

TimeReplayer: Unlocking the Potential of Event Cameras for Video Interpolation

Weihua He¹ Kaichao You^{2,4} Zhendong Qiao¹ Xu Jia³✉ Ziyang Zhang²✉
Wenhui Wang¹✉ Huchuan Lu^{3,5} Yaoyuan Wang² Jianxing Liao²

¹Department of Precision Instrument, Tsinghua University

²Advanced Computing and Storage Lab, Huawei Technologies Co. Ltd

³School of Artificial Intelligence, Dalian University of Technology

⁴School of Software, Tsinghua University ⁵Peng Cheng Laboratory

{hwh20@mails, wwh@mail}@.tsinghua.edu.cn, youkaichao@gmail.com

{xjia, lhchuan}@dlut.edu.cn, {zhangziyang11, wangyaoyuan1, liaojianxing}@huawei.com

Abstract

Recording fast motion in a high FPS (frame-per-second) requires expensive high-speed cameras. As an alternative, interpolating low-FPS videos from commodity cameras has attracted significant attention. If only low-FPS videos are available, motion assumptions (linear or quadratic) are necessary to infer intermediate frames, which fail to model complex motions. Event camera, a new camera with pixels producing events of brightness change at the temporal resolution of μs (10^{-6} second), is a game-changing device to enable video interpolation at the presence of arbitrarily complex motion. Since event camera is a novel sensor, its potential has not been fulfilled due to the lack of processing algorithms. The pioneering work Time Lens introduced event cameras to video interpolation by designing optical devices to collect a large amount of paired training data of high-speed frames and events, which is too costly to scale. To fully unlock the potential of event cameras, this paper proposes a novel TimeReplayer algorithm to interpolate videos captured by commodity cameras with events. It is trained in an unsupervised cycle-consistent style, canceling the necessity of high-speed training data and bringing the additional ability of video extrapolation. Its state-of-the-art results and demo videos in supplementary reveal the promising future of event-based vision.

1. Introduction

Traditionally, recording fast motion in high temporal resolution has been the exclusive functionality of high-speed cameras, which are too expensive to be embedded in personal devices such as smartphones. Video interpolation, the research field of inferring intermediate frames between two

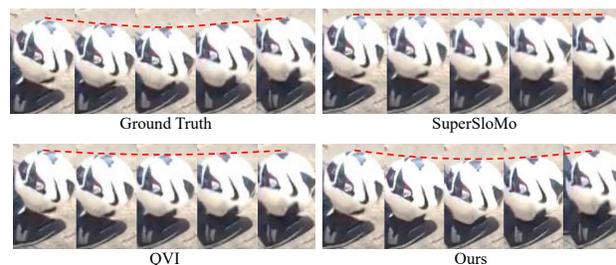


Figure 1. Visual results of interpolating complex motion. Both models with linear motion assumption (SuperSloMo [9]) and quadratic motion assumption (QVI [38]) fail to model the motion of the soccer ball correctly. Benefiting from the high temporal resolution of event data, the proposed method can correctly model nonlinear motion and interpolate intermediate frames. **This figure contains animations. Best viewed in Acrobat Reader.**

given frames, has attracted considerable research interest. It can go beyond the temporal resolution limit of commodity cameras, and can be applicable in diverse downstream tasks such as slow motion generation [2, 9, 38], video editing [32, 46], and virtual reality [1]. Without any further information or proper assumption, the video interpolation problem is under-determined. In the past years, the computer

Corresponding authors: Xu Jia, Ziyang Zhang, and Wenhui Wang.
Webpage: <https://sites.google.com/view/timereplayer>.

vision community has explored a lot in the direction of exploiting proper assumptions. [9] proposes SuperSloMo for video interpolation with linear motion assumption, and [38] improves the result based on a quadratic motion assumption. As the assumption becomes more and more complicated, two obstacles get in the way: (1) linear motion [9] can interpolate video given two frames, but quadratic motion [38] requires four consecutive frames to calculate the acceleration for quadratic interpolation. As the assumption becomes more complex, more frames and more computational costs are needed to interpolate every frame. (2) the underlying motion in the video can be arbitrary, making it difficult to verify pre-set assumptions on the motion type. When the assumption and the motion type mismatch, the interpolated video may look unrealistic.

To deal with the inherent lack of intermediate information of conventional frame-based cameras, we bring in a novel neuromorphic sensor, event camera [6, 14, 33, 40], as a promising solution for video interpolation in case of complex motion. The event camera senses the dynamic change of pixel intensity, where an event is triggered once the change exceeds a certain threshold. The asynchronously event-driven processing fashion leads to a frame-less event stream with microsecond temporal resolution, as well as low power consumption and bandwidth [8, 28]. Since event camera almost records continuously streaming intensity change, it is able to store rich inter-frame information, which is critical to recovering intermediate video frames. The introduction of event camera for video interpolation helps ease the difficulty in modeling complex motion.

Another crucial issue with the task of video interpolation is that it is difficult to collect ground truth for video interpolation in the real world scenarios. Most existing video interpolation models [2, 9, 23, 38] are trained on high-frame-rate dataset, which is constructed by taking an average of consecutive frames of high-frame-rate videos recorded by high-speed cameras. However, these models would suffer from the domain gap between synthetic data and real world data. Therefore, it is important to equip the models with the learning ability to generalize to data without ground truth. There have been a few works [18, 31] aiming to address this issue. They base their methods on cycle consistency, where multiple intermediate frames are predicted and are then used to reconstruct a middle input frame. However, to achieve cycle consistency, these methods have to assume uniform motion between consecutive frames at large timesteps, which makes them also suffer from the same problems as those uniform motion assumption-based methods.

In this work we introduce data from event camera into the video interpolation model design and propose an unsupervised learning framework to train the video interpolation model. Specifically, we introduce event stream to help directly estimate optical flows between an intermediate frame

and input frames instead of computing them as proportions of the computed optical flow between input frames. In this way the uniform linear motion assumption is broken and the proposed optical flow estimation module can compute any complex nonlinear motion. Then an intermediate frame can be predicted by warping input frames according to the estimated optical flows and taking a weighted average of them. With the help of event streams and approximated inverse ones, a video interpolation model is able to predict an intermediate frame given two input frames and can also reconstruct an input frame when the estimated intermediate frame and another input frame are given. Therefore, loss functions can be computed between the reconstruction of input frames and the original ones and are used to train the video interpolation model. Extensive experiments and ablation study on synthetic benchmark datasets demonstrate the effectiveness of the proposed framework, especially on videos with complex motion. In addition, we also further conduct an experiment on real data to validate its generalization ability. Contributions of our work can be summarized as below.

- A novel video frame interpolation method with both video frame and event stream as input is proposed to address complex nonlinear motion.
- We design an unsupervised learning framework for video interpolation with event stream by applying cycle consistency.
- The proposed method performs favorably against state-of-the-art approaches on both synthetic benchmarks datasets and real data. When trained with additional unsupervised data, it achieves the best result thanks to the unsupervised nature, showing the promising future of event-based video interpolation.

2. Related Work

Video Frame interpolation Current state-of-the-art interpolation methods are mainly based on warping bidirectional optical flow. However, most of them [3, 9, 19, 20, 31] simply assumed uniform motion and linear optical flow between consecutive frames, which may fail in approximating the complex and nonlinear movement in the real world. Thus, the nonlinear QVI methods [17, 38] are proposed to solve this problem by learning higher-order acceleration information between frames. But the errors in acceleration estimation may lead to deviation from the motion trajectory in the ground truth. Another essential problem in video interpolation is there are no paired ground truth frames for model supervision in the low-frame-rate video. Therefore, fine-tuning of the supervised method is impossible to be performed on target low-frame-rate image sequence, and the model may suffer performance degradation when applied to the target source. Unsupervised interpolation method [31]

Table 1. Overview of video interpolation methods, including the proposed TimeReplayer in this paper.

Input Modality	Method	Motion assumption	Training data collection	Supervision
RGB frames	SuperSloMo [9]	linear motion assumption	high speed camera	supervised
	QVI [38]	quadratic motion assumption	high speed camera	supervised
	UnSuperSloMo [31]	linear motion assumption	high speed camera	unsupervised
RGB frames + Events	TimeLens [36]	no motion assumption	high speed camera + event camera	supervised
	TimeReplayer	no motion assumption	commodity camera+ event camera	unsupervised

is established to overcome the problem, obtaining supervision from original input frames based on cycle consistency. However, limitations exist since the method assumes longer temporal consistency in three consecutive frames, further exacerbating the disadvantages of the linear models when facing complexity in real world.

Event Camera As a neuromorphic sensor, event camera [6, 14, 33, 40] asynchronously triggers events once the log intensity change of per-pixel reaches threshold. The unique working pipeline brings high temporal resolution (1 μ s), high dynamic range (140dB), and low power consumption. Previous research has established that the event-driven can achieve great advantages in the image and video synthesis, thus various event-driven algorithms have been developed for downstream visual tasks such as motion deblurring [10, 15, 24], optical flow estimation [12, 44], high dynamic range imaging [7] and visual tracking [41]. One of the branches [21, 30] is to reconstruct event streams as intensity images to recover high-frame-rate image sequence. However, the corrupted event signals affect the quality of the reconstructed images. With the development of a novel event camera, DAVIS [5], the event channel and the conventional intensity channel are integrated and share the same photo-sensor array, thus avoiding the registration problem and enabling complementarity between event cameras and conventional intensity cameras. By simultaneously using intensity image and high-speed event stream, we can achieve better visual performance in interpolation tasks. Another high-frame-rate synthesis branch [10, 15, 24] is to recover multiple sharp images from the motion-blurring frames. However, the sequence can be reconstructed only within the temporal range of obtained image information by the intensity channel, which is powerless for moments that are not covered by intensity exposure.

Cycle Consistency Cycle consistency has been widely applied to establish constraints in absence of the direct supervision, such as 3d dense correspondences [43], disambiguating visual relations [42], and unpaired image-to-image translation [45]. When challenged with interpolation task, the fully unsupervised method [31] based on cycle consistency could learn behaviors and synthesize high-frame-rate interpolation from any target lower-frame-rate video sequence. As the most related method to our work, the previous unsupervised method [31] generates two in-

termediate frames between three consecutive frames, and then interpolate a frame using the generated frames to match with the intermediate original input frame. However, to achieve cycle consistency, it has to use multiple input frames and assume uniform motion between consecutive frames at large timesteps, which makes them suffer from artifacts caused by inaccurate motion prediction.

3. Method

In this section, we first describe the representation of event data used in this framework and then explain how event stream contributes to the video interpolation process in case of complex nonlinear motion. Finally, we describe the proposed unsupervised training framework for video interpolation with event data.

3.1. Event Representation.

Event camera asynchronously captures log intensity change $\Delta \log(I_{x,y,t})$ for each pixel and an event $\mathcal{E}_{x,y,t,p}$ is triggered once the change at position (x, y) and timestamp t reaches a certain threshold τ . Therefore, it is able to "continuously" capture intensity change with extreme low latency (in the order of μ s). The polarity p of an event means the sign of $\Delta \log(I_{x,y,t})$, with 1 and -1 representing the positive and negative events respectively. Event stream is a sequence of events within the time period $[t_0, t_0 + T]$ containing N events: $e_{t_0 \rightarrow (t_0+T)} = \{\mathcal{E}_{x_i,y_i,t_i,p_i}\}_{i=0}^{N-1}$.

The raw form of the spatial-temporal event stream is a sequence of '1's and '-1's at different time stamps, which is not friendly to standard neural network models such as CNNs, hence not suitable for direct processing by those networks. Raw event stream needs to be first transformed into an image-like input. In this work event stream is represented as a 4-channel image with the first two channels encoding counts of positive and negative events at each pixel, while the other two channels encoding corresponding to timestamps of the latest triggered positive/negative events at each pixel [26].

As a directional process quantity, the event stream $e_{t_0 \rightarrow t_0+T}$ could be approximately reversed by simultaneously exchanging the positive/negative polarity of events and changing the order of timestamps, whose use would be detailed in video interpolation model. The approximation is referred to as $e'_{(t_0+T) \rightarrow t_0} = \{\mathcal{E}_{x_i,y_i,t_i,-p_i}\}_{i=N-1}^0$.

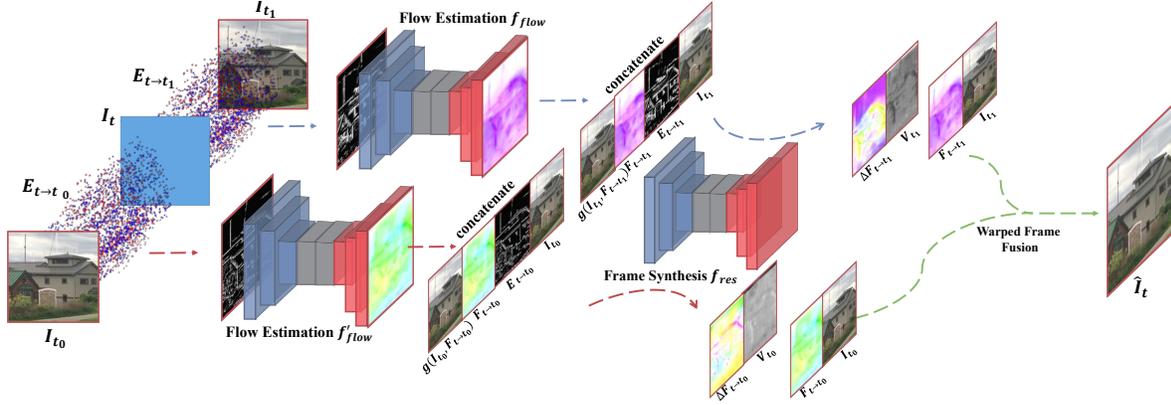


Figure 2. The proposed TimeReplayer model architecture with event stream.

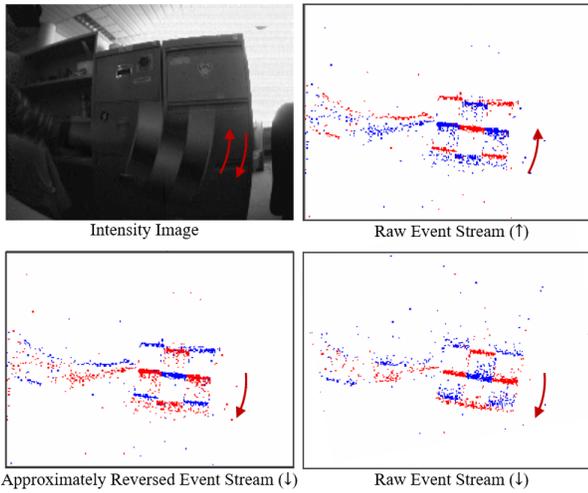


Figure 3. Visualization of reverse events. We can see that although approximately reversed events for downward motion look similar to the actual events for downward motion, there is still much difference between them, especially on non-edge regions.

In fact, due to the characteristics of the hardware circuit of the event camera, the approximately reversed event stream $e'_{(t_0+T)\rightarrow t_0}$ is not exactly the same as the actual backward streaming events. An example is shown in Fig. 3. Therefore we propose to process the original forward streaming event streams and the approximately reversed ones separately. We use $E_{t_0\rightarrow t_1}$ to refer to the event stream from t_0 to t_1 as $E_{t_0\rightarrow t_1} = \begin{cases} e_{t_0\rightarrow t_1}, & \text{if } t_0 \leq t_1 \\ e'_{t_0\rightarrow t_1}, & \text{if } t_0 > t_1 \end{cases}$

3.2. Video Interpolation Model

The proposed model is based on SuperSloMo [9] with three modules, which are described in the following.

Flow estimation. Since event stream inherently carries continuous motion information with itself, optical flow can

be directly computed by feeding event representation to a convolutional neural network. However, due to the difference between approximately reversed event stream and the actual one, we take two flow estimation CNN modules f_{flow} and f'_{flow} , which have the same network architecture but do not share parameters, to deal with forward streaming events $E_{t\rightarrow t_1}$ and backward streaming ones $E_{t\rightarrow t_0}$ respectively. The correspondingly computed optical flows for $E_{t\rightarrow t_1}$ and $E_{t\rightarrow t_0}$ are respectively denoted as $F_{t\rightarrow t_1}$ and $F_{t\rightarrow t_0}$. Compared to most previous methods, which assume that objects move along a straight line at a constant speed and compute the optical flow between an input frame and the targeted intermediate frame as a proportion of the optical flow between two input frames, this method can make a more accurate estimation of optical flow between the interpolated frame and input frames in case of nonlinear motion.

Flow refinement. However, event-based optical flow estimation methods tend to produce sparse output concentrating on edges of moving objects. Thus, another CNN $f_{flow,res}$ is used to refine the initially estimated sparse optical flows $F_{t\rightarrow t_0}$ and $F_{t\rightarrow t_1}$ by computing a residual flow ΔF . The final estimation is obtained by taking the sum of the initial estimation and a residual.

$$\begin{aligned} (\Delta F_{t\rightarrow t_0}, V_{t_0}) &= f_{res}(g(I_{t_0}, F_{t\rightarrow t_0}), I_{t_0}, F_{t\rightarrow t_0}, E_{t\rightarrow t_0}) \\ (\Delta F_{t\rightarrow t_1}, V_{t_1}) &= f_{res}(g(I_{t_1}, F_{t\rightarrow t_1}), I_{t_1}, F_{t\rightarrow t_1}, E_{t\rightarrow t_1}), \end{aligned} \quad (1)$$

where $g(I_{t_0}, F_{t\rightarrow t_0})$ and $g(I_{t_1}, F_{t\rightarrow t_1})$ are warped input frames computed using initially estimated optical flows $F_{t\rightarrow t_0}$ and $F_{t\rightarrow t_1}$, $\Delta F_{t\rightarrow t_0}$ and $\Delta F_{t\rightarrow t_1}$ denote the residual used to refine the optical flows, and V_{t_0} and V_{t_1} represent visibility maps which help alleviate artifacts and occlusion in the later blending process. Two sets of feature maps are separately concatenated and passed through a shared optical flow refinement model $f_{flow,res}$.

Frame synthesis. Finally the target frame \hat{I}_t can be synthesized by blending the warped input frames using refined optical flows. The blending process is taken as a weighted

average of two warped frames with the product of time interval and visibility map as weights, as described in Eq. 2.

$$\hat{I}_t = \frac{1}{Z} \odot ((t_1 - t)V_{t_0} \odot g(I_{t_0}, F_{t \rightarrow t_0} + \Delta F_{t \rightarrow t_0}) + (t - t_0)V_{t_1} \odot g(I_{t_1}, F_{t \rightarrow t_1} + \Delta F_{t \rightarrow t_1})) \quad (2)$$

where Z is a normalization term computed as the sum of $(t_1 - t)V_{t_0}$ and $(t - t_0)V_{t_1}$.

Therefore, as long as two input frames at two time stamps t_0 and t_1 , and the event streams between these two time stamps and the targeted one t are given, we could synthesize the desired frame at that time stamp with the proposed whole video interpolation model f .

3.3. Temporal Cycle Consistency

In this work we focus on training a video interpolation model in an unsupervised fashion [31]. Inspired by cycle consistency, which is first proposed in CycleGAN [45] and UNIT [16] and then becomes a popular constraint in self-supervised learning, we present an unsupervised training framework for video interpolation with event data. As explained in previous section, with the help of event data, the model is able to synthesize an intermediate frame between two input frames at different time stamps; similarly, it can also synthesize an input frame given the predicted intermediate frame and another input frame. In this way, we can impose supervision on the reconstruction of input frames to train the video interpolation model. Specifically, as for an interval between time t_0 and t_1 , given a pair of input frames I_{t_0} and I_{t_1} , and event stream $E_{t_0 \rightarrow t_1}$, then we can use the video interpolation model f to produce an intermediate frame \hat{I}_t in-between as $\hat{I}_t = f(I_{t_0}, I_{t_1}, E_{t \rightarrow t_0}, E_{t \rightarrow t_1})$, where f denotes the whole video interpolation model, $t \in (t_0, t_1)$ denotes the target interpolation time, and $E_{t \rightarrow t_0}$ and $E_{t \rightarrow t_1}$ respectively represent approximately reversed streaming events from time t to time t_0 and the actual forward streaming ones from t to time t_1 .

Similarly, with t_1 as target time, given a pair of input frames I_{t_0} and \hat{I}_t , and the corresponding event streams from time t_1 to t_0 and from time t_1 to t , the same video interpolation model f could be applied to reconstruct frame I_{t_1} as $\hat{I}_{t_1} = f(\hat{I}_t, I_{t_0}, E_{t_1 \rightarrow t}, E_{t_1 \rightarrow t_0})$, where $E_{t_1 \rightarrow t_0}$ and $E_{t_1 \rightarrow t}$ are approximately reversed event streams.

With t_0 as target time, given a pair of input frames I_{t_1} and \hat{I}_t , and the corresponding event streams from time t_0 to t_1 and from time t_0 to t , the same video interpolation model f could also be applied to reconstruct frame I_{t_0} as $\hat{I}_{t_0} = f(\hat{I}_t, I_{t_1}, E_{t_0 \rightarrow t}, E_{t_0 \rightarrow t_1})$, where $E_{t_0 \rightarrow t}$ and $E_{t_0 \rightarrow t_1}$ are the corresponding actual event streams.

The interpolation procedure is visualized in Fig. 4.

In this way, the estimated intermediate frames \hat{I}_{t_0} and \hat{I}_{t_1} could be respectively paired with the original input frames I_{t_0} and I_{t_1} . Therefore we can train the interpolation model

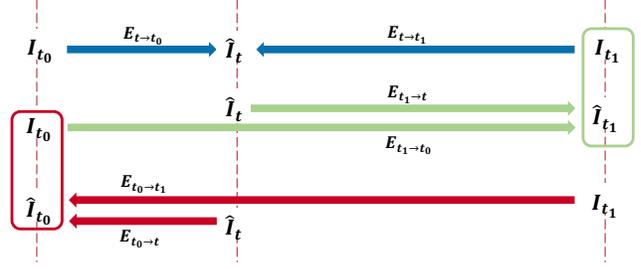


Figure 4. Temporal cycle consistency. I_{t_0} and I_{t_1} are the original input frames, and $E_{t_0 \rightarrow t}$ and $E_{t \rightarrow t_1}$ are input event streams. After obtaining an estimated intermediate frame \hat{I}_t , we could further apply the same video interpolation model to reconstruct I_{t_1} and I_{t_0} , which can be used for imposing cycle-consistent learning.

f by minimizing the reconstruction error between them, *i.e.* cycle consistency loss as $\mathcal{L}_{cc} = \|\hat{I}_{t_0} - I_{t_0}\|_1 + \|\hat{I}_{t_1} - I_{t_1}\|_1$.

The final loss function integrates the above terms $\mathcal{L}_{total} = \lambda_c \mathcal{L}_{cc} + \lambda_w \mathcal{L}_{warp} + \lambda_s \mathcal{L}_{smooth} + \lambda_p \mathcal{L}_{percep}$, where warping loss \mathcal{L}_{warp} , smoothness loss \mathcal{L}_{smooth} and perceptual loss \mathcal{L}_{percep} are defined in a similar way as in SuperSloMo [9] but for both predictions at t_0 and t_1 .

4. Experiment

We use Pytorch [27] to implement the proposed framework. 256×256 patches are cropped from the corresponding consecutive frames and the batch size is set to 28. The models are trained using Adam optimizer [11] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ on 4 NVIDIA Tesla V100 GPUs. The learning rate is initially set to $1e^{-4}$, and then down-scaled by a factor of 0.1 every 200 epochs till 500 epochs. We take peak-signal-to-noise (PSNR), structural-similarity-image-metric (SSIM), and interpolation-error (IE) as a quantitative evaluation metrics for the video interpolation task. Methods with high PSNR and SSIM scores and low IE scores are favored in video interpolation. We use arrows to show the favored values, *i.e.* PSNR \uparrow , SSIM \uparrow , and IE \downarrow .

4.1. Comparison with synthetic events

Training data with synthetic events. For synthetic data, we collected video sequences from the training set of GoPro [22], Adobe240 [35] and Vimeo90k [39]. The GoPro dataset contains 22 diverse videos used for training and 11 additional ones for testing, while the Adobe240 dataset has 112 sequences for training and 8 ones for testing. Both video datasets are recorded with GoPro camera at a frame rate of 240 fps with a resolution of 1280×720 . The Vimeo90k dataset has 51,312 triplets for training, where each triplet contains 3 consecutive video frames with a resolution of 448×256 pixels. All the collected video sequences are paired with corresponding synthetic events generated using popular video-to-event simulator ESIM [29].

Table 2. Quantitative comparison on Adobe240, GoPro, Middlebury (other) and Vimeo90k (interpolation) dataset with synthetic events.

Method	Frame	Event	Supervision	Adobe240						GoPro					
				7 skip (whole)			7 skip (center)			7 skip (whole)			7 skip (center)		
				PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓
E2VID [30]	✗	✓	✓	10.40	0.570	75.21	10.32	0.573	76.01	9.74	0.549	79.49	9.88	0.569	80.08
SepConv [23]	✓	✗	✓	32.31	0.930	7.59	31.07	0.912	8.78	29.81	0.913	8.87	28.12	0.887	10.78
QVI [38]	✓	✗	✓	32.87	0.939	6.93	31.89	0.925	7.57	31.39	0.931	7.09	29.84	0.911	8.57
DAIN [2]	✓	✗	✓	32.08	0.928	7.51	30.31	0.908	8.94	30.92	0.901	8.60	28.82	0.863	10.71
SuperSloMo [9]	✓	✗	✓	31.05	0.921	8.19	29.49	0.900	9.68	29.54	0.880	9.36	27.63	0.840	11.47
Time Lens [36]	✓	✓	✓	35.47	0.954	5.92	34.83	0.949	6.53	34.81	0.959	5.19	34.45	0.951	5.42
UnSuperSloMo [31]	✓	✗	✗	29.92	0.908	9.10	29.36	0.898	9.85	28.23	0.861	10.35	27.32	0.836	11.67
Ours	✓	✓	✗	34.14	0.950	6.25	33.22	0.942	6.64	34.02	0.960	5.02	33.39	0.952	5.59

Method	Frame	Event	Supervision	Middlebury (other)						Vimeo90k (interpolation)					
				3 skip			1 skip			3 skip			1 skip		
				PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓	PSNR↑	SSIM↑	IE↓
E2VID [30]	✗	✓	✓	11.26	0.427	69.73	11.82	0.403	70.15	-	-	-	10.08	0.395	79.89
SepConv [23]	✓	✗	✓	25.51	0.824	6.74	30.16	0.904	3.93	-	-	-	33.80	0.959	3.15
QVI [38]	✓	✗	✓	26.31	0.827	6.58	31.02	0.908	3.78	-	-	-	-	-	-
DAIN [2]	✓	✗	✓	26.67	0.838	6.17	30.87	0.899	4.86	-	-	-	34.20	0.962	3.03
SuperSloMo [9]	✓	✗	✓	26.14	0.825	6.33	29.75	0.887	4.65	-	-	-	32.93	0.948	3.50
Time Lens [36]	✓	✓	✓	32.13	0.908	4.07	33.27	0.929	3.17	-	-	-	36.31	0.962	2.38
UnSuperSloMo [31]	✓	✗	✗	24.86	0.789	7.40	28.27	0.873	4.98	-	-	-	30.38	0.930	3.77
Ours	✓	✓	✗	30.91	0.887	4.89	32.74	0.912	3.63	-	-	-	35.12	0.963	2.79

Table 3. Quantitative comparison on HQF and HS-ERGB dataset with real-world events.

Method	Frame	Event	Supervision	HQF				HS-ERGB (far)				HS-ERGB (close)			
				3 skip		1 skip		7 skip		5 skip		7 skip		5 skip	
				PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
E2VID [30]	✗	✓	✓	6.70	0.315	6.70	0.315	7.01	0.372	7.05	0.374	7.68	0.427	7.73	0.432
DAIN [2]	✓	✗	✓	26.10	0.782	29.82	0.875	27.13	0.748	27.92	0.780	28.50	0.801	29.03	0.807
SuperSloMo [9]	✓	✗	✓	25.54	0.761	28.76	0.861	24.16	0.692	25.66	0.727	27.27	0.775	28.35	0.788
RRIN [13]	✓	✗	✓	26.11	0.778	29.76	0.874	23.73	0.703	25.26	0.738	27.46	0.800	28.69	0.813
BMBC [25]	✓	✗	✓	26.32	0.781	29.96	0.875	24.14	0.710	25.62	0.742	27.99	0.808	29.22	0.820
Time Lens [36]	✓	✓	✓	30.57	0.900	32.49	0.927	32.31	0.869	33.13	0.877	31.68	0.835	32.19	0.839
UnSuperSloMo [31]	✓	✗	✗	23.47	0.740	26.11	0.852	23.80	0.651	25.81	0.700	26.72	0.732	28.38	0.741
Ours	✓	✓	✗	28.82	0.866	31.07	0.931	30.07	0.834	31.98	0.861	29.83	0.816	31.21	0.818

Results. The proposed method is compared with several state-of-the-art interpolation methods, including supervised ones such as Time Lens [37], SuperSloMo [9], quadratic video interpolation (QVI) [38], depth-aware video frame interpolation (DAIN) [2] and an unsupervised one unsupervised SuperSloMo [31]. Similar to Time Lens, Popular video interpolation benchmark datasets are chosen for evaluating our method, such as GoPro, Adobe240, Vimeo90k (interpolation), and Middlebury (other). For GoPro and Adobe240 datasets, we skip 7 frames of each sequence and form a new sequence as input, and the task reconstructs the skipped ones using the interpolation method. Both average performances over the whole 7 skipped frames and the one of the center skipped one are presented for comparison. As for Vimeo90k (interpolation) and the Other set of Middlebury, 1 or 3 frames are skipped respectively due to the limited sequence length. For comparison, we simply use

the pretrained model on the same dataset provided by the authors. The results are summarised in Tab. 2.

Table 2 shows the quantitative comparison with state-of-the-art methods. The method with only event data E2VID performs worst because event data is sparse and only contains relative intensity change, which is not sufficient to reconstruct high-quality intermediate frames. With the help of event data, methods with both intensity images and event data as input, *i.e.* Time Lens, and the proposed method perform favorably against those methods with only intensity images as input, *i.e.* SuperSloMo, DAIN, SepConv, and QVI. Note that our method, which is trained following an unsupervised fashion, even outperforms most supervised methods. This could be attributed to its powerful capability in accurately modeling nonlinear motion by making full use of almost continuously captured event streams.

As shown in the first row of Fig. 5, when cars are parked

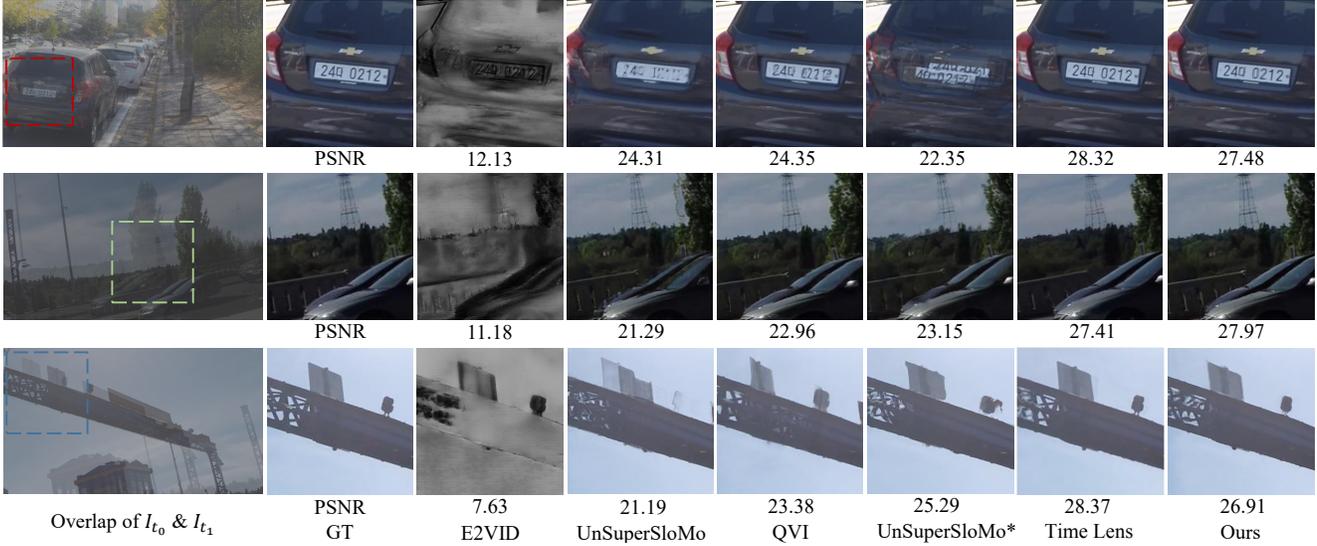


Figure 5. Visual comparison on the Adobe240 and GoPro datasets. In the left, two paired input frames are plotted together; columns in the right are interpolation results from different methods, with a focus on the boxed area.

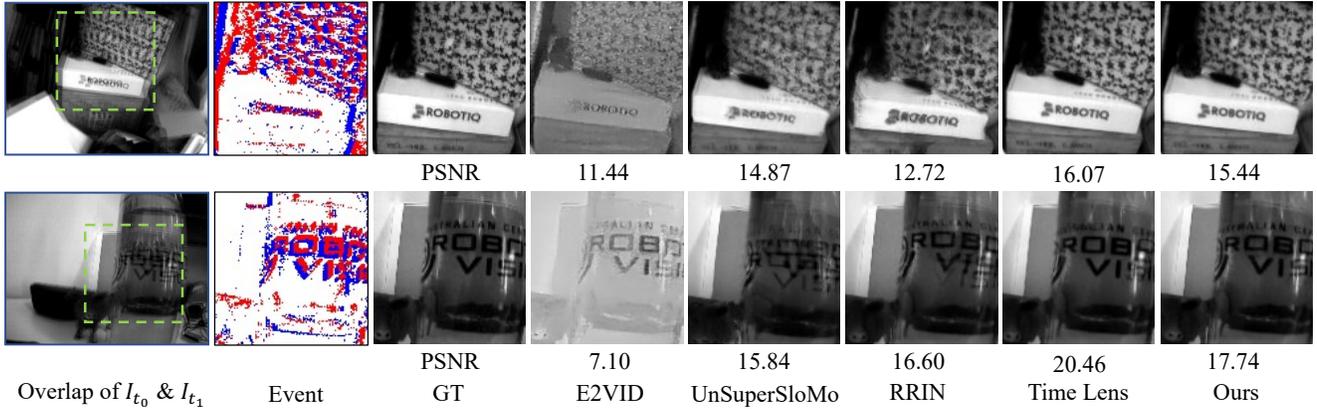


Figure 6. Visual comparison on the HQF dataset.

on the side of the road and the handheld camera is moving, there is a change with car’s position in the image. Only Time Lens and our method, which are able to model motion accurately, can predict a reasonable intermediate frame, especially on license plate. Other methods suffer from the inconsistency of estimation from two input frames, thereby producing artifacts on the numbers. Similar results can also be observed in the other two examples. This demonstrates that event data can indeed help model complex motion with a high-quality intermediate frame interpolation.

4.2. Comparison on real event

Experiments above are conducted on synthetic datasets, where event streams are simulated. In this section, we investigate how the proposed method performs on real events in real-world scenarios. Since simulated events are free from background noise and limited read-out bandwidth,

there is a gap between simulated and real events. Hence, the performance of methods that are trained on synthetic dataset with the supervision of ground truth may degrade on real data.

Training data with real events. High Quality Frames (HQF) [34], and HS-ERGB [36] datasets are chosen for fine-tuning our model trained on synthetic dataset (Vimeo90k). Collected using DAVIS240 event camera, HQF dataset consists of 14 frame sequences with the corresponding event streams in a resolution of 240×180 since both the event channel and the conventional intensity channel are integrated on DAVIS240. HS-ERGB dataset contains sequences from a dual-camera setup, which consists of Prophesee Gen4M 720p event camera and FLIR Black-Fly S RGB camera.

Results. Here we evaluate our method on HQF and HS-ERGB datasets [36], both containing video frame se-

quences without blur and saturation and the corresponding real-world event streams. Far-away and close planar scenes are divided in HS-ERGB dataset for testing since different rectifications are applied. The quantitative results are summarised in Tab. 3. Similar results as the synthetic experiment can be conducted by the quantitative comparison. As shown in Fig. 6, methods without event input produce many artifacts, especially on the edges. The words in the frame could be found suffering when reconstructing frame sequences using frame-only interpolation methods while avoiding artifacts with the help of event input (Time Lens and ours).

4.3. Ablation study

In this section, we investigate the contribution of each component in the proposed approach on the GoPro dataset and report experimental results in Table 4. Three variants are: (1) “linear motion model”: the optical flows between two input frames are first estimated with both intensity images and event data, and then the flow from two input frames to the target intermediate frame is computed as a proportion of the previously estimated flow. It aims to validate the importance of accurate motion modeling and the effectiveness of the proposed method in modeling complex nonlinear motion. (2) “shared flow est module”: to investigate the influence caused by the difference between actual event stream and approximately reversed one to the whole framework, we experiment with only one optical flow estimator which is used to deal with both of them. (3) “UnSuperSloMo*”: to demonstrate the effectiveness of the proposed method, we also compare it with the enhanced version of unsupervised SuperSloMo, where the four-channel event representation is taken as additional input and concatenated with input images before being fed to the model. As shown in Table 4, all these components contribute to the performance of video interpolation.

Table 4. Ablation results on GoPro (7 skip).

Method	PSNR↑	SSIM↑	IE↓
linear motion	32.05	0.939	6.40
shared flow est module	33.78	0.958	5.16
UnSuperSloMo*	32.16	0.942	6.31
Ours	34.02	0.960	5.02

4.4. Benefit of unsupervised VFI with events

An important advantage of our proposed method is its unsupervised nature. Previous methods generate high quality ground truth frames for training VFI model in a supervised manner. They require either carefully controlled slow shooting like HQF [34], or complex registration among high-speed cameras and event cameras [36], which can not be easily obtained with low cost. In contrast, an unsuper-

vised approach can benefit from arbitrary videos and paired events, which could be easily obtained at scale via DAVIS sensor. And the large scale unsupervised training data could further improve the VFI performance.

To demonstrate the benefit of unsupervised training, we select the large scale DDD-17 dataset [4] for unsupervised finetuning, which containing over 12h record of a 346x260 pixel DAVIS sensor. DAVIS sensor produces low-speed frames together with paired events, which cannot be used by previous methods due to the lack of high-speed intermediate frames. Since our TimeReplayer algorithm relies on unsupervised cycle-consistent training, DDD-17 dataset can be used for unsupervised finetuning. The results are summarized in Tab. 5.

Table 5. Quantitative comparison on HQF dataset with real events.

Method	3 skip		1 skip	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Time Lens-syn	28.98	0.873	30.57	0.903
Time Lens-real	30.57(+1.59)	0.900(+0.027)	32.49(+1.92)	0.927(+0.024)
Ours	28.82	0.866	31.07	0.931
Ours+DDD17	31.54(+2.72)	0.920(+0.054)	33.93(+2.86)	0.934(+0.003)

To get better performance, Time Lens [36] needs to be fine-tuned with high-cost real event data. In contrast, our method can be fine-tuned using *low-cost unsupervised event data*, which surpasses Time Lens thanks to the abundance of data in DDD17.

5. Conclusion

In this paper we introduce event camera into the video interpolation framework to help reconstruct intermediate frames at the presence of complex motion. We propose the TimeReplayer algorithm, a cycle-consistency based unsupervised training method which allows a video interpolation model to be trained in an unsupervised way with only two frames and events. It reduces the requirement of large amounts of paired high-frame-rate videos and unlocks the potential of event data in video frame interpolation. Natural data collected by DAVIS camera at a low cost can be used to improve the performance of TimeReplayer. Extensive experiments on both synthetic datasets and real data demonstrate the effectiveness of the proposed method in addressing complex nonlinear motion and reconstructing high-quality intermediate frames.

6. Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No. 2021ZD0200300), and the Natural Science Foundation of China, No. 62106036, 61725202, U1903215, and the Fundamental Research Funds for the Central University of China, DUT No. 82232026.

References

- [1] Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz. Jump: virtual reality video. *ACM Transactions on Graphics (TOG)*, 35(6):1–13, 2016. **1**
- [2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. **1, 2, 6**
- [3] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 2019. **2**
- [4] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Ddd17: End-to-end davis driving dataset. *arXiv preprint arXiv:1711.01458*, 2017. **8**
- [5] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240×180 130 db 3 μ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. **3**
- [6] Tobi Delbruck. Frame-free dynamic digital vision. In *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, volume 1, pages 21–26. Citeseer, 2008. **2, 3**
- [7] Jin Han, Chu Zhou, Peiqi Duan, Yehui Tang, Chang Xu, Chao Xu, Tiejun Huang, and Boxin Shi. Neuromorphic camera guided high dynamic range imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1730–1739, 2020. **3**
- [8] Weihua He, YuJie Wu, Lei Deng, Guoqi Li, Haoyu Wang, Yang Tian, Wei Ding, Wenhui Wang, and Yuan Xie. Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences. *Neural Networks*, 132:108–120, 2020. **2**
- [9] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. **1, 2, 3, 4, 5, 6**
- [10] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2020. **3**
- [11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. **5**
- [12] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *European Conference on Computer Vision*, pages 366–382. Springer, 2020. **3**
- [13] Haopeng Li, Yuan Yuan, and Qi Wang. Video frame interpolation via residue refinement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2613–2617. IEEE, 2020. **6**
- [14] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 μ s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008. **2, 3**
- [15] Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy Ren. Learning event-driven video deblurring and interpolation. In *European Conference on Computer Vision*, volume 3, 2020. **3**
- [16] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. **5**
- [17] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *European Conference on Computer Vision*, pages 41–56. Springer, 2020. **2**
- [18] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8794–8802, 2019. **2**
- [19] Ziwei Liu, Raymond A Yeh, Xiaou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017. **2**
- [20] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1418, 2015. **2**
- [21] Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, 126(12):1381–1393, 2018. **3**
- [22] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. **5**
- [23] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. **2, 6**
- [24] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2019. **3**
- [25] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 109–125. Springer, 2020. **6**
- [26] Paul K. J. Park, Baek Hwan Cho, Jin Man Park, Kyoobin Lee, Ha Young Kim, Hyo Ah Kang, Hyun Goo Lee, Jooyeon

- Woo, Yohan Roh, Won Jo Lee, Chang-Woo Shin, Qiang Wang, and Hyunsurk Ryu. Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1624–1628, 2016. [3](#)
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. [5](#)
- [28] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, et al. Towards artificial general intelligence with hybrid tianjic chip architecture. *Nature*, 572(7767):106–111, 2019. [2](#)
- [29] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982. PMLR, 2018. [5](#)
- [30] 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*, 2019. [3](#), [6](#)
- [31] Fitsum A Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Unsupervised video interpolation using cycle consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 892–900, 2019. [2](#), [3](#), [5](#), [6](#)
- [32] Wenqi Ren, Jingang Zhang, Xiangyu Xu, Lin Ma, Xiaochun Cao, Gaofeng Meng, and Wei Liu. Deep video dehazing with semantic segmentation. *IEEE Transactions on Image Processing*, 28(4):1895–1908, 2018. [1](#)
- [33] Teresa Serrano-Gotarredona and Bernabé Linares-Barranco. A 128×128 1.5% contrast sensitivity 0.9% fpn $3 \mu\text{s}$ latency 4 mw asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. *IEEE Journal of Solid-State Circuits*, 48(3):827–838, 2013. [2](#), [3](#)
- [34] 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 *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 534–549. Springer, 2020. [7](#), [8](#)
- [35] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. [5](#)
- [36] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time Lens: Event-Based Video Frame Interpolation. In *CVPR*, 2021. [3](#), [6](#), [7](#), [8](#)
- [37] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. TimeLens: Event-based video frame interpolation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. [6](#)
- [38] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *arXiv preprint arXiv:1911.00627*, 2019. [1](#), [2](#), [3](#), [6](#)
- [39] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. [5](#)
- [40] Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A dynamic vision sensor with 1% temporal contrast sensitivity and in-pixel asynchronous delta modulator for event encoding. *IEEE Journal of Solid-State Circuits*, 50(9):2149–2160, 2015. [2](#), [3](#)
- [41] Zheyu Yang, Yujie Wu, Guanrui Wang, Yukuan Yang, Guoqi Li, Lei Deng, Jun Zhu, and Luping Shi. Dashnet: A hybrid artificial and spiking neural network for high-speed object tracking. *arXiv preprint arXiv:1909.12942*, 2019. [3](#)
- [42] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1426–1433. IEEE, 2010. [3](#)
- [43] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 117–126, 2016. [3](#)
- [44] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018. [3](#)
- [45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. [3](#), [5](#)
- [46] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004. [1](#)