Sparse-E2VID: A Sparse Convolutional Model for Event-Based Video Reconstruction Trained with Real Event Noise

Pablo Rodrigo Gantier Cadena[1] Yeqiang Qian[*2] Chunxiang Wang[1] Ming Yang[1]
[1]Department of Automation, Shanghai Jiao Tong University
[2]University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University
{Rodrigo.Gantier.C, qianyeqiang, wangcx, mingyang}@sjtu.edu.cn

Abstract

Event cameras are image sensors inspired by biology and offer several advantages over traditional frame-based cameras. However, most algorithms for reconstructing images from event camera data do not exploit the sparsity of events, resulting in inefficient zero-filled data. Given that event cameras typically have a sparse index of 90% or higher, this is particularly wasteful. In this work, we propose a sparse model, Sparse-E2VID, that efficiently reconstructs event-based images, reducing inference time by 30%. Our model takes advantage of the sparsity of event data, making it more computationally efficient, and scales better at higher resolutions. Additionally, by using data augmentation and real noise from an event camera, our model reconstructs nearly noise-free images. In summary, our proposed model efficiently and accurately reconstructs images from event camera data by exploiting the sparsity of events. This has the potential to greatly improve the performance of event-based applications, particularly at higher resolutions. Some results can be seen in the following video: https://youtu.be/sFH9zp6kuWE, [1].

1. Introduction

Event cameras are image sensors that offer several advantages over frame-based cameras [17], such as a 120dB dynamic range, high temporal resolution, and sparsity, among others [11, 28]. Although ConvRNN-based models for event-based image reconstruction preserve most of these features, they do not take full advantage of the sparse and asynchronous properties of event data. Event cameras have a sparse index of 90% or more in the spatial dimensions $u = (x, y)$ [11]. To perform event-based image reconstruction, these models convert the event data into a dense format using voxels. This involves filling each empty space with zeros. As a result, these models mainly process zeros during image reconstruction. Generally, the larger the model (more parameters), the better the quality of the reconstructed images. However, long inference times and excessive memory usage limit the use of such models in mobile applications.

Models such as ET-Net [33] E2VID+ and FireNet+ [30] are considered state-of-the-art (SOTA) for event-based image reconstruction and are capable of producing high-quality images. FireNet+ [30] and the original FireNet [27] share the same architecture, this is also the case with E2VID+ [30] and E2VID [24], the difference is in the training method and data. The difference between FireNet and E2VID is mainly in the inference time. E2VID has an inference time of 360 ms and FireNet has an inference time of 93 ms, all this at a resolution of 720×1280. However, these models process all data in dense format.

To address the issue of dense processing, this paper introduces Sparse-E2VID, an architecture that processes data in sparse format (COO [3]). With Sparse-E2VID, the inference time is reduced to 55 ms (at 720×1280 resolution), which is 30% faster than FireNet [30]. Additionally, Sparse-E2VID reduces the computational cost by 98% compared to FireNet+ [30], while also improving image quality.

Our Sparse-E2VID model is a Recurrent Convolutional model (ConvRNN) like FireNet, but Sparse-E2VID uses sparse convolution [6]. However, it is not as simple as replacing all convolution modules with sparse convolution modules. Recursive models for video reconstruction use the hidden state (as a memory tensor), which accumulates past information (events). Although the hidden state tensor is initialized with zeros, it becomes quite dense after accumulating data.

Sparse convolution is not as efficient as traditional convolution. Still, if the data has a sparse index greater than 80%, it can reduce the inference time and memory usage compared to traditional convolution. In the FireNet and E2VID architecture, ConvRNN modules are at the begin-

[1] Link for Bilibili https://www.bilibili.com/video/BV1dT411W7QQ/?spm_id_from=333.999.0.0&vd_source=cc66e37794f0bcc99d29fb7152a03960

ning and end of the models, but their positioning is not strict, so they can be anywhere in the model. Another aspect to consider is that the required output is an image in dense format. This means that Sparse-E2VID converts the data from sparse to dense format at some point.

Given the concept of the hidden state, the minimum sparsity index of 80%, the possibility of positioning the ConvRNN module anywhere, and the need to provide a final result in dense format, we redesigned the FireNet architecture to use sparse convolution. In Sparse-E2VID, we positioned the ConvRNN module at the end of the model. The first modules are sparse convolutional modules, and the ConvRNN module processes the hidden state tensor at the end of the model with dense convolution to generate an image in dense format. This is a simple but effective solution.

One aspect to note about Sparse-E2VID is that it predicts the gradient of the image. Recent research has shown that it is simpler for neuronal models to predict image gradients [9], which can reduce the number of parameters in a model or result in better image quality with a relatively small model. However, numerical integration of the result is still necessary. In previous work [9], an inverse matrix was used for this purpose, but the computational cost is extremely high and increases exponentially with resolution. In our work, instead we use the Fast Fourier Transform (FFT) to perform the integration, which scales better at higher resolutions and has a time of only 5 ms.

To further improve the quality of reconstructed images and reduce noise, we add real noise from an event camera to the training set. Although E2VID+ and FireNet+ [30] also add noise to their training sets, their noise is synthetic. Therefore, E2VID+ and FireNet+ still generate images with noise, especially in night scenes or at higher resolutions (e.g., 1 megapixel cameras).

In summary, our contributions are two fold:

- Firstly, we propose a sparse architecture that, in conjunction with FFT, reduces inference time by 30% an the computational cost.
- Secondly, we successfully reconstructed images with almost noise-free, thanks to the inclusion of real noise from an event camera and data augmentation.

2. Related Works

2.1. Early Methods

Since the first event cameras were commercialized [17, 28], researchers have been investigating the reconstruction of event-based images. The earliest methods for event-based image reconstruction involved integrating the obtained signal to produce a natural image [2, 7]. One of the first methods used motion prediction (tracking and mapping) to estimate the gradient of the image at a pixel-wise level. This gradient could then be numerically integrated [16]. While this work results in good quality reconstruction of event-based images, it also suffers from gross failures at times. This is due to the motion prediction algorithm, which is not only difficult to obtain but also error-prone.

2.2. Deep learning methods

A direct integration algorithm was proposed to avoid the reliance on motion prediction [26]. However, event cameras are not ideal sensors and hence, the resulting images are often noisy. Nevertheless, with the use of ConvRNN made it possible to achieve high-quality event-based video reconstruction without any constraints or motion prediction [23]. Several studies have built on this approach, resulting in improved video quality [4, 5, 24, 30, 33] and one reduced the inference time [27]. While these models are trained in a supervised manner, it is practically impossible to generate frame-based images that are well-synchronized and free of image blur, underexposure, or overexposure. Thus, event simulators are used to create training sets [15,22]. However, a recent work [21] used the generative event mathematical model to train in a self-supervised manner the FireNet and E2VID models. Although the reconstruction quality is not better than the SOTA, it is still a great advantage to be able to train a model without the need for ground truth data.

2.3. Sparse methods

Although one of the most important features of event cameras is their sparsity, it has not been extensively utilized in event-based video reconstruction. This is because CPUs and GPUs are optimized for processing dense data, and the end product of event-based video reconstruction is typically a dense image. However, some works have explored the use of sparse event data in conjunction with Graph Convolutional Networks (GCN) [8, 12, 25] and sparse convolution [20] for object classification and detection tasks. Despite this, there has been little investigation into event-based video reconstruction. In this work, we leverage the sparsity of event cameras to reduce inference time and computational cost for event-based video reconstruction. Our model, Sparse-E2VID, is 30% faster and computational cheaper compared to the FireNet architecture.

3. Method

3.1. Architecture

As mentioned earlier, the FireNet architecture is well-known for its fast inference time, which is achieved through a balance between the number of parameters and image quality. Therefore, we started with this model and adapted it for use with sparse convolution in our model, Sparse-E2VID.

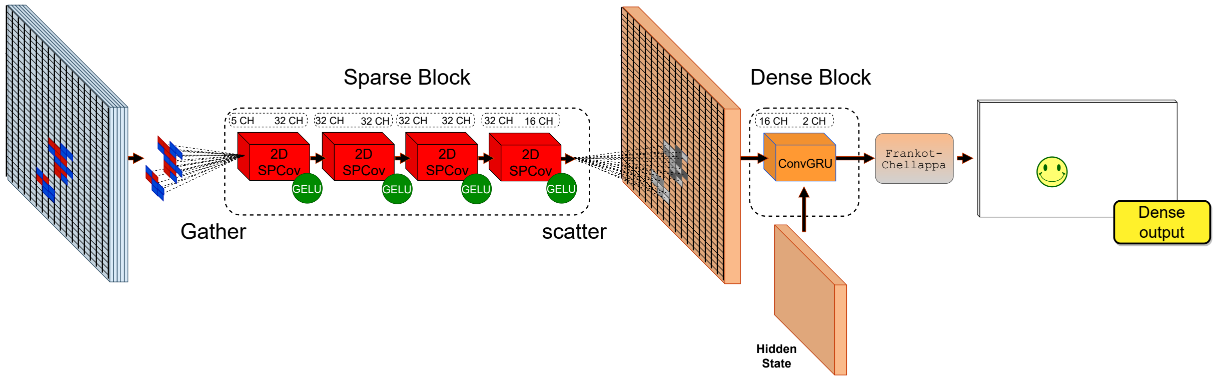


Figure 1. Diagram of Sparse-E2VID. The network comprises two main blocks: the Sparse Block and the Dense Block. The Sparse Block has four 2D Sparse convolution layers with a kernel size of 3 and GELU activation. The Dense Block contains only one layer, which is the Convolutional Gated Recurrent Unit (ConvGRU).

The FireNet architecture includes two ConvGRU modules that are placed at the start and end of the model. However, the location of these modules is flexible, as they can be placed anywhere in the model. Nevertheless, since these modules process the hidden state, which accumulates past information throughout the video reconstruction, it eventually becomes a dense tensor.

To achieve efficiency comparable to or higher than dense convolution, sparse convolution requires a sparse index of at least 80%. As mentioned earlier, the hidden state of the ConvGRU module is a dense tensor. Therefore, in Sparse-E2VID, we use a single ConvGRU module positioned at the end that employs dense convolution. This allows us to use sparse convolution in the initial layers of Sparse-E2VID, with the conversion to a dense format occurring at the ConvGRU module. Figure 1 shows the architecture of our model.

The Sparse Block has four 2D Sparse convolution layers with a kernel size of 3 and GELU activation. The first layer has an input of 5 channels and an output of 32 channels. The second and third layers have an input and output of 32 channels. The fourth layer has an input of 32 channels and an output of 16 channels. The Dense Block contains only one module, which is the ConvGRU unit with 16 channels in the input and 2 channels on the output. All layer uses a kernel size of 3×3. The model estimates the image gradient, and the final image is obtained using the Frankot-Chellappa algorithm [1, 10].

3.2. FFT image integration

Unlike FireNet, E2VID, and other models, Sparse-E2VID predicts the image gradient (later integrated using the FFT). Previous work [9] has shown that by predicting the gradient instead of the full image, we can reduce the size of the neural network while maintaining image quality. However, in this work, the authors used the inverse matrix $\overrightarrow{u} = \mathcal{A}^{-1}\overrightarrow{b}$ for integration [32]. Obtaining this inverse matrix is costly, especially in resolutions equal to or higher than CGA, VGA and HD. Moreover, integrating using the inverse matrix is prohibitively expensive, rendering this method infeasible. To overcome this problem, inspired by early work on event-based image reconstruction [16], we use the Frankot-Chellappa algorithm [1,10], which is shown in equation 1. Where s_x and s_y represent the gradient calculated by Sparse-E2VID, $\mathcal{F}$ is the 2D FFT function, and f_x and f_y are the grid coordinates.

$$\mathcal{F}\left[s\right] = \frac{-if_x\mathcal{F}\left[s_x\right] - if_y\mathcal{F}\left[s_y\right]}{2\pi(f_x^2 + f_y^2)} \tag{1}$$

3.3. Input data

Like FireNet+, we utilize an intermediate event representation [34] in our work. Specifically, we transform the events to voxels, using $B = 5$ bins in the temporal resolution, as shown in equation 2, while preserving the spatial resolution.

$$\begin{aligned} t_k^* &= (B-1) \times \frac{t_k - t_{min}}{t_{max} - t_{min}} \\ \mathsf{E}_{x,y,t} &= \sum p_k max(0, 1 - |t_n - t_k^*|) \end{aligned} \tag{2}$$

However, this tensor is transformed into a sparse COO format to be processed by our Sparse-E2VID model. There are two forms of sparse convolution: ”normal” sparse convolution [13] and submanifold sparse convolution [14]. The ”normal” sparse convolution works like traditional dense convolution, but operations are only performed when the kernel finds a non-zero value. In contrast, with submanifold sparse convolution, operations are only performed when the center of the kernel encounters a non-zero value. Consequently, submanifold convolution is computationally

Algorithm 1 Event Noise concatenation

```
procedure CONCATENATE EVENTS($Noise, Train$)                ▷ Event data in [x, y, t, p] format
    Input: $Noise, Train$
    output: $y$                                               ▷ Noise and Train data concatenation
    $n, x \leftarrow Noise, Train$
    $n_t \leftarrow (n_t - n_t[0])$                           ▷ Subtracted the value of the first timestamp
    $x_t \leftarrow (x_t - x_t[0])$                           ▷ Subtracted the value of the first timestamp
    $n_t \leftarrow n_t / n_t[-1]$                            ▷ Normalize noise data within 0 - 1 values
    $n_t \leftarrow n_t * x_t[-1]$                            ▷ Synchronize noise data
    $idx \leftarrow where(n_{x,y} < max(x_{x,y}))$
    $n \leftarrow n[idx]$                                     ▷ Eliminate events greater than the training resolution
    $y \leftarrow x \parallel n$                              ▷ Concatenate train and noise data
    $y \leftarrow sort(y, y_t)$                               ▷ Sort the concatenated data by its timestamp
```

cheaper, and for this reason, we chose this type of convolution.

3.4. Training data

For training event-based video reconstruction models, event simulators are typically used to produce event sequences that are paired with highly synchronized full-frame images. In our work, similar to FireNet+, we use the ESIM [22] simulator to generate the training set. However, FireNet+ only uses daytime images from the MS COCO dataset [18]. In contrast, we use 30% nighttime images [2] and 70% daytime images from the MS COCO dataset to improve the model's performance in nighttime scenarios.

Another aspect to note is that FireNet+ introduces synthetic noise in the training data to reduce noise in the video reconstruction. However, these models are unable to completely filter out noise, which becomes more noticeable when using higher resolution event cameras or in night scenes. To address this limitation, we propose a simple yet effective solution in which real noise from an event camera is added to the training sequences. This allows Sparse-E2VID to reconstruct almost noise-free images, even in night scenes.

To capture noise from an event camera, we used the Prophesee Gen 4.0 HD camera. To do this, we covered the camera lens and recorded. In some sequences, the event camera was pointed at a non-textured surface, such as a wall, and kept still. The sequences where recorded at different light conditions, daytime, nighttime, indoors and outdors (with no moving object). In total we recorded 20 sequences, each of 10 seconds. We then used these recorded noise sequences in our training data.

To introduce noise into the training data, we followed several steps. Firstly, we subtracted the value of the first timestamp from both the training and noise sequence to ensure they started at time zero. Next, we normalized the timestamps of the noise sequence to a range of 0 to 1. We then multiplied the last timestamp value (i.e., the highest value) from the training sequence by the timestamps of the noise sequence to temporally synchronize the two sequences. Finally, we concatenated the two sequences and sorted them based on their timestamps. The detailed process is shown in algorithm 1.

3.5. Training details

To train our model, we utilized the Many-to-One (M2O) training scheme [4]. While different schemes can be used to train an RNN or ConvRNN model, most models that reconstruct event-based videos use the Many-to-Many (M2M) scheme. In contrast to the M2M scheme, where the loss function is run once for each sample during the training stage, the M2O scheme only runs the loss function once. This reduces training time by 40%, because of the relatively big computational and time cost of the loss function. Additionally, this method enables us to perform data augmentation by varying the number of events in a sequence. Events do not need to be synchronized with the images; they only need to match the last event with the last ground truth image. We also incorporated random stops, similar to FireNet+ and E2VID+. However, for our stops, we added noise from a real event camera, and the stopping probability within a sequence was set to 80%.

The loss function used in this work is a combination of Mean Squared Error (MSE), Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity Index (SSIM). For the LPIPS function, the VGG16 model with 5 intermediate layers was used. Each feature map (coming from each layer) is multiplied by $[\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, 1]$ weight. To calculate the L_2 error of the feature maps later. To prevent the reconstructed images from being blurry, we reduced the impact of MSE by setting $\lambda = 0.2$. The complete loss function is presented in Equation 3.

[2] The night images were obtained from www.pixel.com, a free photo repository.

Table 1. Comparison with SOTA methods of video reconstruction, on low reslution (240 × 180), lower is better.

	1-SSIM			MSE			LPIPS		
Experiment	E2VID+	Sparse E2VID	FireNet+	E2VID+	Sparse E2VID	FireNet+	E2VID+	Sparse E2VID	FireNet+
bike_bay_hdr	**0.55485**	0.60857	0.69178	**0.03348**	0.05736	0.08131	**0.64656**	0.75104	0.85852
boxes	**0.52938**	0.62412	0.64885	**0.04414**	0.09417	0.09466	**0.64613**	0.73270	0.83270
desk	**0.46450**	0.55839	0.56700	**0.03964**	0.11655	0.11315	**0.69371**	0.92372	0.95514
desk_fast	**0.43876**	0.53766	0.54470	**0.03643**	0.09833	0.09517	**0.70014**	0.91419	0.94066
desk_hand_only	**0.46109**	0.50588	0.52888	**0.04664**	0.09878	0.09986	**0.77272**	0.94610	0.95130
desk_slow	**0.39598**	0.50137	0.46651	**0.03918**	0.10439	0.08846	**0.61470**	0.90516	0.80388
engineering_posters	**0.52867**	0.66264	0.64825	**0.03570**	0.07583	0.04618	**0.63258**	0.67328	0.77404
high_texture_plants	**0.44663**	0.72862	0.59296	**0.02252**	0.05081	0.04642	**0.71303**	0.94203	0.89636
poster_pillar_1	**0.57243**	0.67897	0.67910	**0.02254**	0.04540	0.03998	**0.75394**	0.84058	0.87530
poster_pillar_2	**0.57901**	0.58862	0.70727	0.03898	**0.02426**	0.06043	**0.74459**	0.83776	0.90499
reflective_materials	**0.51040**	0.63025	0.67669	**0.03900**	0.07721	0.08352	**0.71379**	0.83758	0.95104
slow_and_fast_desk	**0.48912**	0.57173	0.63475	**0.03013**	0.06778	0.07720	**0.61738**	0.75759	0.86131
slow_hand	**0.57125**	0.63465	0.69921	**0.04071**	0.07412	0.10448	**0.71322**	0.81017	0.95279
still_life	**0.47320**	0.65992	0.62571	**0.02809**	0.09307	0.05226	**0.72851**	0.87256	0.95143
Mean	**0.50109**	0.60652	0.62226	**0.03551**	0.07701	0.07736	**0.69221**	0.83889	0.89353

Table 2. Comparison with SOTA methods of video reconstruction, on HD reslution (720x1280), lower is better.

	1-SSIM			MSE			LPIPS		
Experiment	E2VID+	Sparse E2VID	FireNet+	E2VID+	Sparse E2VID	FireNet+	E2VID+	Sparse E2VID	FireNet+
colition_1	0.60352	**0.39989**	0.58599	0.11473	**0.13023**	0.32517	0.09683	**0.04042**	0.18272
colition_2	**0.23503**	0.52484	0.51225	**0.04566**	0.17263	0.23323	**0.05079**	0.05365	0.14863
cubeBox_night	**0.50315**	0.40037	0.56429	**0.10254**	0.13112	0.32568	0.07542	**0.05023**	0.17193
notebook_night	0.33366	**0.12868**	0.26951	0.04088	**0.02550**	0.04376	0.09135	**0.05922**	0.19272
legoCam_night	0.40923	**0.10897**	0.25289	0.03644	**0.01933**	0.03726	0.10269	**0.06185**	0.19715
Mean	0.41691	**0.31255**	0.43698	**0.06805**	0.09576	0.19302	0.08341	**0.05307**	0.17863

$$\mathcal{L} = \lambda \times MSE + LPIPS + (1 - SSIM) \qquad (3)$$

For training our model, we adopted the one cycle learning rate scheduling policy [29], with a learning rate of $LR = 1 \times 10^{-3}$ and a batch size of 2. Each sequence in the training data consists of 25 samples, and we trained the model for 200 epochs with adamW [19].

4. Results

4.1. Evaluation procedure

To validate our experiments, we used the same test set as FireNet+ and E2VID+, which utilized an event camera with a resolution of 240 × 180. Additionally, we created a small test set with an event camera with a resolution of 720×1280. This additional dataset was included to explore image reconstruction at higher resolutions. The dataset was recorded with a beam splitter. We used the Prophesee Gen 4.0 HD camera event and a global shutter camera at 30 FPS. We transformed the images taken with the global shutter camera to the event camera dimension. A microcontroller synchronized the frames of both cameras, recording the timestamps.

For event grouping (sampling), we use the method of constant number of events. The number of events used is equal to 1% of the spatial resolution of the event camera $W \times H \times 0.01$, where H is the height and W is the width.

To maintain consistency in the measurement of all models, we use custom metrics, with which we evaluate E2VID+, FireNet+ and our Sparse-E2VID model. The image quality metrics used in our evaluation are SSIM, MSE, and LPIPS. The SSIM is calculated by (1-SSIM), so we have an index where a lower value indicates a better image quality.

For the LPIPS, we used a custom function with the VGG16 model that included five layers and their respective weights; $[\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, 1]$. In addition, we assessed the inference speed and computational cost (in terms of average FLOPs) of the models. All tests were conducted on an i7-7700HQ laptop with an Nvidia GTX 1060 MAX-Q featuring 6 GB of video memory (VRAM).

To compute the temporal consistency error, we need to calculate the optical flow between two consecutive ground truth frames; $\mathcal{I}_k$ and $\mathcal{I}_{k-1}$. We use RAFT [31] to obtain the

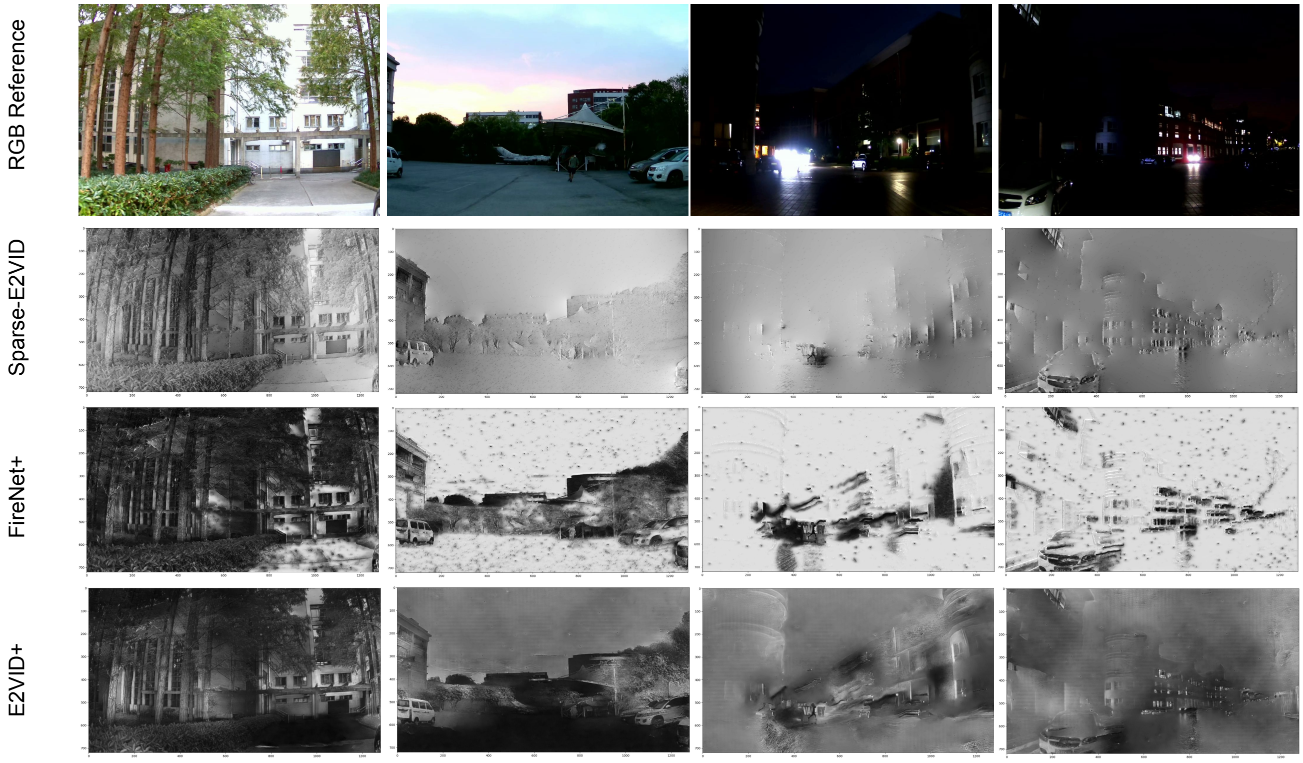


Figure 2. Qualitative results in different light conditions (from left to right), day, sunset and night. We can notice that Sparse-E2VID contains almost no noise and controls the dynamic range better.

Table 3. Models profile at different resolutions, lower is better.

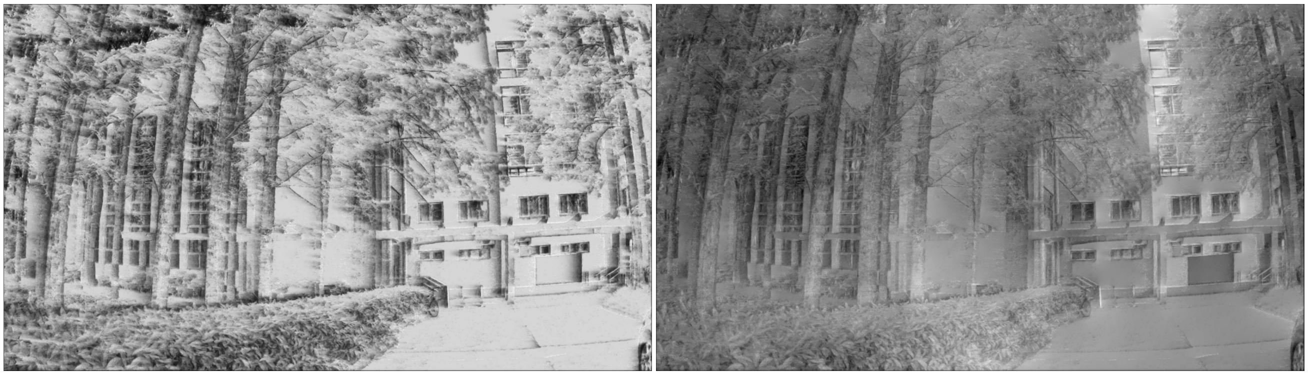
	Models		
	E2VID+	Sparse-E2VID	FireNet+
Num parametres	10710467	**25578**	37777
MFLOPs 720x1280	837237.658	**1822.9**	69423.206
MFLOPs 180x240	40117.638	**85.444**	3254.212
inference time 180x240	18.972 ms	9.045 ms	**5.242 ms**
inference time 720x1280	357.862 ms	**53.811 ms**	90.275 ms
Memory 720x1280	4.8GB	**0.9 GB**	3.0GB
Memory 180x240	0.54GB	0.55GB	**0.52GB**

backward optical flow map $F^k_{k-1} = I_k \Rightarrow I_{k-1}$. Then, we use the warping function $W(\cdot)$ to calculate the past ground truth image $\mathcal{I}^k_{k-1}$ and the past predicted image $\hat{\mathcal{I}}^k_{k-1}$. One thing we need to mention is, we need the past ground truth image $\mathcal{I}^k_{k-1}$ to obtain the mask $\mathcal{M}_k$. Then, the temporal consistency is obtained by comparing the original past predicted image $\hat{\mathcal{I}}_{k-1}$ and the calculated past predicted image $\hat{\mathcal{I}}^k_{k-1}$, as shown in equation 4, where $\alpha = 50$ and $\epsilon = 1e{-}6$.

$$\begin{aligned} \mathcal{I}^k_{k-1} &= W(\mathcal{I}_k, F^k_{k-1}) \\ \hat{\mathcal{I}}^k_{k-1} &= W(\hat{\mathcal{I}}_k, F^k_{k-1}) \\ M_k &= exp(-\alpha \times (\mathcal{I}_{k-1} - \mathcal{I}^k_{k-1})^2) \\ \mathcal{L}^{tc}_k &= \frac{M_k \times \left|\hat{\mathcal{I}}_{k-1} - \hat{\mathcal{I}}^k_{k-1}\right|}{\left|\hat{\mathcal{I}}_{k-1}\right| + \left|\hat{\mathcal{I}}^k_{k-1}\right| + \epsilon} \end{aligned} \tag{4}$$

4.2. Results and discussion

Table 1 demonstrates that, our model, Sparse-E2VID has better results than FireNet+. One thing we want to point



a) Sparse-E2VID variant: Predicts an image directly

b) Original Sparse-E2VID: Predicts an image by the gradient

Figure 3. This figure shows the difference between the original Spade-E2VID and a variant of it. The variant has the same architecture as the original Spade-E2VID, but it skips the gradient and directly reconstructs images from events. As we can see, the original Spade-E2VID generates better quality images than the variant.

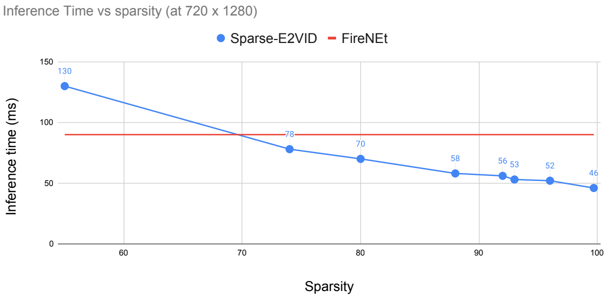


Figure 4. Sparsity vs. Inference Time: Sparse-E2VID has a dynamic inference time that decreases or increases according to the number of events, unlike FireNet or other types of architectures that have a static inference time.

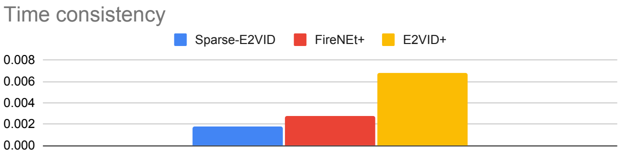


Figure 5. Time consistency in our dataset with 5 sequences in HD resolution. The temporal consistency of Sparse-E2VID is better than FireNet+ and E2VID+, lower is better.

out is that this is achieved with a smaller number of parameters. This is posible, because Sparse-E2VID predicts the gradient of the image, making it easier for the model to perform image reconstruction. According to the event generation model $-\nabla \mathbf{L} \cdot v\Delta t \approx \Delta \mathbf{L}$, were the $v\Delta t$ is the optical flow in a delta time, the image gradient $\nabla \mathbf{L}$ has a more direct relationship with the event data $\Delta \mathbf{L}$. Fugure 3 shows an example of the difference between two variants of Sparse-E2VID. Figure 3 a) shows an image reconstructed by a variant that directly reconstructs images from events without passing through the gradient. Figure 3 b) shows an image predicted by the original Sparse-E2VID model, which has better quality.

At higher resolutions (720×1280), Sparse-E2VID is comparable to E2VID+ and it is superiority to FireNet+, as it is presented in Table 2. It is noteworthy that Sparse-E2VID can reconstruct images with minimal noise, as seen in Figure 2. However, it should be noted that the noise reduction is specific to a certain event camera (the one from which the noise was sampled). If we use a different event camera data to reconstruct images, the video may have some noise (minimal). We can reduce the noise by collecting more noise data from various event cameras.

One thing to note is that Sparse-E2VID has a significantly lower computational cost than FireNet+, with only 2% of the cost, as Table 3 shows. While the computational cost varies with the sparse index, the highest computational cost is found in the module that performs the dense convolution. The ConvGRU module has a computational cost of MFLOPs = 1794 at a resolution of 720 x 1280. Therefore, we can consider the computational cost as constant.

Another important aspect is the speed of inference. Although Sparse-E2VID is slower than FireNet+ at lower resolutions due to the additional step of numerical integration, which takes 5 ms, the impact of this step is less significant at higher resolutions (720×1280). As a result, Sparse-E2VID is 30% to 40% faster than FireNet+. This difference in inference speed is due to the dynamic variation in the number of events in the spatial dimension of event cameras (sparsity).

In Figure 4, we can observe the inference time of Sparse-E2VID and FireNet. Sparse-E2VID has a dynamic inference time that is directly related to the sparsity index. An event camera typically has an average sparsity of 90%, resulting in an average inference time of 55ms for our model. In low motion or night scenarios, we can achieve up to 99% sparsity, resulting in a faster inference time of 45ms. It is worth noting that the sparse rate of an event camera never drops below 80% in normal situations. Therefore, in the worst-case scenario, our model reaches an inference time of 70ms, which is 22% faster than FireNet.

Due to the utilization of the M2O training scheme, we did not to use the time consistency loss function. Despite this, our model is not plagued by flickering or jittering, as evident in Figure 5. Notably, Sparse-E2VID refrains from performing normalization on the input data. As highlighted in SPADE-E2VID [4], the act of normalization can impair temporal consistency. This is due to the fact that during the normalization of the data input, the consistency between frames can be compromised.

5. Conclusion

Our architecture, Sparse-E2VID, offers a significant reduction in the computational cost for event-based video reconstruction. Our model's computational cost is only 2% of that of the FireNet architecture, and it also reduces the inference time by an average of 30%. This was achieved through the use of sparse convolution in the architecture design. Additionally, Sparse-E2VID is effective at reducing noise due to its training with real noise from an event camera.

We hope that the inclusion of real noise in the training sequences and our architecture will inspire new research, not only in event-based image reconstruction but also in other applications.

6. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62103261/62173228/61873165.

References

[1] Amit K. Agrawal, Ramesh Raskar, and Rama Chellappa. What is the range of surface reconstructions from a gradient field? In *European Conference on Computer Vision*, 2006. 3

[2] Ahmed Nabil Belbachir, Stephan Schraml, Manfred Mayerhofer, and Michael Hofstätter. A novel hdr depth camera for real-time 3d 360 panoramic vision. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 425–432. IEEE, 2014. 2

[3] Nathan Bell and Michael Garland. Implementing sparse matrix-vector multiplication on throughput-oriented processors. In *Proceedings of the conference on high performance computing networking, storage and analysis*, pages 1–11, 2009. 1

[4] Pablo Rodrigo Gantier Cadena, Yeqiang Qian, Chunxiang Wang, and Ming Yang. Spade-e2vid: Spatially-adaptive denormalization for event-based video reconstruction. *IEEE Transactions on Image Processing*, 30:2488–2500, 2021. 2, 4, 8

[5] Jonghyun Choi, Kuk-Jin Yoon, et al. Learning to super resolve intensity images from events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2768–2776, 2020. 2

[6] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022. 1

[7] Matthew Cook, Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika Steger. Interacting maps for fast visual interpretation. In *The 2011 International Joint Conference on Neural Networks*, pages 770–776. IEEE, 2011. 2

[8] Yongjian Deng, Hao Chen, Haiyan Liu, and Youfu Li. A voxel graph cnn for object classification with event cameras. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1162–1171, 2021. 2

[9] Hadar Cohen Duwek, Albert Shalumov, and Elishai Ezra Tsur. Image reconstruction from neuromorphic event cameras using laplacian-prediction and poisson integration with spiking and artificial neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1333–1341, 2021. 2, 3

[10] Robert T. Frankot and Rama Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):439–451, 1988. 3

[11] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1

[12] Daniel Gehrig and Davide Scaramuzza. Pushing the limits of asynchronous graph-based object detection with event cameras. *ArXiv*, abs/2211.12324, 2022. 2

[13] Benjamin Graham. Spatially-sparse convolutional neural networks. *ArXiv*, abs/1409.6070, 2014. 3

[14] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *ArXiv*, abs/1706.01307, 2017. 3

[15] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1312–1321, 2021. 2

[16] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43:566–576, 2008. 2, 3

[17] Juan Antonio Leñero-Bardallo, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. A 3.6 mus latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011. 1, 2

[18] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 4

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[20] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 415–431. Springer, 2020. 2

[21] Federico Paredes-Vallés and Guido CHE de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3446–3455, 2021. 2

[22] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018. 2, 4

[23] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3857–3866, 2019. 2

[24] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019. 1, 2

[25] Simon Thomas Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12361–12371, 2022. 2

[26] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision*, pages 308–324. Springer, 2018. 2

[27] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pages 156–163, 2020. 1, 2

[28] Teresa Serrano-Gotarredona and Bernabé Linares-Barranco. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 48(3):827–838, 2013. 1, 2

[29] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. *ArXiv*, abs/1803.09820, 2018. 5

[30] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *European Conference on Computer Vision*, pages 534–549. Springer, 2020. 1, 2

[31] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. *ArXiv*, abs/2003.12039, 2020. 5

[32] Vitaly A Volpert. *Elliptic partial differential equations*, volume 530. Springer, 2011. 3

[33] Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Event-based video reconstruction using transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2563–2572, 2021. 1, 2

[34] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997, 2018. 3