2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

E-VFIA : Event-Based Video Frame Interpolation with Attention

Onur Selim Kılıç, Ahmet Akman and A. Aydın Alatan

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160276

Abstract— **Video frame interpolation (VFI) is a fundamental vision task that aims to synthesize several frames between two consecutive original video images. Most algorithms aim to accomplish VFI by using only keyframes, which is an ill-posed problem since the keyframes usually do not yield any accurate precision about the trajectories of the objects in the scene. On the other hand, event-based cameras provide more precise information between the keyframes of a video. Some recent state-of-the-art event-based methods approach this problem by utilizing event data for better optical flow estimation to interpolate for video frame by warping. Nonetheless, those methods heavily suffer from the ghosting effect. On the other hand, some of kernel-based VFI methods that only use frames as input, have shown that deformable convolutions, when backed up with transformers, can be a reliable way of dealing with long-range dependencies. We propose event-based video frame interpolation with attention (E-VFIA), as a lightweight kernel-based method. E-VFIA fuses event information with standard video frames by deformable convolutions to generate high quality interpolated frames. The proposed method represents events with high temporal resolution and uses a multi-head self-attention mechanism to better encode event-based information, while being less vulnerable to blurring and ghosting artifacts; thus, generating crispier frames. The simulation results show that the proposed technique outperforms current state-of-the-art methods (both frame and event-based) with a significantly smaller model size.**

Multimedia material: The code is available at `https://github.com/ahmetakman/E-VFIA`

I. INTRODUCTION

In robotics applications, where fast-moving agents are involved, high frame-rate video streams are necessary for agile reaction of control systems. There has been dedicated hardware for capturing high frame-rate videos; however, they are quite expensive. Therefore, increasing the frame rate with the involvement of additional processing yields more affordable high frame-rate videos and smoother slow-motion videos.

Video frame interpolation (VFI) is a vision task where a non-captured image sample is inserted between consecutive frames to increase the rate of low frame-rate videos. Since the movements appearing on the scene can be complex and their displacements are large, video frame interpolation is mostly considered as a challenging task.

The state-of-the-art methods that only use frames aim to estimate motion from consecutive images accurately. Such frame-only methods are divided into two mainstream categories, as (i) flow-based methods [1]–[6], and (ii) kernel-based methods [7]–[10].

Although frame-only approaches are regarded as precise when relatively short trajectories are involved, they might fail to estimate complex movements of fast-moving objects. The main reason for these algorithms fail in situations with fast-moving objects is due to the fact that they can only make linear predictions for the trajectories between keyframes to model the complex motion of the objects. Recent advances in mobile camera technologies lead more affordable (relative to the dedicated hardware) devices to have relatively higher frame-rate video recording capability, but this is also limited by the memory requirements of the devices, making it impossible to record long sequences of high frame-rate video by such hardware. Thus, the use of VFI methods still offer memory and storage efficiency.

Event-based cameras are novel vision sensors that detect only pixel-level brightness changes. The pixels in the sensor array asynchronously output information at high speeds (up to 1-μs precision [11]) for a high dynamic range (up to 120 dB [11]). Therefore, event-based cameras are suitable sensors for obtaining additional information related to the moving objects in the scene. On the other hand, the asynchronous and fast nature of the event sensor implicates low data bit-rate due to the limited number of event data, while agile sensing is possible. Therefore, they provide beneficial information in temporal information that applies to VFI problem.

Recent studies [12] [13] argue that using additional information from the event-based sensors might outperform frame-only interpolation methods for VFI. The methods in [12] and [13] are follow-up studies of each other that estimate intermediate frames by warping the images after estimating the motion displacements by the help of events. The aforementioned algorithms have primarily been built on a combination of specific purpose-trained hourglass networks, which are computationally expensive due to the high number of parameters. Based on our preliminary simulations, the interpolated frames in [12] and [13] suffer from ghosting and blurring artifacts.

Transformers [14] are novel neural structures initially developed to model long-range dependencies in natural language processing (NLP) tasks. Researchers took inspiration from the success of the transformers in NLP and applied a similar approach to various vision tasks [15]. The idea of using transformers has been employed recently for frame-only VFI tasks both in kernel-based [7] and flow-based methods [5]. However, they also perform poorly in case of fast-moving objects present in the scene. Aforementioned studies inspire our work for modeling long-term dependencies using transformer structures in a lighter manner.

Motivation. The algorithms proposed in [5], [7], and [16]

All authors are members of Electrical and Electronics Engineering Department and Center for Image Analysis (OGAM) in Middle East Technical University (METU), 06800 Ankara, Turkey {selimk,ahmet.akman,alatan}@metu.edu.tr

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

mainly argue that if the movements of the visual content in consecutive images is encoded by a network and fused with the real images, one can achieve accurate generation of the intermediate frame as long as the optical flow modeling is accurate enough. The fact that frame-only methods suffer from modeling of complex and fast motions leads to use of additional sensor information. The event cameras might be an excellent way to capture more accurate temporal information. Additionally, the success of *deformable convolutions* [17] for synthesizing intermediate frames motivates for a kernel-based video frame interpolation method, which should be based upon deformable convolution concept and backed up with an attention mechanism. In this work, we propose an attention-based, light-weight VFI network, namely *event-based video frame interpolation with attention* (E-VFIA), that outperforms the visual quality of the current SoA event-based VFI methods in BS-ERGB dataset [12].

The main contributions of this work can be summarized as follows:

- We propose E-VFIA, the first kernel-based algorithm to utilize deformable convolutions to fuse event-based information and standard images for video frame interpolation with events.
- E-VFIA achieves significant improvement (up to $1.04dB$) against the state-of-the-art methods that use only key-frames and events together with key-frames. The proposed method achieves this promising result by approximately 2.07 million parameters which is much less than its counterparts, making it more suitable for mobile robotics applications.
- We have utilized voxel grids to represent events. In order to extract the temporal information from the events efficiently, we analyze the effect of voxel grid size on the VFI performance, and it is observed that using voxel grids with higher temporal resolutions improves performance.
- We utilize both temporal and spatial pooling operations to associate fast-moving objects between consecutive images. By the help of such pooling operations, the objects that are moving fast are predicted better, since utilization of images at different resolutions enables data aggregation between different ends of the image.

II. Related Work

Video Frame Interpolation. VFI is a well-studied problem with mature solutions. The recent learning based methods can be divided into three groups: flow-based [1]–[6], kernel-based [7]–[10], and phase-based [18].

Flow-based approaches. Essentially, flow-based methods [1]–[6] estimate intermediate flow from left and right frames. Flow-based methods assume the motion in the scene is linear in its trajectory. Common architectural selections are mostly a variation of hourglass backbones; hence their performance is limited in occlusions, nonlinear trajectories, and brightness changes.

Kernel-based approaches. Kernel-based methods [7]–[10] aim to estimate a series of convolutional kernels that model the movements of the image patches implicitly. However, they might fail to model movements larger than the kernel size. A recent study [8] enables kernel-based methods to model large motions by the development of deformable convolutions concept [17]. A recent promising approach [7] extends this concept by utilizing an encoder-decoder mechanism and aims to model the long-term dependencies.

Phase-based approaches. As indicated in [8], the method in [18] takes video frames as linear combinations of different waveforms with different directions and frequencies. This approach interpolates phase and magnitude of each band of wavelet transform. This method is considered effective and efficient in both performance and runtime. Unfortunately, it is still constrained with large displacements, especially for high-frequency components.

Utilization of Additional Sensors. The idea of using additional information coming from an auxiliary camera has been investigated in the literature by low resolution high frame-rate cameras [19], [20]. It should be noted that the temporal resolutions of event cameras are significantly higher with respect to affordable low resolution high frame-rate cameras.

Event-Based Approaches. Event cameras offer higher dynamic range capability in addition to low data-rate and high temporal information flow. Thus, exploiting the properties of the event camera can be considered as a novel direction to investigate for VFI. Some studies [12], [13], [21]–[24] aim to overcome the VFI problem by using the information from event-based cameras. In a promising approach, [13] flow masks for intermediate frames from event voxels are estimated. Then, the algorithm generates artificial frames by warping the adjacent images by the estimated flow. Finally, an attention averaging module is used for fusing three sister images obtained from warping refinement module. In a two stage method [22], the network fuses events with images in the first stage, whereas in the next part, the fused vectors are evaluated and integrated with a subpixel transformer network. The results in [13] are improved further by introducing a motion spline estimator, and multiscale feature fusion modules [12]. The motion spline estimator enables the pipeline to take into account the continuous nature of the movements on the scene. Since this network [12] has approximately 70M parameters for encoding the events to voxels, the number of time bins is limited in terms of memory. Such an approach leads event representation to be more blurry.

This paper addresses the primary limitations mentioned in [12] and [25]. We extend the approach in [7] by using both events and frames. Since the event information already represents valuable details, we adopt a simpler input stage with a smaller attention mechanism, significantly reducing the number of parameters. The proposed method is relatively lightweight and can represent events in higher temporal resolution with deeper voxel grids while achieving improved interpolation results.

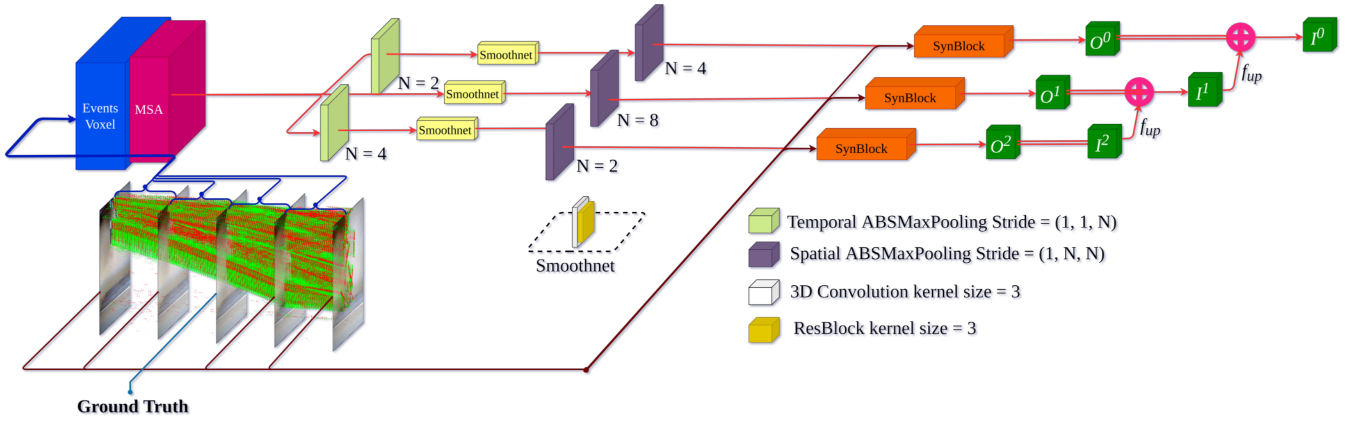


Fig. 1: Overview of the proposed method. Our method consists of three main parts. First events are represented with voxel grids. Then the voxel grids go through the SmoothNet and spatio-temporal absPooling layers. Finally, the constructed feature maps are fused with RGB frames by deformable convolutions that are embedded in parallel SynBlocks.

III. Proposed Method

Our aim is insertion of a non-existent image sample into a video sequence that is composed of four images and four event intervals. Let the interpolated image be denoted by I_0. Assume the input images are denoted as I_{-2}, I_{-1}, I_{+1}, and I_{+2}, which represent the successive original keyframes. The input event intervals are indicated by $E_{-2\to-1}$, $E_{-1\to0}$, $E_{0\to1}$, and $E_{1\to2}$ which are described in terms of voxels whose dimensions are $4 \times N_{TB} \times H \times W$ (N_{TB}: number of time bins, H: height, W: width).

The block diagram of the proposed system is presented in Figure 1. The proposed method is composed of two stages. The events are converted into voxel representation in the first stage and passed to the multi-head self-attention (MSA) block. The resultant vector is passed through a series of blocks in which pooling operations are performed. As a slight difference from the literature, before applying maximum pooling, the absolute values of the vector elements are evaluated. However, after the pooling operation, the polarities of the vector elements are preserved; this stage is denoted as *absPooling*.

As a result of the first stage, three feature vectors are obtained, each of which encodes information coming from different resolutions. By the help of such pooling operations, the objects that are moving fast are predicted better, since using images at different resolutions makes it easier to associate fast-moving objects between consecutive images. In order to make this association even better, both the temporal and spatial domain absPooling operations are used.

In the second stage, the feature vectors due to events resulting from the first step are provided as input to three SynBlocks (Synthesis Blocks [7]) to fuse with RGB images. Every SynBlock takes four RGB frames and one event feature vector. Each SynBlock outputs an intermediate frame downscaled with $l \in 0, 1, 2$. The outputs of these three SynBlocks are combined to create the final output of our algorithm, which is shown in Figure 1, where f_{up} stands for bilinear upscaling, and O^l is an output of a SynBlock. Such a downscaling/upscaling structure helps the network to model trajectories of fast-moving objects better.

Event-Volume Representation and Input Stage. Event data coming from an event-based camera is asynchronous and sparse. The data is usually in μs precision in time and streamed as a series of events (packages of $[x, y, timestamp, polarity]$). In a voxel representation, timestamps in the voxel grid are accumulated according to the number of time bins. The voxels for our case have dimensions of $4 \times N_{TB} \times H \times W$ and a similar approach as in [13] is followed for the generation of the voxels. The first event interval $E_{-2\to-1}$ and the second event interval $E_{-1\to0}$ are used directly, but the third $E_{0\to1}$ and the fourth $E_{1\to2}$ event intervals are utilized as time-reversed. The first stage consists of two aforementioned absPooling layers. The first pooling layer calculates absPooling operation in the temporal dimension, whereas the second pooling layer calculates the same operation on its spatial counterpart. The multi-head self-attention mechanism (MSA), whose number of heads is 16, calculates attention values over the temporal dimension. These results are fed into SmoothNet blocks where these blocks simply include convolutional and ResNet layers, as in [7]. This step is performed for the spatial smoothness of event feature vectors before the second (spatial) pooling operation. Therefore, the resultant three feature vectors, which encode different event-depth information, are processed in the first stage of the proposed method.

Fusion with Frames. SynBlocks are adapted from [7] and presented in Figure 2. SynBlocks are the main synthesis blocks that can be considered as a multi-frame generalization of [8]. The input part of a SynBlock starts from unbinding operation that the feature vectors F^l are divided into four in the temporal dimension. Each of these four temporal parts passes three parallel convolutional blocks generating four sisters of convolutional kernels. Each convolutional block constructs these convolutional kernels, and these kernels are created by 2D convolutions operating over event feature maps that produce deformable kernels for images. In other words, these three kernels are weights $W_t^l \in R^{K\times H\times W}$, horizontal offsets, $\alpha_t^l \in R^{K\times H\times W}$ and vertical offsets $\beta_t^l \in R^{K\times H\times W}$, where K stands for the number of sampling locations of every kernel. Using these predicted kernels, the result of the deformable convolution is reached, as in [8]:

$$O_t^l(x,y) = \sum_{n=1}^{K} W_t^l(n,x,y) I_t^l(x+\alpha_t^l(n,x,y), y+\beta_t^l(n,x,y))$$

Therefore, RGB frames, I_t^l, are fused by the help of feature vectors, W_t^l, α_t^l and β_t^l that are obtained from convolutional blocks to reach deformable convolution block operation [8]. In the last step of a SynBlock, O_t^l is blended with learned masks. The masks are acquired by using standard convolutional neural networks on the feature map that is the concatenation of the SynBlock input. The result of a SynBlock is obtained by elementwise multiplication,

$$O^l = \sum_t B_t^l \cdot O_t^l$$

Finally, O_t^l vectors are elementwise multiplied with the vector output of standalone convolutional block B_t^l.

As each SynBlock generates the output at scale $l \in 0, 1, 2$ the intermediate image is determined as,

$$\hat{I}_0^l = f_{up}(\hat{I}_0) + O^l$$

where f_{up} stands for bilinear upscaling, and O^l is an output of a SynBlock. This operation concludes the fusion of frames with event feature maps and constructs intermediate frames.

The proposed method is a relatively simple yet effective approach for the utilization of additional temporal information supplied by the events. Differing from [7], the initial multi-head self-attention block in the input layer is small-sized and significantly reduces the model size, while still modeling trajectories by the help of precise event information. On the other hand, it should be pointed out that the voxels are supplied into the network in a dense manner, which means the sparsity of the events is not fully exploited in our work.

IV. Experiments and Results

A. Implementation Details

All of our work on deep learning is performed by using PyTorch [26] framework. The training has performed for 36 epochs, where the learning rate, starting from 0.0008, is halved every eight epochs. We have used AdaMax optimizer [27] with $\beta_1 = 0.9, \beta_2 = 0.999$ and the training batch size is set as 6. The training has been executed on a workstation with two 2080TI GPUs. Due to the memory requirements, the training has been performed with images and associated events whose resolution is 256×256. We have tested and compared the proposed method with both full-scaled and downscaled frames with 256×256 resolution, since the obtained quantitative results can differ significantly with different sized images.

B. Quantitative Results

All the algorithms are tested on BS-ERGB [12] dataset. It should be noted that the test results of [13] are replicated from [12] directly. For the algorithms in [7] and [25], we have conducted experiments by using the codes provided by the authors and the listed the best results.

TABLE I: Comparison of our proposed method in BS-ERGB dataset in low resolution

Method	Input	#Parameters (M)	PSNR (dB)	SSIM
FLAVRFV [25]	Frames	42.4	31.72	0.9469
VFIT [7]	Frames	29.0	32.08	0.9449
Timelens [13]	Frames Events	72.2	28.36	0.9320
Timelens++ [12]	Frames Events	53.9	28.56	-
Ours	**Frames Events**	**2.07**	**32.23**	**0.9581**

TABLE II: Comparison of our proposed method in BS-ERGB dataset in full scale

Method	Input	#Parameters (M)	PSNR (dB)	SSIM
FLAVRFV [25]	Frames	42.4	27.642	0.8729
VFIT [7]	Frames	29.0	28.00	0.8767
Timelens [13]	Frames Events	72.2	23.97	0.7838
Timelens++ [12]	Frames Events	53.9	-	-
Ours	**Frames Events**	**2.07**	**29.04**	**0.8771**

Tables I and II present the quantitative results for BS-ERGB dataset in full-scale and down-scaled to 256×256, where peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) values are used for comparison. Due to the rich temporal information stored in event voxels and the utilization of deformable convolutions, the proposed method clearly outperforms the current state-of-the-art event-based method by a significant margin of 5 dB PSNR on BS-ERGB dataset [12]. Moreover, the proposed method surpasses the state-of-art frame-only method by 1 dB, while utilizing only 6.9% of its original model size. As plotted in 3 ,and 4 the proposed method significantly reduces the number of parameters compared to both event-based and frame-only methods. That is, the proposed method promises faster and more feasible local computation option.

C. Qualitative Results

We have also compared the proposed method qualitatively by other frame-only and event-based methods with the full-scale BS-ERGB [12] dataset. Typical examples of the outcomes are illustrated in Figures 5 and 6. As it can be observed, our method provides crispier interpolated frames. The 7 shows that the proposed method achieves more accurate results in all three channels as the other methods have more heterogeneous pixel-wise difference maps. Furthermore, the

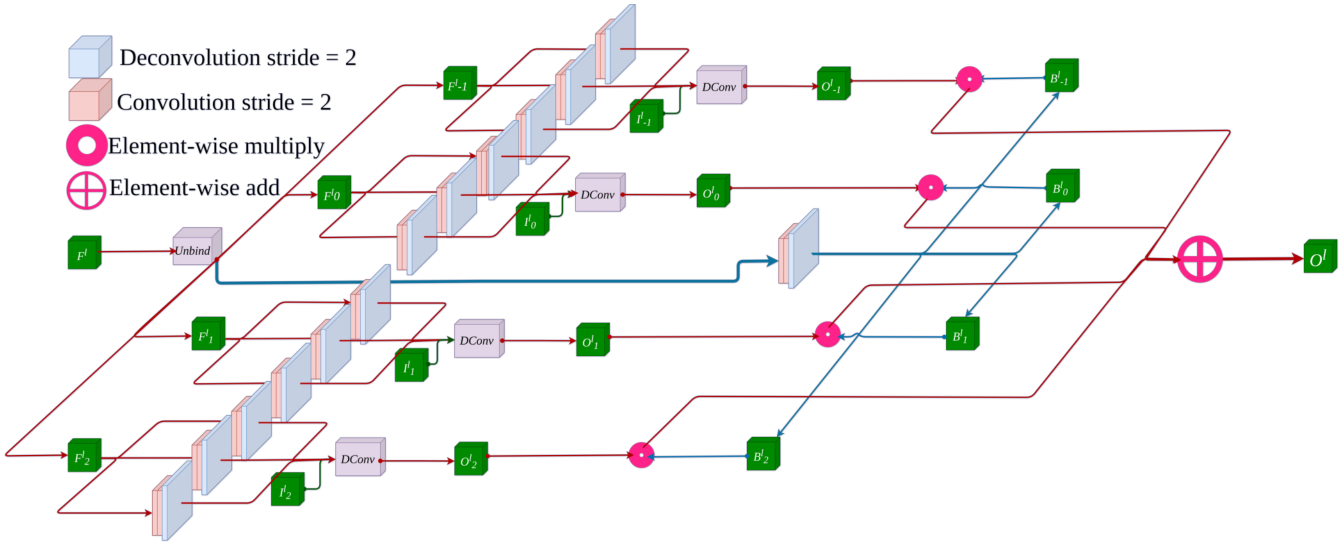


Fig. 2: The general structure of SynBlocks. A SynBlock takes four RGB frames and one event feature vector. Using these inputs, the SynBlocks fuse the event-based information and RGB images using kernels by utilizing deformable convolutions.

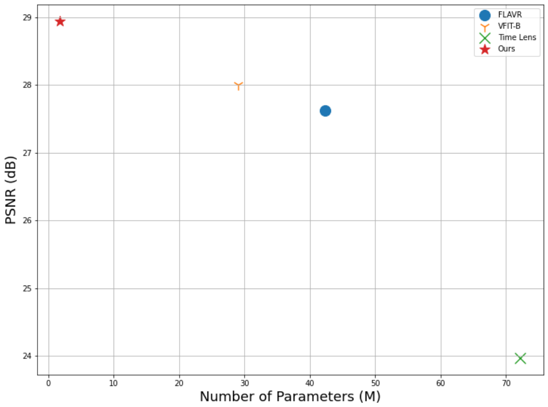


Fig. 3: Comparison of the performance of different algorithms (FLAVR [25], VFIT-B [7], Time Lens [13], Time Lens++ [12], and our proposed method): number of parameters vs. PSNR values for BS-ERGB [Full Scale] [12] dataset.

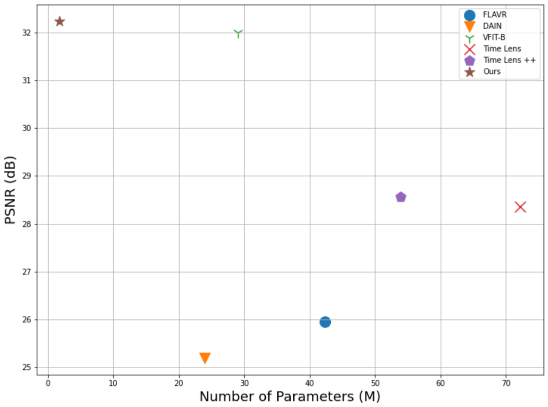


Fig. 4: Comparison of the performance of different algorithms (FLAVR [25], VFIT-B [7], Time Lens [13], Time Lens++ [12], and our proposed method): number of parameters vs. PSNR values for BS-ERGB [12] dataset.

objects that have complex motion with non-linear trajectories are estimated more accurately compared to other techniques.

D. Ablation Studies

We have conducted a series of ablation studies to better comprehend the proposed strategy. First, in order to validate the effectiveness of the multi-head self-attention mechanism, we have discarded this mechanism during simulations. Additionally, we have also analyzed the effect of Sep-STS layers proposed in [7] on event voxels.

1) Results for Events without Attention Mechanism: To investigate the significance of the attention mechanism, we have trained a network without the mechanism in the input stage.

The quantitative results in Table III indicate that the attention mechanism in the input stage of the network increases the performance by 0.59dB. This result is important, since the temporal attention mechanism intensifies the crucial elements of the voxels.

TABLE III: Comparison in BS-ERGB dataset

Method	**PSNR (dB)**	**SSIM**
w/o MSA	31.64	0.9496
Ours	**32.23**	**0.9581**

2) Result for Events with Sep-STS Layers: The authors of [7] have introduced Sep-STS blocks for the encoder part

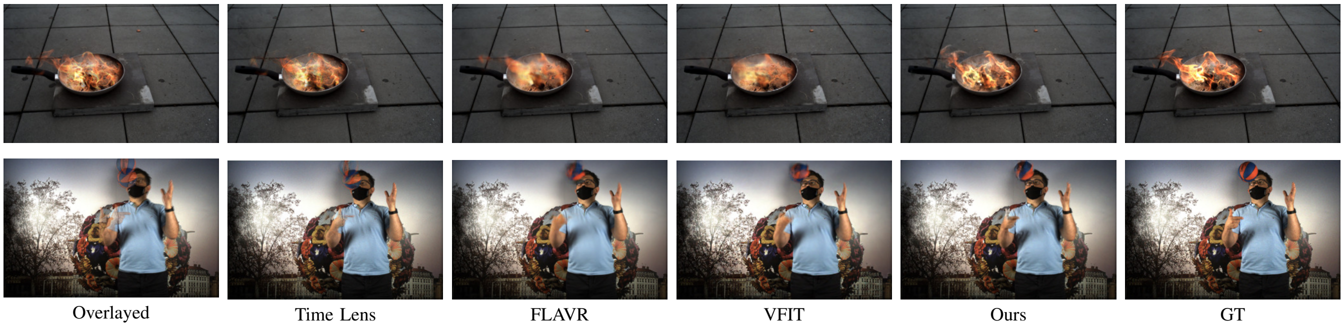


Fig. 5: Qualitative comparisons against the state-of-the-art video interpolation algorithm. Our method is less prone to blur and ghosting effects. Thus, it provides more realistic increments on frame-rates of real videos.

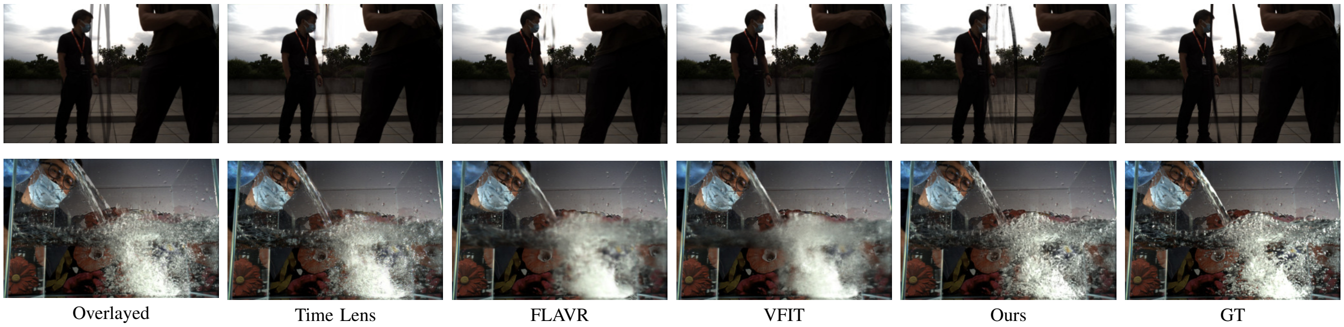
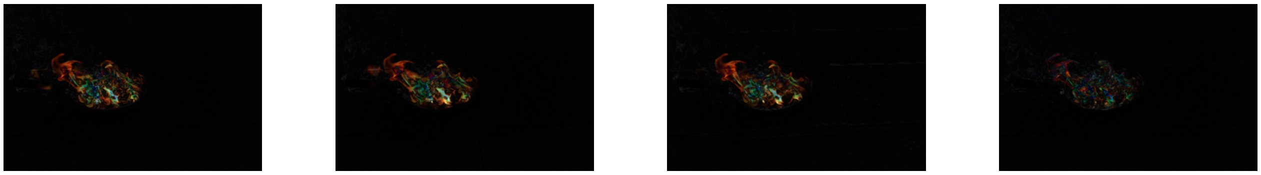


Fig. 6: Qualitative comparisons against the state-of-the-art video interpolation algorithms.



(a) Time Lens (b) FLAVRFV (c) VFIT (d) Ours

Fig. 7: Pixel-wise differences of the fire sample.

of the network adapted from [8]. We have compared the encoder of [7] with the first part of the proposed algorithm to investigate the performance of Sep-STS blocks by using events with both approaches.

The comparative results of these experiments are given in Table IV. Although the number of parameters presented by the Sep-STS blocks is relatively large, utilization of a simpler input stage yields better results due to the valuable temporal information in events. One can argue that Sep-STS might require more training data to yield better inference.

TABLE IV: Comparison in BS-ERGB dataset

Method	**PSNR (dB)**	**SSIM**
Sep-STS Encoder	31.19	0.9436
Ours	**32.23**	**0.9581**

V. Conclusion

In this paper, we propose E-VFIA, a lightweight and high quality video frame interpolation algorithm that uses frames obtained from RGB cameras, as well as event information obtained from event cameras. The proposed method encodes event information with a multi-head self-attention mechanism and fuses the encoded information with frames under deformable convolution-based synthesis blocks. Consequently, the E-VFIA achieves 3.87 dB and 5.07 dB PSNR improvement over the state-of-the-art event-based method for low and high-resolution images, respectively. The model achieves this result by only 3.8% of its model size. It should be emphasized that based on qualitative comparisons, our method is less prone to blur and ghosting artifacts, whereas in some cases, suffers from the color inconsistency of the fast-moving objects. Finally, it can be argued that the proposed deformable convolutions strategy is a promising approach to fuse event information and visual appearance in any vision or robotics application, specifically VFI.

References

[1] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, “Depth-aware video frame interpolation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] J. Park, K. Ko, C. Lee, and C.-S. Kim, "Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation," in *European Conference on Computer Vision*, 2020.

[3] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, "Quadratic video interpolation," in *NeurIPS*, 2019.

[4] H. Sim, J. Oh, and M. Kim, "Xvfi: extreme video frame interpolation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[5] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video frame interpolation with transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 3532–3542.

[6] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[7] Z. Shi, X. Xu, X. Liu, J. Chen, and M.-H. Yang, "Video frame interpolation transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 482–17 491.

[8] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee, "Adacof: Adaptive collaboration of flows for video frame interpolation," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5315–5324, 2020.

[9] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[10] S. Niklaus, L. Mai, and O. Wang, "Revisiting adaptive convolutions for video frame interpolation," in *IEEE Winter Conference on Applications of Computer Vision*, 01 2021, pp. 1098–1108.

[11] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 44, no. 01, pp. 154–180, jan 2022.

[12] S. Tulyakov, A. Bochicchio, D. Gehrig, S. Georgoulis, Y. Li, and D. Scaramuzza, "Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 755–17 764.

[13] S. Tulyakov, D. Gehrig, S. Georgoulis, J. Erbach, M. Gehrig, Y. Li, and D. Scaramuzza, "Time lens: Event-based video frame interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 16 155–16 164.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] H. Lee, T. Kim, T.-y. Chung, D. Pak, Y. Ban, and S. Lee, "Adacof: Adaptive collaboration of flows for video frame interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," 2017. [Online]. Available: https://arxiv.org/abs/1703.06211

[18] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, and C. Schroers, "Phasenet for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 498–507.

[19] A. Paliwal and N. K. Kalantari, "Deep slow motion video reconstruction with hybrid imaging system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1557–1569, jul 2020. [Online]. Available: https://doi.org/10.1109%2Ftpami.2020.2987316

[20] A. Gupta, P. Bhat, M. Dontcheva, B. Curless, O. Deussen, and M. Cohen, "Enhancing and experiencing spacetime resolution with videos and stills," in *International Conference on Computational Photography*. IEEE, 2009. [Online]. Available: http://grail.cs.washington.edu/projects/enhancing-spacetime/

[21] S. Lin, J. Zhang, J. Pan, Z. Jiang, D. Zou, Y. Wang, J. Chen, and J. Ren, "Learning event-driven video deblurring and interpolation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 695–710.

[22] Z. Yu, Y. Zhang, D. Liu, D. Zou, X. Chen, Y. Liu, and J. Ren, "Training weakly supervised video frame interpolation with events," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 14 569–14 578.

[23] J. Han, Y. Yang, C. Zhou, C. Xu, and B. Shi, "Evintsr-net: Event guided multiple latent frames reconstruction and super-resolution," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 4862–4871.

[24] Z. Wang, Y. Ng, C. Scheerlinck, and R. Mahony, "An asynchronous kalman filter for hybrid event cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 448–457.

[25] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran, "Flavr: Flow-agnostic video representations for fast frame interpolation," *ArXiv*, vol. abs/2012.08512, 2020.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980