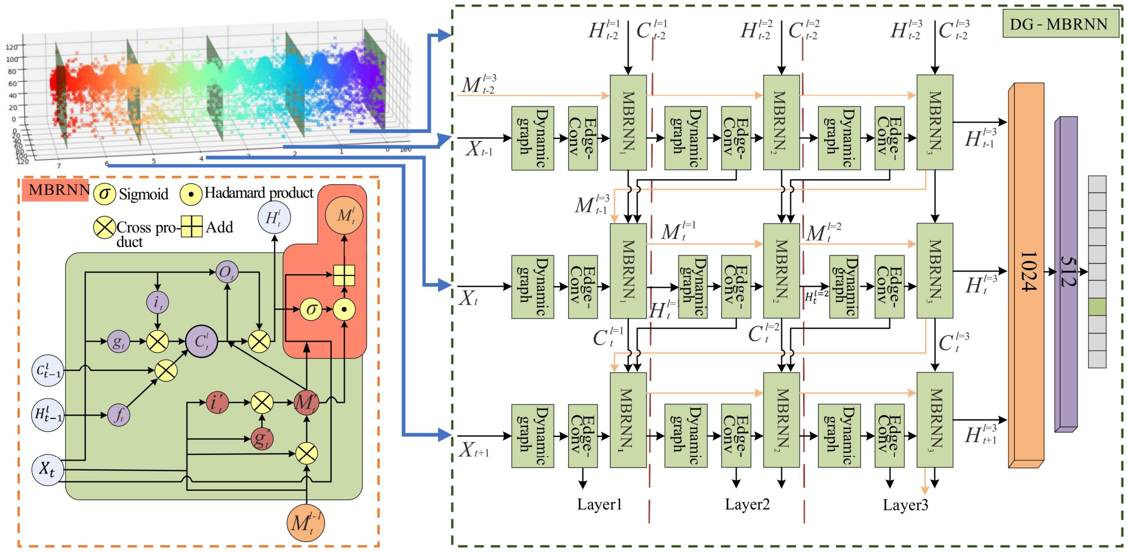


Fig.1 Framework of memory-boosting RNN with dynamic graph for event-based action recognition

In addition, the existing datasets also suffer from issues such as too simple actions and too limited categories. By simulating events from the RGB Human3.6m dataset, SCARPELLINI et al[13] released event-Human3.6m, an event-based human pose estimation dataset. However, due to the complexity and asynchrony of event streams, the simulated data generated by the algorithms has limitations. MIAO et al[14] used an event camera to capture action dataset of ten actions, such as arm waving, jumping, sitting, etc. AMIR et al[15] used an event camera to capture action dataset of eleven actions, such as waving, air guitar, clapping, etc. In these two real event datasets, the action type is relatively simple, and there are many repeated or continuous looping actions that may not fully represent real-world scenarios.

To overcome the aforementioned challenges, we propose a method called dynamic graph memory-boosting recurrent neural network (DG-MBRNN). This approach segments the event streams into multiple graph-structured data and feeds them into the RNN, which preserves temporal information while reducing model complexity. By making spatiotemporal information pass through all RNN cells, we use RNN with boosting spatiotemporal memory to improve the extraction of spatial features from the event stream. Our method includes a dynamic construction mechanism to rebuild the event graph based on feature distance in each RNN cell, which enriches the graph with more local information. Furthermore, we collect a complex action dataset of 36 non-cyclic actions with similar actions to evaluate the algorithm's effectiveness. We plan to release the dataset upon publication.

We propose a novel RNN architecture, DG-MBRNN, to address the challenges in event-based action recognition. Fig.1 shows the network architecture of DG-MBRNN. The dynamic graph, EdgeConv, and MBRNN together form a DG-MBRNN cell, which takes the features of the event streams, hidden information, and spatiotemporal information as inputs and outputs updated values. Notably, the spatiotemporal information is represented by the orange line and passes through all MBRNN cells. Finally, the event features in the graph output by each MBRNN cell are concatenated and passed as input to an MLP with softmax to obtain the final classification scores.

Event stream is a group of asynchronous events, where each event $e(x,y,t,p)$ presents a change in the intensity of light at position (x,y) at time t that exceeds a certain threshold value. p is a binary value polarity which indicates the pixel gets brighter or darker than before. The event stream containing n events can be represented as $X=\{e_1,e_2,...,e_n\}\subseteq\mathbb{R}^F$, where F is the feature dimension. Specifically, the initial event stream has F=4.

To extract spatiotemporal information between events, we use two-dimensional convolution for feature extraction. Therefore, we construct a graph $\mathcal{G}=(V,\varepsilon)$ by connecting K-nearest-neighbor events based on their feature distances, where $V=\{1,2,...,n\}$ represents the set of event indices and $\varepsilon\subseteq V\times V$ represents the set of edges. In each DG-MGRNN cell, the graph is updated by following steps.

First, edge features are updated by EdgeConv. The edge feature value between event e_i and e_j is defined as

$$\varepsilon_{ij}=e_i\oplus(e_j-e_i), \tag{1}$$

where $\oplus$ represents the concatenation. The edge feature includes both the event features and the feature differences, which endows $\mathcal{G}$ with the capability of expressing local features. Second, we update event features based on edge features as

$$e_i=\max_{j:(i,j)\in\varepsilon} h_\Theta(e_i,e_j), \tag{2}$$

where max means taking the maximum value. This creates a new graph $\mathcal{G}=(V,\varepsilon)$ with feature information, where its size is $(n,k,2*F)$. And the graph can be used to extract features using two-dimensional convolution.

In order to improve the ability of the network to extract spatial-temporal features from event streams, we introduce the spatiotemporal memory $\mathcal{M}$[12]. As indicated by the orange arrows in Fig.1, $\mathcal{M}$ involved the computation of all cells in the network and acquires information from them. However, as the iterations proceed, long-standing features in $\mathcal{M}$ become ineffective and redundant. To address this, the confidence information is extracted from X, H, and $\mathcal{M}$, and passed as input to Sigmoid function to obtain the weakening matrix $\boldsymbol{R}$:

$$\boldsymbol{R}=\sigma(\mathcal{W}_{\boldsymbol{R}}*[X,H,\mathcal{M}]), \quad (3)$$

where H is the hidden feature of the RNN, $\mathcal{W}$ represents the convolutional kernel, and * denotes the convolution operation. Compensation information is extracted from X, H, and $\mathcal{M}$ to build the boosting matrix $\boldsymbol{D}$:

$$\boldsymbol{D}=\mathcal{W}_{\boldsymbol{D}}*[X,H,\mathcal{M}], \quad (4)$$

where $\mathcal{M}$ is updated by

$$\mathcal{M}=\mathcal{M}\odot\boldsymbol{R}+\boldsymbol{D}, \quad (5)$$

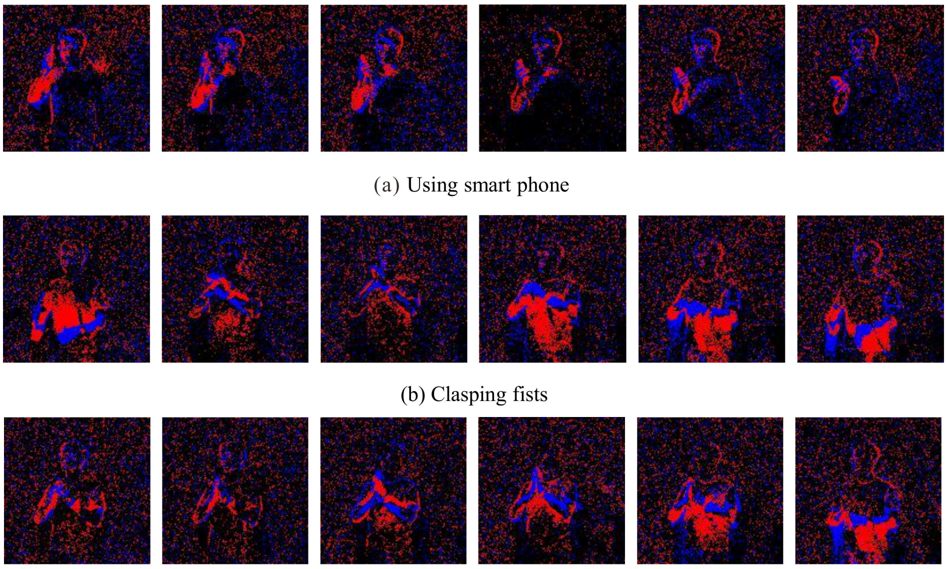
where $\odot$ is the symbol for Hadamard product. The values of the weakening matrix $\boldsymbol{R}$ are between 0 and 1, which can weaken the ineffective features in $\mathcal{M}$. To avoid affecting the effective features, the boosting matrix $\boldsymbol{D}$ is used to compensate for the weakened $\mathcal{M}$.

The gesture dataset[15] is captured using a DVS128 camera and consists of 1 342 event streams of 11 hand gestures performed by 29 participants in different lighting conditions, including natural light, fluorescent light, and LED light, with a resolution of 128×128.

In order to build a more challenging event-based complex action dataset, we capture the event streams with a CeleX5 camera with a resolution of 1 280×800. As shown in Tab.1, we design 36 non-cyclic actions. Particularly, these actions have similar movements. Additionally, to increase the difficulty of recognition, the actors are not required to complete the actions at the same speed. The dataset is recorded by 31 different actors of different age, height and gender. A total of 2 232 pieces of data are collected. For different actions and actors, each data has a variable duration spanning from approximately 1 s to 2 s. As shown in Fig.2, some similar actions are presented, and each image is obtained by accumulating events in 0.02 s. Specifically, the red point represents a decrease in light intensity, while blue represents an increase in light intensity.

Tab.1 Action list of our dataset

Heart-to-heart	Sneezing	Raising the hand	Gesture of OK	Gesture of like	Gesture of negative	Putting hands together	Using smart phone	Gesture of farewell
Left shoulder pain	Right shoulder pain	Clenching one fist	Silencing	Rejecting	Wiping the nose	Pointing up	Pointing down	Pointing left
Pointing right	Gesture of stop	Waving to come	Waving to leave	Spreading the hands	Applauding	Gesture of greet	Shaking the head	Nodding
Gesture of bye	Drinking	Standing up	Sitting down	Jumping	Back pain	Pacing back and forth	Neck pain	Toothache



(a) Using smart phone

(b) Clasping fists

(c) Putting hands together

Fig.2 Some similar actions in our dataset

Considering the computational complexity, we use the OctreeGrid[16] algorithm to downsample the dataset. The OctreeGrid filtering algorithm is applied to significantly reduce the number of events while preserving the spatio-temporal structure of the event streams.

After downsampling, the data is divided into segments, and a fixed number of events are sampled under distance constraints from each segment for training and testing. The performance comparison of different numbers of event stream segments and event numbers for one MBRNN in DG-MBRNN on gesture dataset is shown later. Through experiments, it was found that the best performance is achieved by inputting 4 event stream segments with 1 024 events each. As shown in Fig.1, each data segment is input into the recurrent network in time order.

In our proposed DG-MBRNN, the value of K in the K-nearest neighbor algorithm is set to 20. We use three layers of MBRNN with a 128-dimensional hidden layer. For the first layer, four segments of event stream were input in the time direction. The batch size for training is set to 8, and the learning rate is set to 0.01 for the first 100 epochs, and then reduced to 0.001 for the next 400 epochs. The experiments have been conducted on a Linux system equipped with two NVIDIA RTX 3090 graphics cards.

We use the accuracy of method on the test set as the evaluation criterion.

Tab.2 presents the results of ablation experiments conducted on the dynamic graph, which demonstrates that its inclusion effectively improves accuracy. Since the graph is reconstructed based on feature distances, events of the same type are clustered together, which leads to the graph containing more rich local features. Tab.3 shows the results of ablation experiments conducted on the spatiotemporal memory. The result demonstrates that using either the weakening or the boosting matrix independently results in decreased accuracy, while their simultaneous usage leads to increased accuracy. The possible reason for the situation described above might be that the weakening matrix weakens invalid features while also affecting valid features, and the boosting matrix enhances valid features while also affecting invalid features, which leads to invalid features are suppressed and valid features are balanced.

Increasing the number of edges in the graph can increase the features contained in the graph, but if the number of edges continues to increase, the two events at both ends of the edge will no longer have similar features, which will have a negative impact on the results. Tab.4 shows the effect of K value on the experimental results, using 4 event sequences with 1 024 events each. The results show that the accuracy increases first and then decreases with the increase of K value, and reaches the optimal value when K is 20, which confirms our hypothesis. Tab.5 shows the impact of the number of time-flow segments and the number of events per segment on the experimental results with K=20. "-" indicates that the experiment is not conducted due to hardware limitations. The results show a trend where the accuracy initially increases as the number of segments increases, but then decreases after reaching a certain point. Another trend is that as event number increases, the accuracy first increases rapidly and then grows slowly. Finally, the combination of 4 event sequences and 1 024 events per input achieves the best performance.

Tab.2 Performance comparison of DG-MBRNN on gesture dataset with and without dynamic graph

Method	Accuracy (%)
-	97.06
+ dynamic graph	**99.11**

Tab.3 Performance comparison of different spatio-temporal memory in DG-MBRNN on gesture dataset

Method	Accuracy (%)
-	98.81
+ the weakening matrix ***R***	98.39
+ the boosting matrix ***D***	97.83
+ both ***R*** and ***D***	**99.11**

Tab.4 Performance comparison of different *K* of KNN in DG-MBRNN on gesture dataset

K	Accuracy (%)
10	95.59
15	97.82
20	**99.11**
25	98.71

Tab.5 Performance comparison of different numbers of event stream segments and event numbers for one MBRNN in DG-MBRNN on gesture dataset

Number of event segments	Event number for one MBRNN (%)		
	512	1 024	2 048
1	89.43	91.97	96.28
2	91.72	95.92	97.53
3	93.12	97.97	98.29
4	96.68	**99.11**	-
5	95.43	98.84	-

In particular, the unpartitioned event streams are directly constructed as graph structures for training and testing, as shown in the first row of Tab.5. The experiments show that under the same total number of input event points, the network has similar recognition capabilities regardless of whether the event streams are split. For example, the accuracy of 96.28% is achieved by directly inputting 2 048 event points, while the accuracy of 95.92% is achieved by partitioning 2 048 into two graph data, and the accuracy of 96.68% is achieved by partitioning 2 048 into four graph data, indicating that splitting the event streams into sequential graph data can preserve most of the spatiotemporal information of the event streams. So, splitting the event stream into sequential graph data is an effective data processing method. Additionally, using the RNN with boosting spatiotemporal memory can effectively handle segmented graph structures, making it possible to process longer temporal actions without affecting recognition accuracy.

Tab.6 shows the comparison results between our proposed DG-MBRNN and state-of-the-art (SOTA) methods on the gesture dataset[15]. Except for the slightly lower performance compared to the temporal binary representation method based on the Inception3D network, DG-MBRNN outperforms other methods, including the temporal binary representation method based on the long short-term memory (LSTM) network. The Inception3D network has a complex structure and requires a large amount of computation, while DG-MBRNN achieves similar results with only three layers of RNN network. In addition, although the DG-MBRNN uses dynamic graph and EdgeConv, similar to the dynamic graph converlutional neural network (DGCNN), the DG-MBRNN gets better performance than DGCNN, which shows that using RNN with boosting spatiotemporal memory can effectively improve the accuracy of event-based action recognition. Overall, the above results demonstrate the effectiveness of our proposed DG-MBRNN.

Tab.6 Performance comparison with SOTA methods on gesture

Method	Type	Accuracy (%)
ST filter + CNN[1] (2019)	Image-based	94.85
Spatial-temporal images[2] (2020)	Image-based	97.4
ACE-BET[3] (2022)	Image-based	98.88
PointNet++[6] (2019)	PointCloud	94.1
PAT[7] (2019)	PointCloud	96
ST-EvNet[8] (2020)	PointCloud	97.27
DGCNN[9] (2020)	PointCloud	98.56
RG-CNN[10] (2019)	PointCloud	90.62
Temporal binary representation[5] (2021)	representation	**99.62 (Inception3D)**
Temporal binary representation[5] (2021)	representation	97.73 (LSTM)
Time-surfaces + KNN[4] (2020)	representation	97.2
DG-MBRNN (ours)	PointCloud-RNN	99.11

DG-MBRNN achieves an accuracy of 92.19% on our dataset, which is quite impressive considering the challenging nature of our proposed dataset that contains many similar and diverse actions, especially for low-resolution event stream data. For instance, Fig.2 shows the action of clasping fists and putting hands together. The performance of DG-MBRNN on our proposed dataset is only slightly lower than that on the gesture dataset, which fully demonstrates that DG-MBRNN can effectively handle the complex situations with similar and diverse actions.

Tab.7 shows the performance comparison between the PointCloud-based method DGCNN[9] and DG-MBRNN on our dataset. The accuracy gap between DGCNN and DG-MBRNN is greater on our dataset than on the gesture dataset, indicating that our dataset is more effective in verifying the algorithm's effectiveness and the DG-MBRNN is superior.

Tab.7 Performance comparison with DGCNN methods on our proposed dataset

Method	Accuracy (%)
DGCNN[9]	83.68
DG-MBRNN (ours)	92.19

In this paper, we propose a DG-MBRNN method for event-based action recognition tasks. DG-MBRNN uses a novel RNN module that corrects spatiotemporal memory by weakening and boosting matrices, highlighting long-term effective features and shielding invalid features. DG-MBRNN uses EdgeConv and dynamic graph to aggregate events with similar properties, which increases the local feature expression ability of event points. We have constructed a new event-based action recognition dataset that contains more actions and similar actions compared to existing datasets, which will greatly promote the development of event-based action recognition. Through experiments on the gesture and our datasets, the effectiveness of the DG-MBRNN method has been verified.

Ethics declarations

Conflicts of interest

The authors declare no conflict of interest.

References

[1] ROHAN G. Spatiotemporal filtering for event-based action recognition[EB/OL]. (2019-03-19) [2023-02-20]. https：//doi.org/10.48550/arXiv.1903.07067.

[2] HUANG C X. Event-based action recognition using timestamp image encoding network[EB/OL]. (2020-09-28) [2023-02-20]. https：//doi.org/10.48550/arXiv.2009.13049.

[3] LIU C, QI X, LAM E Y, et al. Fast classification and action recognition with event-based imaging[J]. IEEE access, 2022, 10：55638-55649.

[4] MARO J M, IENG S H, BENOSMAN R. Event-based gesture recognition with dynamic background suppression using smartphone computational capabilities[J]. Frontiers in neuroscience, 2020, 14：275.

[5] INNOCENTI S U, BECATTINI F, PERNICI F, et al. Temporal binary representation for event-based action recognition[C]//Proceedings of 25th International Conference on Pattern Recognition (ICPR), September

13-18, 2020, Milan, Italy. New York：IEEE, 2021：10426-10432.

[6] WANG Q, ZHANG Y, YUAN J, et al. Space-time event clouds for gesture recognition：from RGB cameras to event cameras[C]//Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), January 7-11, 2019, Waikoloa, HI, USA. New York：IEEE, 2019：1826-1835.

[7] YANG J, ZHANG Q, NI B, et al. Modeling point clouds with self-attention and gumbel subset sampling[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 15-20, 2019, Long Beach, USA. New York：IEEE, 2019：3323-3332.

[8] WANG Q, ZHANG Y, YUAN J, et al. ST-EVNet：hierarchical spatial and temporal feature learning on space-time event clouds[C]//Proceedings of Neural Information Processing Systems, December 6, 2020, Beijing, China. New York：IEEE, 2020.

[9] CHEN J, MENG J, WANG X, et al. Dynamic graph CNN for event-camera based gesture recognition[C]//2020 IEEE International Symposium on Circuits and Systems (ISCAS), May 17-20, 2020, Spain. New York：IEEE, 2020：1-5.

[10] BI Y, CHADHA A, ABBAS A, et al. Graph-based spatio-temporal feature learning for neuromorphic vision sensing[J]. IEEE transactions on image processing, 2020, 29：9084-9098.

[11] SONG Y, LIU G, WANG G, et al. SDN traffic prediction based on graph convolutional network[J]. Computer science, 2021, 48(6A)：392-397.

[12] WANG Y, LONG M, WANG J, et al. PredRNN：recurrent neural networks for predictive learning using spatiotemporal LSTMs[J]. Advances in neural information processing systems, 2017, 30.

[13] SCARPELLINI G, MORERIO P, DEL BUE A. Lifting monocular events to 3D human poses[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 20-25, 2021, Online. New York：IEEE, 2021：1358-1368.

[14] MIAO S, CHEN G, NING X, et al. Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection[J]. Frontiers in neurorobotics, 2019, 13：38.

[15] AMIR A, TABA B, BERG D, et al. A low power, fully event-based gesture recognition system[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, July 21-26, 2017, Hawaii, USA. New York：IEEE, 2017：7243-7252.

[16] LEE K H, WOO H, SUK T. Point data reduction using 3D grids[J]. The international journal of advanced manufacturing technology, 2001, 18：201-210.