Event Camera Denoising Using Asynchronous Spatio-Temporal Event Denoising Neural Network

Wanji Wu, Hexiong Yao, Chunlei Zhai, Zhiqiang Dai, Xiangwei Zhu

School of Electronic and Communication Engineering, Sun Yat-sen University, Shenzhen, China
(wuwj53, yaohx5, zhaichlei)@mail2.sysu.edu.cn, (daizhiqiang, zhuxw666)@mail.sysu.edu.cn

Keywords: Event Camera, Background Activity Noise, Denoising, Spatiotemporal Attention Embedding.

Abstract

Compared to conventional cameras, event cameras represent a noteworthy advancement in neuromorphic imaging technology, garnering considerable attention from researchers due to their distinct advantages. However, event cameras are susceptible to significant levels of measurement noise, which can detrimentally affect the performance of algorithms reliant on event stream for tasks such as perception and navigation. In this study, we introduce a novel method for denoising event stream, aiming to filter out events that do not accurately reflect genuine logarithmic intensity changes within the observed scene. Our approach focuses on the asynchronous nature and spatiotemporal properties of events, culminating in the development of a novel Asynchronous Spatio-Temporal Event Denoising neural Network(ASTEDNet). This network operates directly on event streams, circumventing the need to convert event stream into denser formats like image frames, thereby preserving their inherent asynchronous nature. Drawing upon principles from graph encoding and temporal convolutional networks, we incorporate spatiotemporal feature attention mechanisms to capture the temporal and spatial correlations between events. This enables the classification of each active event pixel in the original stream as either representing a genuine intensity change or noise. Comparative evaluations conducted on multiple datasets against state-of-the-art methods demonstrate the remarkable efficacy and robustness of our proposed algorithm in noise removal while retaining meaningful event information within the scene.

1. Introduction

Recent advancements in technology and algorithms, along with the increased compactness and affordability of information, have made Active Pixel Sensor (APS) imaging sensors, such as traditional cameras, essential for visual navigation and localization tasks, including visual simultaneous localization and mapping (SLAM). However, APS sensors face significant challenges when subjected to high-speed motion or extreme lighting conditions, such as those found in tunnels. These conditions can lead to issues like motion blur, overexposure, and underexposure, which adversely affect computer vision tasks.

The advent of event cameras offers a novel approach to tackling the aforementioned challenges. Event cameras are asynchronous sensors designed to mimic the biological structure of the human retina. Traditional cameras integrate all pixels over integration calculation cycles and readout cycles to form a single frame image. In contrast, event cameras detect logarithmic changes in intensity at each pixel. As depicted in Figure 1, When the change surpasses a predetermined threshold, an event is triggered, providing the corresponding pixel’s row/column index, timestamp, and polarity. This mechanism equips event cameras with several advantages, including low redundancy, low power consumption, high temporal resolution (microseconds) and high dynamic range (up to 120 dB)(Gallego et al., 2020). The innovative methods of visual data acquisition and processing by event cameras have prompted a paradigm shift in visual algorithms. Since their inception, event cameras have demonstrated significant potential in challenging computer vision tasks such as object detection(Li et al., 2022), depth estimation(Rebecq et al., 2018), visual navigation and localization(Vidal et al., 2018; Huang et al., 2023).

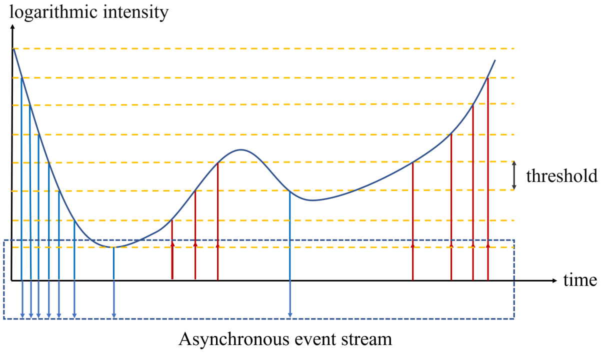


Figure 1. Event camera imaging principle. Events are generated asynchronously.

However, due to the immaturity of event camera hardware and its differential imaging mechanism, a significant amount of noise is generated, which severely hampers the performance of event cameras in subsequent visual tasks(Czech and Orchard, 2016). Under consistent illumination conditions, thermal noise and junction leakage current contribute to this noise. This type of noise is referred to as background activity noise(Guo and Delbruck, 2022), which is the primary focus of current research and also the central concern of this paper.

Given the distinctive data representation of event cameras, traditional frame-based denoising methods face challenges in direct application to event cameras. Mainstream denoising methods exploit the discrepancy between noise and the spatiotemporal correlation of genuine active events. Traditional event denoising methods typically assess spatiotemporal correlation by statistically analyzing event counts in the neighborhood or temporal disparities between events(Delbruck et al., 2008; Padala et al., 2018). However, these methods often require predefined

This contribution has been peer-reviewed.

threshold settings, significantly influencing denoising accuracy and causing fluctuations in performance across scenarios. Subsequently, a series of methods(Liu et al., 2015; Khodamoradi and Kastner, 2018; Guo and Delbruck, 2022; Wu et al., 2020; Guo and Delbruck, 2022) emerged that aim to reduce operational complexity by employing different event storage strategies. Nevertheless, these methods commonly overlook potential knowledge between neighborhoods, limiting their effectiveness when noise sharply increases. To address this limitation, deep neural networks(Duan et al., 2021b; Baldwin et al., 2020; Fang et al., 2022) have been introduced to fully explore latent spatiotemporal correlations and achieve better denoising results. However, there are certain issues present in current event denoising networks.: some methods require event streams to be converted into frames or other intermediate forms, leading to the loss of temporal information, while others solely focus on spatial features, neglecting temporal properties. Deep learning event denoising methods based on spatiotemporal correlation remain largely unexplored.

In this context, we introduce a novel event-driven denoising model named ASTEDNet, inspired by point cloud denoising theory and tailored to the temporal dynamics of event streams. ASTEDNet is capable of discerning spatiotemporal correlations between newly detected events and previously active ones within the same spatiotemporal vicinity. Our spatiotemporal feature embedding module draws insights from DGCNN(Wang et al., 2019b) and TCN(Bai et al., 2018). Embracing the principles of Graph Neural Network(GNN), we construct localized graphs and extract edge embeddings, facilitating the depiction of inter-point relationships. Subsequently, we employ temporal convolutional networks and introduce spatial and channel attention mechanisms(Woo et al., 2018) to further capture spatiotemporal features. Finally, genuine active events are identified through a binary classifier. Our approach handles events strictly in the order of their occurrence, eliminating the need for any intermediary conversions. Consequently, it effectively preserves both the asynchronous and sparse characteristics of event streams while fully leveraging their continuous temporal properties. Additionally, it facilitates the exploration of spatiotemporal correlations between newly arrived events and previously active ones within the same spatiotemporal neighborhood. The primary contributions of this paper are outlined as follows:

- We introduce a novel architecture for denoising event streams, which preserves the asynchronous and sparse characteristics of the data and supports end-to-end training.
- We develop dynamic graph-based feature encoding modules tailored to event stream and spatiotemporal feature extraction modules leveraging temporal convolutional networks and attention mechanisms. These modules effectively capture potential spatial correlations within the event stream.
- We assess the performance of our method on event datasets with diverse noise and illumination conditions, and analyze its excellence and robustness.

2. Related Works

2.1 Statistical-based Threshold Filter Methods

The earliest traditional method for event denoising involves a straightforward threshold filter, employing statistical principles to filter out anomalies by identifying low-density events. For example, BAF calculates the density of each event within its local spatiotemporal neighborhood and sets a threshold to reflect the spatiotemporal correlation between each event and its neighbors. The nearest neighbor filter(Liu et al., 2015) identifies events with fewer neighboring events in the surrounding pixels over a specific time frame as BA noise, thereby filtering out BA noise in the event stream. Building upon this approach, numerous more efficient and high-performance methods have emerged. For instance, KNoise(Khodamoradi and Kastner, 2018) achieved an $O(N)$ space complexity advantage by allocating two memory blocks to store the latest events in rows and columns. The dual-window filter (DWF)(Guo and Delbruck, 2022) further reduces memory usage by employing a first-in-first-out (FIFO) queue, storing only the most recent events and comparing them with new events to determine whether to insert the new event into the queue. To address memory and computational complexity concerns, Ynoise(Feng et al., 2020) proposes a density matrix where each incoming event is projected into its respective spatiotemporal region. By calculating the density of each incoming event within its spatiotemporal domain and prioritizing high-density events, event denoising is achieved. While the aforementioned filters prove effective in certain scenarios, their denoising accuracy heavily relies on threshold selection, requiring manual threshold adjustments when event density fluctuates, and they are susceptible to failure in high-noise scenarios. Furthermore, PUGM(Wu et al., 2020) utilize Iterated Conditional Modes (ICM) to minimize the energy function of the Probabilistic Undirected Graph Model (PUGM) for denoising events, but this denoising method is complex and computationally expensive, with a long run time.

2.2 Fitting-based Filter Methods

Some research investigates event denoising from alternative perspectives, employing fitting strategies for this purpose. The Time Surface (TS) approach(Lagorce et al., 2016) transforms the Dirac function of time into a logarithmic representation that monotonically decreases with time, facilitating the formation of a regular manifold known as the time surface, and subsequently removing events that disrupt surface smoothness. Another method, termed Inceptive Event Time Surfaces (IETS)(Baldwin et al., 2019) , recognizes that continuous events at individual pixels result from significant intensity changes in the scene, with initial events (IE) preceding subsequent tracking events (TE). IETS filters noise by extracting initial events corresponding to edge contours. EV-Gait(Wang et al., 2019a) adopts an optical flow fitting perspective, verifying motion consistency through velocity analysis and filtering events that disrupt the smoothness of optical flow surfaces to achieve denoising. The Guided Event Filter (GEF)(Duan et al., 2021a) , based on the linear optical flow assumption, combines gradients of Active Pixel Sensor (APS) frames, associating events with adjacent image frames through a motion model. It then extracts mutual structures between event frames and image gradients, removing mismatched events for denoising. While these fitting methods excel in handling individual moving objects, they may overlook many useful spatiotemporal correlations, leading to performance degradation in specific scenarios, such as low-light conditions or complex scenes.

2.3 Learning-based Methods

Moreover, some academic researchers have integrated neural networks to enhance denoising performance. Guo et al.(Guo and Delbruck, 2022) utilized a lightweight Multi-Layer Perceptron Denoising Filter (MLPF) to calculate the probability of noise occurrence for each event. Alkendi et al.(Alkendi et al., 2022) introduce a GNN combined with transformers, which classifies each active event pixel in the original stream as either a real intensity change or noise. Furthermore, recent advancements include Convolutional Neural Network (CNN) methods. EDnCNN(Baldwin et al., 2020), for instance, integrates APS and IMU data to compute event probabilities, which are then used as labels for training a binary classification network. This approach constructs multiple time surfaces for events and their neighboring events, encodes them, and inputs them into the network to identify noise events. Although AEDNet(Fang et al., 2022) leverages the classical PointNet from point cloud deep learning methods as its backbone to denoise events element-wise, it does not fully exploit the temporal properties of event streams at the network level. Additionally, EventZoom(Duan et al., 2021b) employs an efficient U-Net network as the backbone to perform event denoising and super-resolution in a noisy-to-noisy manner. However, it introduces regularization operations, sacrificing the advantage of the high temporal resolution of event stream and failing to fully utilize temporal continuity.

3. Methodology

Our objective is to systematically eliminate BA noise events from the asynchronous event stream. To achieve this, we begin by dissecting the underlying mechanism of noise generation in event cameras and the spatiotemporal characteristics of event stream. Subsequently, leveraging these insights, we provide a succinct overview of our proposed denoising methodology.

3.1 Problem Statement

We begin by elucidating how event cameras operate asynchronously, responding to individual pixels and generating event streams. The functionality of event cameras starkly contrasts with that of frame-based cameras, which operate at fixed frame rates. An event is represented as a tuple $(\mathbf{u}, t, p)$. Specifically, in a noise-free scenario, when the logarithmic intensity change L exceeds a constant threshold value C since the last event was triggered at pixel $\mathbf{u} = (x, y)$, an event $(\mathbf{u}, t, p)$ is triggered at pixel $\mathbf{u}$ and time t, as shown in formula (1).

$$p = \begin{cases} +1, & L(\mathbf{u}, t) - L(\mathbf{u}, t - \Delta t) \geq C \\ -1, & L(\mathbf{u}, t) - L(\mathbf{u}, t - \Delta t) \leq -C \end{cases} \tag{1}$$

Where $\mathbf{u} = (x, y)$ represents the pixel position, t denotes the timestamp, and $p \in \{-1, 1\}$ indicates the polarity, signifying the direction of brightness change (1 for increase and -1 for decrease). Δt denotes the time interval since the occurrence of the last event at pixel $\mathbf{u} = (x, y)$. Triggering multiple events (or event stream) can be expressed as:

$$E(x, y, t) = \{p_n \delta (x - x_n, y - y_n, t - t_n)\}_{n=1}^{N}, \tag{2}$$

where $\delta(\cdot)$ denotes the Dirac delta function. In summary, the distinctive data representation of event cameras presents challenges in directly applying frame-based denoising methods to event cameras.

Event denoising represents a fundamental classification task. Unlike tasks such as classification and segmentation, which involve abstracting scene content at a higher level, event denoising focuses on inferring signal features at the pixel level. The operational principle of event cameras suggests that genuine events generated by intensity gradients often correspond to scene or object edges, exhibiting continuous spatiotemporal characteristics. Conversely, random BA noise typically appears as isolated, unconnected points, lacking meaningful spatiotemporal correlation. In simpler terms, events clustered in both space and time are more likely to represent genuine signals, whereas isolated events are more likely to be attributed to noise. Therefore, event denoising can be achieved by learning from features that represent local time or space.

Generally, the approach to event denoising involves sampling neighboring events within the spatiotemporal vicinity of the target event according to a specific strategy to analyze its spatiotemporal properties and determine if it constitutes noise. Given a spatiotemporal neighborhood W, an asynchronous event $E = \{x_n, y_n, t_n \in W : n = 1, \ldots, N\}$ is derived from a pixel array. Unlike structured frames, the pixel array represents an intuitive set of discrete and sparse points in the spatiotemporal domain. Therefore, the asynchronous spatiotemporal event stream poses distinct challenges compared to traditional frame-based techniques. To integrate asynchronous event streams with deep learning methodologies, the transformation of time-series point sets into continuous measurements using kernel functions is essential(Gehrig et al., 2019). Formally, the event representation can be described as follows:

$$F(x, y, t) = \sum_{n=1}^{N} E(x_n, y_n, t_n) f(x - x_n, y - y_n, t - t_n), \tag{3}$$

Here, $f(x, y, t)$ represents the kernel function, which can be either a manually designed function or a neural network architecture. The kernel function is crucial for transforming the asynchronous event stream into a continuous measurement space or embedding.

As the asynchronous event stream consists of sparse points in the spatiotemporal domain rather than structured frames, we analyze event representation from the perspective of event-based signal processing. Given the temporal properties of event stream, the event stream can be seen as a time series with spatial position information. For each event, we form an event sequence by sampling its neighboring events and directly process these sequences using neural networks, without the need for frame conversion. This approach preserves the spatial discreteness and temporal continuity properties of the original events. Such representation enables end-to-end spatiotemporal modeling and maximizes the utilization of spatiotemporal cues in the event stream, thereby enhancing denoising task performance to the fullest extent possible.

3.2 Spatiotemporal Neighborhood Sampling

Our approach employs a per-event binary classification strategy, facilitating asynchronous processing to leverage the advantages of preserving high temporal resolution of events. Upon the arrival of a new event, we employ a straightforward Spatiotemporal Sampling technique to select neighboring events of the target event. Subsequently, through a neural network architecture, we perform spatiotemporal modeling on the selected neighboring events to extract features of the newly arrived event, thereby accomplishing the denoising classification task.

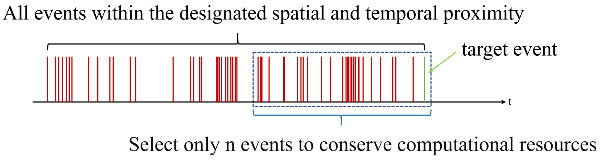

All events within the designated spatial and temporal proximity
target event
t
Select only n events to conserve computational resources

Figure 2. Spatiotemporal neighborhood sampling strategy. Green represents newly arrived events, and red represents previously arrived events filtered out according to formula (4), sorted by timestamp.

The sampling strategy is outlined as follows:

$$\begin{aligned} S(x,y,t) = \{E(x,y,t)||x - x_\text{i}\ | &\leq x_\text{band}\ , \\ |y - y_\text{i}| &\leq y_\text{band}\ , \\ t - t_\text{i} &\geq -t_\text{band}\ \} \end{aligned} \tag{4}$$

Where $E(x_i, y_i, t_i)$ represents the newly arrived event. x_{band} and y_{band} indicate spatial sampling of neighboring events within a rectangular region centered around the target event, while t_{band} specifies strict selection of events occurring before the arrival of the new event. We strictly select events with timestamps preceding the arrival of the new event to respect the temporal nature of the event stream, avoiding the disruption of the high temporal resolution characteristic of event cameras, thereby showcasing the asynchronous processing advantage of our method for event streams. Given the event camera's high temporal resolution, a large number of events are generated within a short period. We employ a K nearest neighbor (kNN) algorithm, similar to those commonly used in point cloud processing, to select only the n events closest in time to the current event in its neighborhood, sufficient for subsequent feature extraction and to prevent computational resource wastage, as depicted in Figure 2.

Moreover, if the number of events in the spatiotemporal neighborhood of the newly arrived event falls below the set threshold, as discussed in Section 3.1, there is a high likelihood of this event being deemed noise. In such cases, our Spatiotemporal Sampling acts similar to traditional event threshold filters, with t_{band} in equation (4) representing the parameters of the threshold filter. A higher t_{band} retains more events. To set tt_{band} reasonably, we utilize adaptive thresholds as proposed by Fang et al.(Fang et al., 2022):

$$t_\text{band} = \frac{t_\text{end} - t_\text{start}}{\text{int}\left(\frac{N}{K}\right)} \tag{5}$$

where N represents the event number in batch processing. Equation (9) assumes that an average of K events adequately describes a complete transient motion, and events generated within t_{band} are temporally correlated. The parameter K depends on the hardware configuration of the camera and the complexity of the scene.

It should be noted that in this study, we excluded the utilization of event polarity as node features. This decision was made due to the sensitivity of event polarity to changes in scene illumination, which can vary according to different camera parameters. Therefore, in practice, we only utilize the three attributes: x, y, and t.

3.3 Asynchronous Spatiotemporal Attention Embedding

As depicted in the Figure 3, our ASTAE integrates three components: DGEM leverages the concept of graph encoding for feature encoding of event sequences; TCAN models the spatiotemporal dependencies within event sequences. Subsequently, the data is passed to the ANP module for conventional neural network processing. Prior to feeding the data into our network framework, normalization is applied to the event values sampled by STNS after subtracting them from the target event.

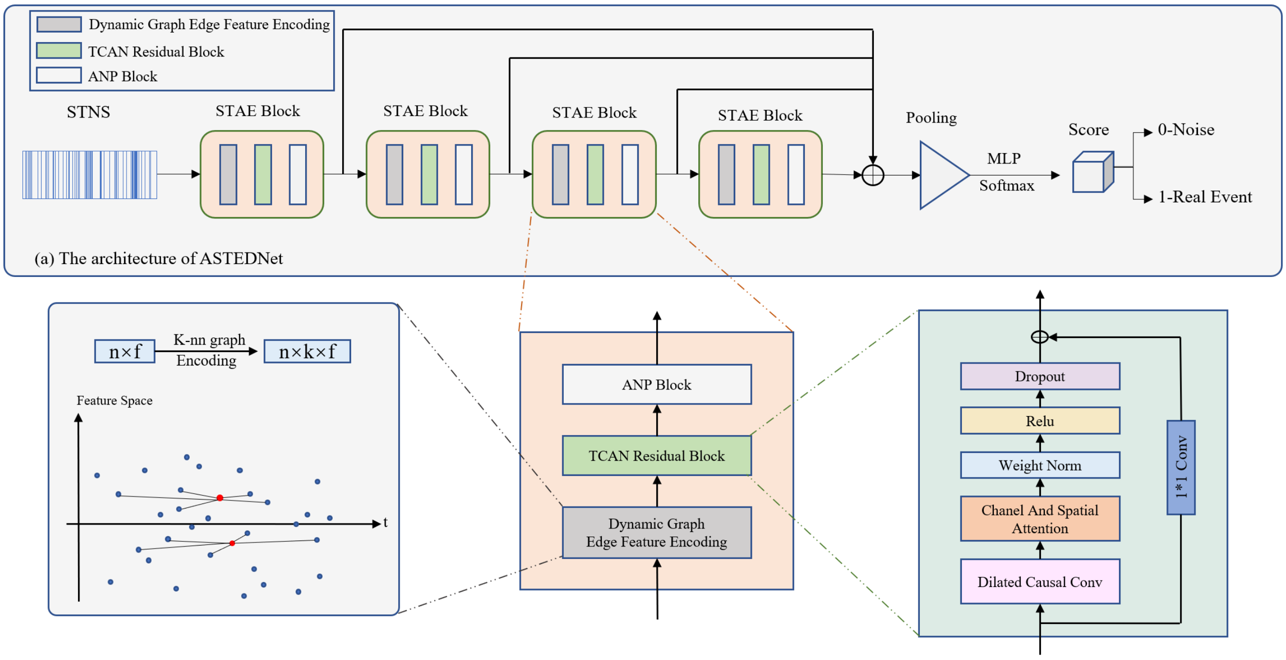
3.3.1 Dynamic Graph Encoding Module (DGEM)

Our module for embedding edge features in dynamic graphs is inspired by the structure of DGCNN(Wang et al., 2019b) for encoding event sequences. Illustrated in the Figure3(c), this module follows the principles of GNN. It computes pairwise distance matrices in the feature space and selects the k nearest events for each event. By constructing a local neighborhood graph and leveraging local feature structures on connecting edges between adjacent event pairs, it computes the differences between each event and its neighbors. These differences are then concatenated with the original events to form local features. This process facilitates feature encoding for each event in the spatiotemporal neighborhood, promoting spatiotemporal interplay among sampled neighborhood events and enhancing the information content of each event. The set of k nearest neighbors for an event varies across layers of the network and is computed based on the embedding sequence. Consequently, the local neighborhood graph is not static but dynamically updated after each layer of the network.

The proximity relationships in the feature space differ from those in the input, leading to non-local diffusion of information throughout the entire event neighborhood when this module is stacked. It's important to note that not only does this module influence the feature space, but subsequent TCAN modules and their attention layers also play a role in integrating the feature space. By incorporating spatiotemporal neighborhood information for each event, this dynamic graph edge feature embedding module, when applied in multi-layer network architectures, enhances the network's ability to learn the global properties of the entire event spatiotemporal neighborhood.

3.3.2 Time Convolution Attention Network (TCAN)

In some studies, events are initially transformed into two-dimensional or three-dimensional representations such as event images or event voxels before processing. However, this preprocessing step restricts the utilization of the temporal aspect of events. In essence, these approaches struggle to effectively leverage the temporal properties of asynchronous events. In contrast, as described in Section 3.1, we represent event neighborhoods as time series. Building upon this representation, we introduce a Temporal Convolution Attention Module to effectively explore the spatiotemporal information embedded in continuous event streams. Our TCAN improves upon TCN by modeling the spatiotemporal characteristics of event feature sequences encoded by DGEM. Additionally, it integrates channel attention and spatial attention CBAM(Woo et al., 2018) to focus on which parts and features of the event sequence are relevant for the current denoising task. TCAN combines dilated causal convolutions, channel attention, spatial attention, weight normalization layers, activation functions, dropout layers, etc., to form a residual structure(He et al., 2016). Following each



(a) The architecture of ASTEDNet

(c) Dynamic Graph Edge Feature Encoding Module

(b) STAE Block

(d) TCAN Residual Block

Figure 3. Architecture of ASTEDNet. (a)The architecture of ASTEDNet. Our architecture directly processes events sampled by STNS. It is stacked by (b) Spatio-Temporal Attention Embedding (STAE). STAE consists of (c)Dynamic Graph edge feature Encoding Module (DGEM) (d)Time Convolution Attention Network (TCAN) and ANP Block.

TCAN Residual Block is an ANP (ActivationNormalization-Pooling) layer, which reduces feature dimensions and restores the data shape to match the original input for subsequent connection with DGEM.

Stacking multiple TCAN Residual Blocks with DGEM facilitates the diffusion of information across the entire event neighborhood, enabling TCAN and DGEM to collaboratively extract spatiotemporal information. Subsequently, binary classification is performed using fully connected layers. Since our approach retains the original event stream format, subsequent tasks can leverage time series processing techniques, such as Spiking Neural Network(SNN) and Recurrent Neural Network(RNN), for continuous advanced tasks.

4. Experiments

4.1 Evaluation Metric

An effective event denoiser should preserve the majority of signals while effectively removing most noise events occurring outside the tracker after denoising. Moreover, it should demonstrate consistent performance across various scenarios. To quantitatively assess the performance of our proposed denoising model and compare it with the latest models on training and testing datasets, we first establish the following terms and metrics for calculation based on both raw data and denoised data.

TP represents the number of correctly predicted true positive active events, TN represents the number of correctly predicted true negative noise events, FP represents the number of noise events incorrectly predicted as true positive active events, and FN represents the number of true positive active events incorrectly predicted as noise. Our method retains events predicted as true positive active events (TP and FP) and removes events predicted as noise (TN and FN). The following metrics are defined based on the data before and after denoising.

Remaining Signal Ratio. The percentage of Remaining Signal after denoising, denoted as RSR, is defined as the ratio of the number of true positive active events after denoising to the original number of true positive active events before denoising,

$$RSR = \frac{TP}{TP + FN} \times 100 \tag{6}$$

Remaining Noise Ratio. The percentage of Remaining Noise after denoising, denoted as RNRf, is defined as the ratio of the number of noise events after denoising to the number of noise events before denoising,

$$RNR = \frac{FP}{TN + FP} \times 100 \tag{7}$$

Signal to Noise Ratio. Signal to Noise Ratio before denoising:

$$SNR = 10 \times \log_{10}\left(\frac{TP + FN}{TN + FP}\right) \text{ in } dB \tag{8}$$

Signal to Noise Ratio after denoising:

$$SNR = 10 \times \log_{10}\left(\frac{TP}{FP}\right) \text{ in } dB \tag{9}$$

RSR reflects the ability of a denoising model to retain true active events, while RNR represents the model’s ability to remove noise. Meanwhile, SNR comprehensively reflects the relative situation of signals and noise in the scene. A good denoising model needs to preserve true active events to the maximum extent while removing noise events as much as possible. In fact, a balance needs to be struck among SNR (in dB), RSRf, and RNR. Good denoising models often exhibit high RSR and SNR values and low RNR values.

4.2 Results On DVSCLEAN

Our approach requires supervised learning on datasets with accompanying labels. DVSCLEAN(Fang et al., 2022) is an event denoising dataset composed of simulated and real-world data. Its simulated dataset consists of events generated by the ESIM algorithm from provided image or video datasets, combined with artificially added noise, thus providing labels that can be used to train our model. The simulated dataset comprises 49 scenes, each containing data with noise event proportions of 50% and 100%. We select 10 scenes from these as a validation set, while the remaining scenes are used to train our model. The real-world dataset of DVSCLEAN includes three levels of scene complexity: simple indoor scenes, complex indoor scenes, and complex outdoor scenes. It consists of event streams and frame-based image data recorded by Celex-V cameras and traditional cameras with a resolution of 1280 × 800. We validate the effectiveness of our model on both the simulated and real-world datasets.

We conducted a comparative analysis of our method with TS(Lagorce et al., 2016), DWF(Guo and Delbruck, 2022), Event Denoising Convolutional Neural Network (EDnCNN(Baldwin et al., 2020)), and Asynchronous Event Denoising Neural Network (AEDNet(Fang et al., 2022)), all evaluated on the same dataset. The denoising performance of these five algorithms is summarized in Table 1. TS exhibits a high RNR score, indicating its limited ability to remove noise despite retaining most true active events. Additionally, TS demonstrates a relatively high RSR score, yet its SNR value is suboptimal. Conversely, DWF shows a low RSR value, implying a higher removal rate of true active events. Among the learning-based methods, three exhibit comparable RSR values. Notably, our ASTEDNet achieves the highest SNR score and the lowest RNR value across both 50% and 100% noise ratio scenarios, indicating superior performance.

Additionally, we conducted visualizations of the denoised event streams from the DVSCLEAN dataset, presenting them cumulatively in the form of two-dimensional images, as depicted in Figures 4 and 6. Due to the simplicity of the scenes and the high resolution of 1280 × 800 in the simulated dataset, noise that is close to real events in the denoised event stream may not be readily apparent. Thus, we also included timestamps for 3D visualization in Figure 5. Figures 4 and 5 reveal that, irrespective of the noise ratio scenarios, TS retains a substantial amount of noise, while DWF loses many true events, indicating suboptimal performance for both traditional filtering methods. The three learning-based methods exhibit commendable performance. However, as depicted in Figure 5, EDNCNN retains more noise in high noise ratio scenarios, and AEDNet shows some isolated noise in such scenarios. Overall, as the noise ratio increases, the denoising performance of EDNCNN and AEDNet deteriorates. In contrast, our ASTEDNet not only preserves the genuine event structure but also eliminates almost all noise in both low and high noise ratio scenarios, demonstrating exceptional robustness.

50% **noise ratio**			
Denoising Algorithms	**SNR**	**RSR**	**RNR**
Raw	3	＼	＼
TS	9.321	97.02	22.69
DWF	22.29	40.13	0.4737
EDNCNN	17.97	**97.97**	3.128
AEDNET	26.27	97.42	0.4595
ASTEDNet(ours)	**28.24**	96.13	**0.2884**
100% **noise ratio**			
Denoising Algorithms	**SNR**	**RSR**	**RNR**
Raw	0	＼	＼
TS	4.109	**97.14**	37.72
DWF	17.92	37.62	0.6075
EDNCNN	15.38	97.05	2.814
AEDNET	24.89	95.69	0.3103
ASTEDNet(ours)	**27.68**	94.18	**0.1607**

Table 1. Performance comparison on simulated data of DVSCLEAN

Goodlight-750lux			
Denoising Algorithms	**SNR**	**RSR**	**RNR**
Raw	19.71	＼	＼
TS	22.87	92.07	44.49
DWF	21.45	28.35	19.01
EDNCNN	19.79	**99.93**	98.20
AEDNET	30.65	93.51	7.518
ASTEDNet(ours)	**31.49**	96.82	**6.43**
Lowlight-5ux			
Denoising Algorithms	**SNR**	**RSR**	**RNR**
Raw	9.70	＼	＼
TS	15.30	82.79	35.61
DWF	14.97	25.22	11.68
EDNCNN	13.99	**95.12**	55.22
AEDNET	22.00	80.02	7.341
ASTEDNet(ours)	**23.94**	80.00	**4.699**

Table 2. Performance comparison on ED-KoGTL

Figure 6 displays the visualization effect of the DVSCLEAN real-world dataset, encompassing both indoor and outdoor scenes. TS and DWF exhibit significant performance variations across different scenes due to their sensitivity to parameter settings. EDNCNN effectively preserves object edge contours in indoor scenes but eliminates many meaningful features in outdoor scenes. Both AEDNet and our method demonstrate strong performance in both indoor and outdoor scenes, with our method outperforming AEDNet in filtering out isolated noise.

4.3 Results On ED-KoGTL

The ED-KoGTL(Alkendi et al., 2022) dataset is recorded using a DAVIS346C camera mounted on a 6-DOF robotic arm, with ground truth labels obtained from known object ground truth data (KoGTL). These labels utilize the Canny edge detector to extract edge information from APS and designate detected edge events as true events. We conducted algorithm testing under two common lighting conditions: good lighting (750 lux) and low lighting (5 lux). In comparison to the Goodlight 750 lux scene, the low light 5 lux scene exhibits more noise due to the tendency of event-based cameras to produce increased noise in dim lighting. Denoising results are depicted in Figure 7 and Table 2. Our method achieves a highest SNR value and the lowest RNR value. While EDNCNN attains the maximum RSR value, its RNR value is excessively high, rendering it insensitive to noise near real edges and thereby retaining more noise. Notably, from the visualization diagrams under low light conditions, our method's superiority over other methods is evident.

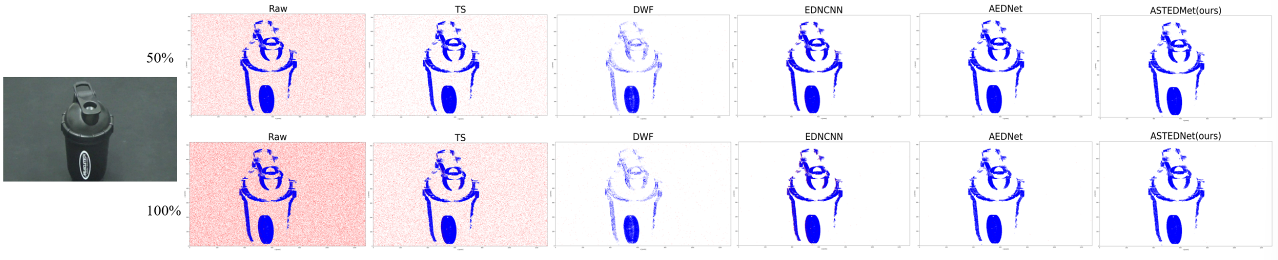


Figure 4. 2D visualization of denoising results of five algorithms on simulated data of DVSCLEAN.

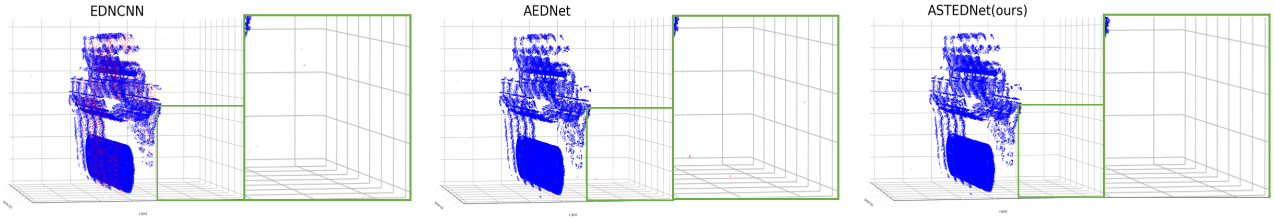


Figure 5. 3D visualization of denoising results of three learning-based methods on simulated data of DVSCLEAN.

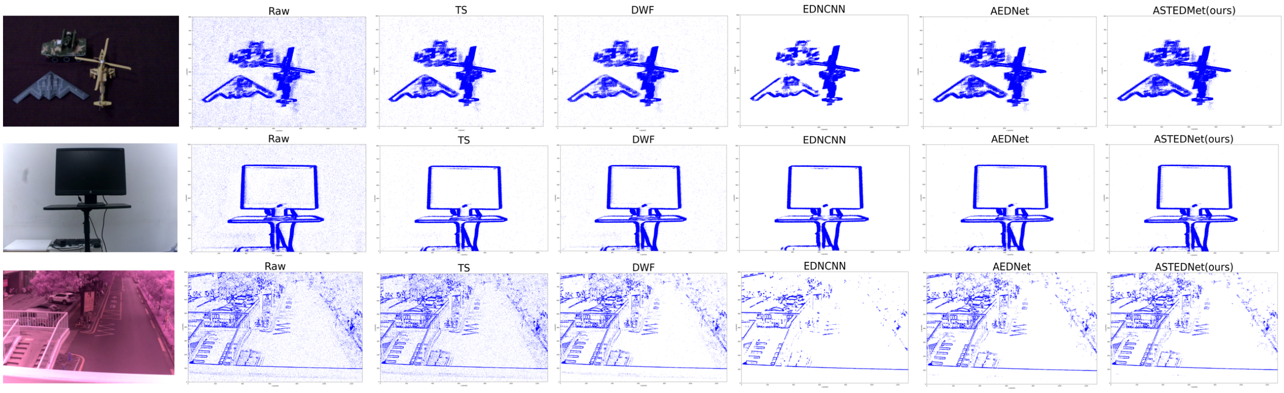


Figure 6. 2D visualization of denoising results of five algorithms on real world data of DVSCLEAN dataset.

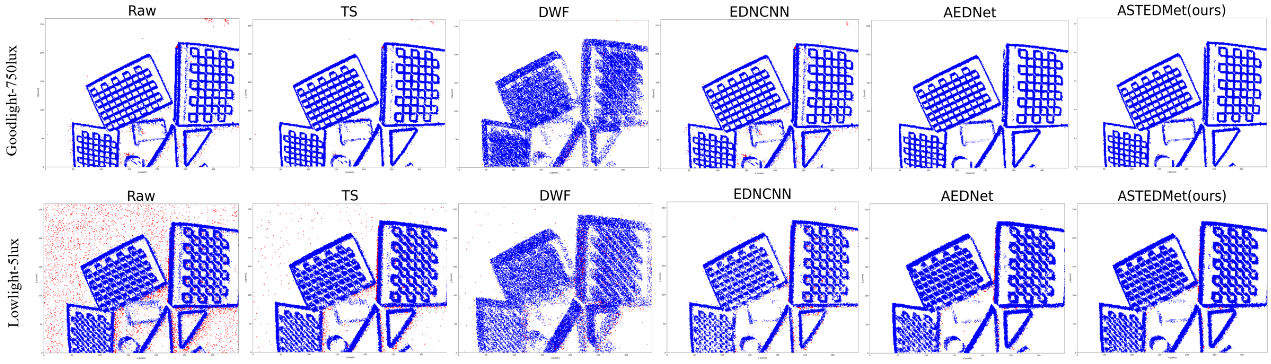


Figure 7. 2D visualization of denoising results of five algorithms on ED-KoGTL.

5. Conclusion

In this study, we introduce an innovative event-driven deep learning approach for event denoising. Our method effectively preserves the spatiotemporal and asynchronous characteristics inherent in original event stream. Notably, we handle events in the form of event sequences and devise spatiotemporal feature embedding units specifically tailored to the distinct data format and attributes of events. Through dynamic learning at the neural network level, we discern the spatiotemporal features of events, accurately distinguishing noise from genuine activity within the original event stream. Comparative analysis with state-of-the-art solutions across diverse datasets demonstrates the superior denoising capability of our ASTEDNet, adept at retaining meaningful events while suppressing noise effectively. We anticipate that this research will unlock the latent potential of event cameras in pivotal visual tasks such as navigation and localization.

References

Alkendi, Y., Azzam, R., Ayyad, A., Javed, S., Seneviratne, L., Zweiri, Y., 2022. Neuromorphic camera denoising using graph neural network-driven transformers. *IEEE Transactions on Neural Networks and Learning Systems*.

Bai, S., Kolter, J. Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Baldwin, R., Almatrafi, M., Asari, V., Hirakawa, K., 2020. Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1701–1710.

Baldwin, R. W., Almatrafi, M., Kaufman, J. R., Asari, V., Hirakawa, K., 2019. Inceptive event time-surfaces for object classification using neuromorphic cameras. *Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part II 16*, Springer, 395–403.

Czech, D., Orchard, G., 2016. Evaluating noise filtering for event-based asynchronous change detection image sensors. *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, IEEE, 19–24.

Delbruck, T. et al., 2008. Frame-free dynamic digital vision. *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, 1, Citeseer, 21–26.

Duan, P., Wang, Z. W., Shi, B., Cossairt, O., Huang, T., Katsaggelos, A. K., 2021a. Guided event filtering: Synergy between intensity images and neuromorphic events for high performance imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11), 8261–8275.

Duan, P., Wang, Z. W., Zhou, X., Ma, Y., Shi, B., 2021b. Eventzoom: Learning to denoise and super resolve neuromorphic events. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12824–12833.

Fang, H., Wu, J., Li, L., Hou, J., Dong, W., Shi, G., 2022. Aednet: Asynchronous event denoising with spatial-temporal correlation among irregular data. *Proceedings of the 30th ACM International Conference on Multimedia*, 1427–1435.

Feng, Y., Lv, H., Liu, H., Zhang, Y., Xiao, Y., Han, C., 2020. Event density based denoising method for dynamic vision sensor. *Applied Sciences*, 10(6), 2024.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K. et al., 2020. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1), 154–180.

Gehrig, D., Loquercio, A., Derpanis, K. G., Scaramuzza, D., 2019. End-to-end learning of representations for asynchronous event-based data. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5633–5643.

Guo, S., Delbruck, T., 2022. Low cost and latency event camera background activity denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1), 785–795.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, K., Zhang, S., Zhang, J., Tao, D., 2023. Event-based simultaneous localization and mapping: A comprehensive survey. *arXiv preprint arXiv:2304.09793*.

Khodamoradi, A., Kastner, R., 2018. $O(N)$ O (N)-Space Spatiotemporal Filter for Reducing Noise in Neuromorphic Vision Sensors. *IEEE Transactions on Emerging Topics in Computing*, 9(1), 15–23.

Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., Benosman, R. B., 2016. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7), 1346–1359.

Li, J., Li, J., Zhu, L., Xiang, X., Huang, T., Tian, Y., 2022. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Transactions on Image Processing*, 31, 2975–2987.

Liu, H., Brandli, C., Li, C., Liu, S.-C., Delbruck, T., 2015. Design of a spatiotemporal correlation filter for event-based sensors. *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 722–725.

Padala, V., Basu, A., Orchard, G., 2018. A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth. *Frontiers in neuroscience*, 12, 328064.

Rebecq, H., Gallego, G., Mueggler, E., Scaramuzza, D., 2018. EMVS: Event-based multi-view stereo3D reconstruction with an event camera in real-time. *International Journal of Computer Vision*, 126(12), 1394–1414.

Vidal, A. R., Rebecq, H., Horstschaefer, T., Scaramuzza, D., 2018. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2), 994–1001.

Wang, Y., Du, B., Shen, Y., Wu, K., Zhao, G., Sun, J., Wen, H., 2019a. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6358–6367.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., Solomon, J. M., 2019b. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5), 1–12.

Woo, S., Park, J., Lee, J.-Y., Kweon, I. S., 2018. Cbam: Convolutional block attention module. *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

Wu, J., Ma, C., Li, L., Dong, W., Shi, G., 2020. Probabilistic undirected graph based denoising method for dynamic vision sensor. *IEEE Transactions on Multimedia*, 23, 1148–1159.