Event-driven Video Frame Synthesis

Zihao W. Wang† Weixin Jiang† Kuan He† Boxin Shi‡ Aggelos Katsaggelos† Oliver Cossairt†
†Northwestern University ‡Peking University
{winswang, weixinjiang2022}@u.northwestern.edu

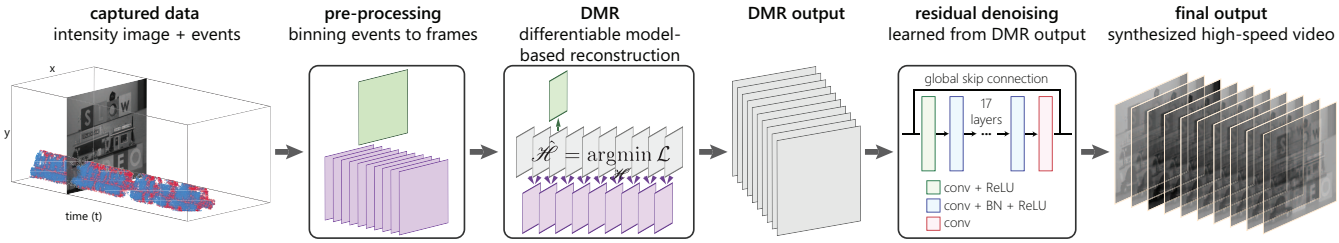


Figure 1: We propose a fusion framework of intensity image(s) and events for high frame-rate video synthesis. Our synthesis process includes a differentiable model-based reconstruction and a residual "denoising" process.

Abstract

Temporal Video Frame Synthesis (TVFS) aims at synthesizing novel frames at timestamps different from existing frames, which has wide applications in video codec, editing and analysis. In this paper, we propose a high frame-rate TVFS framework which takes hybrid input data from a low-speed frame-based sensor and a high-speed event-based sensor. Compared to frame-based sensors, event-based sensors report brightness changes at very high speed, which may well provide useful spatio-temoral information for high frame-rate TVFS. Therefore, we first introduce a differentiable fusion model to approximate the dual-modal physical sensing process, unifying a variety of TVFS scenarios, e.g., *interpolation, prediction and motion deblur. Our differentiable model enables iterative optimization of the latent video tensor via autodifferentiation, which propagates the gradients of a loss function defined on the measured data. Our differentiable model-based reconstruction does not involve training, yet is parallelizable and can be implemented on machine learning platforms (such as TensorFlow). Second, we develop a deep learning strategy to enhance the results from the first step, which we refer as a residual "denoising" process. Our trained "denoiser" is beyond Gaussian denoisers and shows properties such as contrast enhancement and motion awareness. We show that our framework is capable of handling challenging scenes including both fast motion and strong occlusions. Demo and code are released at: https://github.com/winswang/int-event-fusion/tree/win10.*

1. Introduction

Conventional video cameras capture intensity signals at fixed speed and output signals frame by frame. However, this capture convention is motion agnostic. When the motion in the scene is significantly faster than the capturing speed, the motion is usually under-sampled, resulting in motion blur or large discrepancies between consecutive frames, depending on the shutter speed (exposure time). One direct solution to capture fast motion is to use high speed cameras, in exchange with increased hardware complexity, degraded spatial resolution and/or reduced signal-to-noise ratio. Moreover, high speed moments usually happen instantaneously in-between regular-speed context. Consequently, either we end up collecting long sequences of frames with a great amount of redundancy, or the high-speed moment is missed before we realize to turn on the "slow-motion" mode.

We argue that high speed motion can be acquired and synthesized effectively by augmenting a regular-speed camera with a bio-inspired event camera [8, 24]. Compared to conventional frame-based sensors, event pixels *independently* detect logarithmic brighness variation over time and output "events" with four attributes: 2D pixel address, polarity (*e.g.*, "1": brightness increase; "0": brightness decrease) and timestamp ($\sim 10\mu s$ latency). This new sensing modality has salient advantages over frame-based cameras: 1) the asynchronism of event pixels results in sub-millisecond temporal resolution, much higher than regular-speed cameras ($\sim$ 30 FPS); 2) since each pixel responds only to intensity changes, the temporal redundancy and

power consumption can be significantly reduced; 3) sensing intensity changes in logarithmic scale enlarges dynamic range to over 120 dB[1]. However, event-based cameras have increased noise level over low frame-rate cameras. And the bipolar form of output does not represent the exact temporal gradients, introducing challenges for high frame-rate video reconstruction from event-based cameras alone.

In this paper, we propose a high frame-rate video synthesis framework using a combination of regular-speed intensity frame(s) and neighboring event streams, as shown in Fig. 1. Compared to intensity-only or event-only TVFS algorithms, our work takes advantages from both ends, *i.e.*, high-speed information from events and high contrast spatial features from intensity frame(s). Our contributions are listed below:

1. We introduce a differentiable fusion model which is able to model various temporal settings. We consider three fundamental cases, *i.e.*, interpolation, prediction and motion deblur, which can serve as building blocks for other complex settings. The problem can be solved by automatic differentiation that does not involve training. We refer to this process as Differentiable Model-based Reconstruction (DMR).
2. We introduce a novel event binning strategy and compare it against conventional stacking-based binning strategy [2, 3, 34, 40]. Our binning preserves the temporal information of events necessary for high frame-rate video reconstruction. Additionally, we perform statistical evaluation for our binning strategy on the existing dataset [29].
3. We introduce a deep learning strategy for further improving the DMR results. We model the DMR artifacts as additive "noise" and perform "denoising" via deep residual learning. During training, we augment the samples by randomizing *all* the parameters of the DMR. We show preliminary results that the trained residual denoiser (RD) has properties including contrast enhancement and motion awareness, which is beyond a Gaussian denoiser.

2. Related work

2.1. Multimodal sensor fusion

Fusion among different types of sensing modalities for improved quality and functionality is an interesting topic. A related problem to ours is to spatially upsample functional sensors, *e.g.*, depth or hyperspectral sensors, with a high resolution guide image. The fusion problem can be formulated as joint image filtering via bilateral [20], multilateral filters [9] or Convolutional Neural Network (CNN) based approach [23]. For high speed video sensing, a fusion strategy can be employed between high speed video cameras (low spatial resolution) and high spatial resolution still cameras (low speed) [5, 12, 13, 37, 44].

Our paper investigates the temporal upsampling problem. While previous approaches investigate in the framework of compressive sensing [1, 14, 17, 26, 35, 38, 41], we formulate our work as fusing event streams with intensity images to obtain a temporally dense video. Compared to existing literature [36] which integrates events per pixel across time, our differentiable model utilizes "tanh" functions as event activation units and imposes sparsity constraints on both spatial and temporal domain.

2.2. Event-based image and video reconstruction

Converting event streams (binary) to multiple-valued intensity frames is a challenging task, yet has been shown beneficial to downstream visual tasks [34]. Existing strategies for image reconstruction include dictionary learning [3], manifold regularization [30], optical flow [2], exponential integration [32, 36], conditional Generative Adversarial Networks (GAN) [40] and Recurrent Neural Network (RNN) [34]. Compared to existing algorithms, our work unifies different temporal frame synthesis settings, including interpolation, extrapolation (prediction) and motion deblur (reconstructing a video from a motion-blurred image).

2.3. Non-event-based video frame synthesis

1) Interpolation: Early work on video frame interpolation has focused on establishing block-wise [10] and/or pixel-wise [21, 27] correspondences between available frames. Improved performance has been achieved via coarse-to-fine estimation [4], texture decomposition [42], and deep neural networks (DNN) [16]. Recent DNN-based approaches include deep voxel flow [25], separable convolution [31], flow computation and interpolation CNN [18]. 2) Prediction: Recent work on future frame prediction has proposed to use adversarial nets [28], temporal consistency losses [6] and layered cross convolution networks [43]. 3) Motion deblur: Recent work on resolving a sharp video/image from blurry image(s) has leveraged adversarial loss [22], gated fusion network [47], ordering-invariant loss [19], *etc*.

3. Approach

3.1. Image formation

Assume there exists a high frame-rate video denoted by tensor $\mathscr{H} \in \mathbb{R}^{h\times w\times d}$, $d > 1$[2]. The forward sensing process results in two observational tensors, *i.e.*, the intensity frame tensor $\mathscr{F}$ and event frame tensor $\mathscr{E}$. Our goal is to recover tensor $\mathscr{H}$ based on the observation of intensity and event data.

[1] Typical dynamic range of a conventional camera is 90 dB

[2] $\mathscr{H}$ is indexed on time axis starting from 1. Color channel is omitted here.

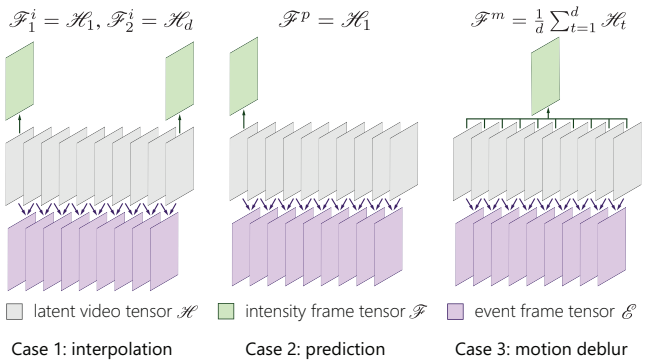


Figure 2: Forward models considered in this paper (See Section 3.1 for mathematical explanation). Case 1: interpolation from two observed intensity frames and event frames. Case 2: prediction from one observed intensity frame at the beginning and event frames. Case 3: Motion video from a single observed intensity frame and event frames.

Intensity frame tensor. We consider three sensing cases, *i.e.* 1) interpolation from the first and last frames of $\mathscr{H}$; 2) prediction based on the first frame of $\mathscr{H}$ and 3) motion deblur, in which case the intensity tensor is the summation over time. This can be visualized in Fig. 2.

Event frame tensor. As previously introduced, a pixel fires a binary output/event if the log-intensity changes beyond a threshold (positive or negative). This thresholding model can be viewed in Fig. 3a. Mathematically, the event firing process can be expressed as,

$$e_t = \begin{cases} 1 & \theta > \epsilon_p \\ -1 & \theta < -\epsilon_n \\ 0 & otherwise \end{cases}, \tag{1}$$

where $\theta = \log(I_t + b) - \log(I_0 + b)$. If $e_t = 0$, no events are generated. In order to approximate this event firing process, we model each event frame as a function of the adjacent frames from the high frame-rate tensor $\mathscr{H}$, *i.e.*,

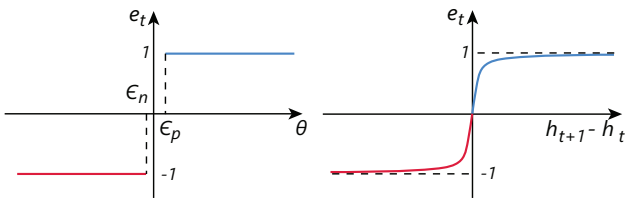
$$\mathscr{E}_t = \tanh\left\{\alpha[\mathscr{H}_{t+1} - \mathscr{H}_t]\right\}, \tag{2}$$

where α is a tuning parameter to adjust the slope of the activation curve. This function can be viewed in Fig. 3b. Based on this formulation, a video tensor with d temporal frames correspond to $d-1$ event frames.

3.2. Differentiable model-based reconstruction

The DMR is performed by minimizing a weighted combination of several loss functions. The objective function is formed as,

$$\hat{\mathscr{H}} = \underset{\mathscr{H}}{\operatorname{argmin}}\, \mathcal{L}_{pix}(\mathscr{H}, \mathcal{F}, \mathcal{E}) + \mathcal{L}_{TV}(\mathscr{H}) \tag{3}$$



(a) the event firing process (b) our proposed activation unit

Figure 3: Comparison of the event firing process and our proposed differentiable model. h_t denotes a pixel of $\mathscr{H}_t$.

Pixel loss. The pixel loss includes per-pixel difference loss against intensity and event pixels in ℓ_1 norm, *i.e.*,

$$\begin{aligned}\mathcal{L}_{pix}(\mathscr{H}, \mathcal{F}, \mathcal{E}) = \mathbb{E}_{fpix}[\|\mathcal{F} - \mathcal{A}(\mathscr{H})\|_1] \\ + \lambda_e \mathbb{E}_{epix}[\|\mathcal{E} - \mathcal{B}(\mathscr{H})\|_1],\end{aligned} \tag{4}$$

over the entire available data range. $\mathcal{F}$ and $\mathcal{E}$ denote the captured intensity and event data, respectively. $\mathcal{A}$ and $\mathcal{B}$ denote the forward sensing models described in Fig. 2 and Equation (2). $\mathbb{E}_x$ represents expectation with respect to the observed pixels/events.

Sparsity loss. We employ total variation (TV) sparsity in the spatial and temporal dimensions of the high-res tensor $\mathscr{H}$. The TV sparsity loss is defined as:

$$\mathcal{L}_{TV}(\mathscr{H}) = \lambda_{xy}\mathbb{E}_{hpix}\left[\left\|\dot{\mathscr{H}}_{xy}\right\|_1\right] + \lambda_t \mathbb{E}_{hpix}\left[\left\|\dot{\mathscr{H}}_t\right\|_1\right], \tag{5}$$

where $\dot{\mathscr{H}}_{xy} = \frac{\partial \mathscr{H}}{\partial x} + \frac{\partial \mathscr{H}}{\partial y}$ and $\dot{\mathscr{H}}_t = \frac{\partial \mathscr{H}}{\partial t}$. We later denote $\mathcal{L}_{TV_{xy}} = \mathbb{E}_{hpix}[\left\|\dot{\mathscr{H}}_{xy}\right\|_1]$ and $\mathcal{L}_{TV_t} = \mathbb{E}_{hpix}[\left\|\dot{\mathscr{H}}_t\right\|_1]$. $\mathcal{L}_{TV_{xy}}$ can be viewed as a denoising term for intensity tensor, and $\mathcal{L}_{TV_t}$ can be viewed as an event denoising term. A comparison of the performance for each loss function is shown in Fig. 4. The figure shows a synthetic case for single-frame interpolation. We use three frames, resulting in two event frames (Equation (1)). Combining the spatial and temporal TV losses resullts in better performance.

Implementation. We use stochastic gradient descent to optimize Equation (3) so as to reconstruct the latent high-res tensor. Our algorithm is implemented in TensorFlow. We use Adam optimizer. The learning rate varies depending on the tensor size as well as related parameters. Empirically, we recommend 0.002 as initial value. We recommend to schedule the learning rate to decrease $5\times$ every 200 epochs. The momenta $\beta_1 = 0.9, \beta_2 = 0.99$. For the case of interpolation, we initialize the high-res tensor $\mathscr{H}$ by linearly blending the two available low-res frames. For prediction and motion deblur, we initialize the high-res tensor using the available single low-res frame. An example of the optimization progress can be viewed in Fig. 5. As the loss decreases, both PSNR and SSIM increase and gradually converge.

(a) $\mathcal{L}_{pix}$ (b) $\mathcal{L}_{pix} + \lambda_t \mathcal{L}_{TV_t}$ (c) $\mathcal{L}_{pix} + \lambda_{xy} \mathcal{L}_{TV_{xy}}$ (d) $\mathcal{L}_{pix} + \mathcal{L}_{TV}$ (e) ground truth

Figure 4: Comparison of different loss functions (simulated single-frame interpolation). $\mathcal{L}_{TV} = \lambda_t \mathcal{L}_{TV_t} + \lambda_{xy} \mathcal{L}_{TV_{xy}}$.

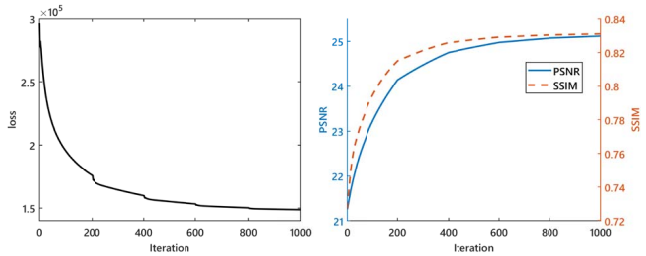


Figure 5: Loss values and accuracy (PSNR and SSIM) during DMR optimization.

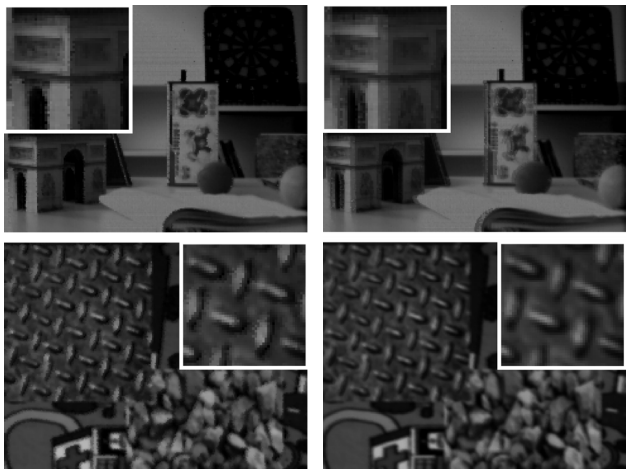
3.3. Binning events into event frames

Our event sensing model requires binning events into frames. The ideal binning strategy would be "one frame per event". However, this binning strategy is unnecessarily expensive. For example, the events between two consecutive frames (22 FPS in [29]) may vary from thousands to tens of thousands, resulting in computational challenges and redundancy. However, events happening at different locations but at very close timestamps can be processed in the same event frame. Therefore, we design and compare two binning strategies:

Binning 1 (proposed): For an incoming event, if its spatial location already has an event in the current event frame, then cast it into a new event frame; otherwise, this incoming event will stay in the current event frame. In this case, each event frame should only have three values, *i.e.*, {-1, 0, 1}.

Binning 2: Similar to several previous work [2, 3, 34, 40], where events are stacked/integrated over a time window, we allow each event frame to have more than three values. However, since the "tanh" function in Equation (2) only outputs values between -1 and 1, we modify our event sensing model to have a summation operation over several sub-event frames. Mathematically, $\mathscr{E}_{b2} = \sum_t \mathscr{E}_t$.

We show DMR results for a frame interpolation case using DAVIS dataset [29] in Fig. 6. We use two consecutive intensity frames and the events in-between. In Row 1 ("slider_depth"), 9 event frames are binned from over 7, 700 events using Binning 1. Row 2 ("simulation_3_planes") has



(a) Binning 1 (b) Binning 2

Figure 6: Comparison of two binning strategies applied to frame interpolation using the DAVIS dataset.

19 event frames from over 40, 000 events. For Binning 2, we match the sub-event frame number with Binning 1 so as to compare the performance. Frame #2 is shown. Our results show that Binning 1 preserves sharp spatial structures[3]. We will use Binning 1 in subsequent experiments.

3.4. Learning a residual denoiser

DMR is an iterative reconstruction approach based on a differentiable model, which does not involve training. The benefit of DMR is that it can handle a variety of fusion settings (interpolation, prediction, deblur, *etc.*) and is independent of optimizers. Although DMR does not involve training, it requires case-specific parameter tuning. Moreover, we observe that the DMR results may have visual artifacts. This is due to the ill-posedness of the fusion problem and different noise levels between the two sensing modalities.

In order to address these issues, we model the artifacts outcome of DMR as additive "noise" and propose a "de-

[3]A more detailed analysis and complete slow motion videos can be found in the supplementary material.

Table 1: Augmentation recipe

source	notation	value range
Eq. (1)	ϵ_p, ϵ_n	(0, 0.05)
Eq. (2)	α	(8, 20)
Eq. (4)	λ_e	(0.1, 0.5)
Eq. (5)	λ_{xy}	(0.3, 0.8)
Eq. (5)	λ_t	(0.2, 0.6)
	event percentage	(0%, 20%)
	PBR learning rate	(0.001, 0.009)
	PBR epoch(s)	(1, 350)

noising" process to remove the artifacts. Inspired by ResNet [15] and DnCNN [45], we employ the residual learning scheme and train a residual denoiser (RD). Rather than training the denoiser from various levels of artificial noise, we design to train the network from the outcome of DMR. Mathematically, the residual $\mathscr{R}$ is expressed as,

$$\mathscr{R} = \hat{\mathscr{H}} - \mathscr{H}_g, \tag{6}$$

where $\hat{\mathscr{H}}$ represents the reconstructed frame from DMR, and $\mathscr{H}_g$ represents the ground truth frame. We use a residual block similar to [46], which has a {conv + ReLU} and a {conv} layer at the beginning and end, with 17 intermediate layers of {conv + BN + ReLU}. The kernel size is 3×3 with stride of 1. The loss function for our denoiser is the mean squared error of $\hat{\mathscr{H}}$ and $\mathscr{R}$. During training, we augment data by randomizing the configuration parameters (including the running epochs) in DMR, summarized in Table 1. The goal of this augmentation is 1) to prevent overfitting; 2) to enforce learning of our DMR process; 3) to alleviate effects due to non-optimal parameter tunning. Our denoiser is single-frame, as we seek to enhance each DMR output frame iteratively *without* comprimising the variety of DMR fusion settings.

4. Experiment results

We design several experiments to show the effectiveness of our framework.

- For DMR, we evaluate the three cases (interpolation, prediction and motion deblur) described in Fig. 2 on the DAVIS dataset [29], and compare against state-of-the-art event-based algorithms, *i.e.*, Complementary Filter [36] and Event-based Double Integral [32].
- For RD, we first discuss how to use the trained RD. We compare two strategies, *i.e.*, 1) to use RD after every DMR iteration; 2) to use RD only when DMR is converged. We then evaluate the effectiveness of trained RD by comparing with Gaussian denoisers, *e.g.*, DnCNN [45] and FFDNet [46].
- Finally, we compare our results with a non-event-based frame interpolation algorithm, SepConv [31].

4.1. Results for DMR

Interpolation. We first show interpolation results in Fig. 7. We use three consecutive frames from [29], withholding the middle frame. The intermediate events bin into 20 event frames. The ground truth middle frame is the closest to Frame #10.

Prediction. We next show frame prediction results, corresponding to Case 2 in Fig. 2. We withhold the end frame of two consecutive frames and seek to predict it using the start frame and "future" events. The results are shown in Fig. 8. Compared to CF [36], our results are less noisy and closer to the ground truth.

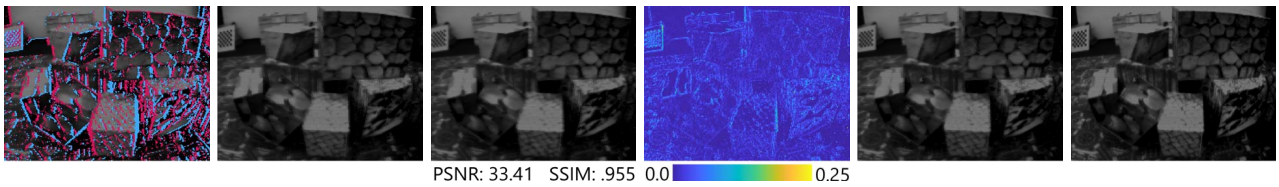
Motion deblur. Corresponding to Case 3 in Fig. 2, we compare our DMR results with state-of-the-art, Event-based Double Integral (EDI) [32], shown in Fig. 9. Compared to EDI, our results preserves sharp edges while alleviating event noise.

4.2. Results for RD

Data preparation. We use publicly available high-speed (240 FPS) video dataset, the Need for Speed dataset [11]. The reason we choose this dataset is because it has rich motion categories and content (100 videos with 380K frames) which involves both camera and scene/object motion. As introduced in Section 3.4, our RD is trained on the output of DMR process. As a proof of concept, we simulate solving a single-frame prediction problem, *i.e.* given two consecutive video frames, we first simulate the latent event frame. Next, a DMR is performed to predict the end frame.

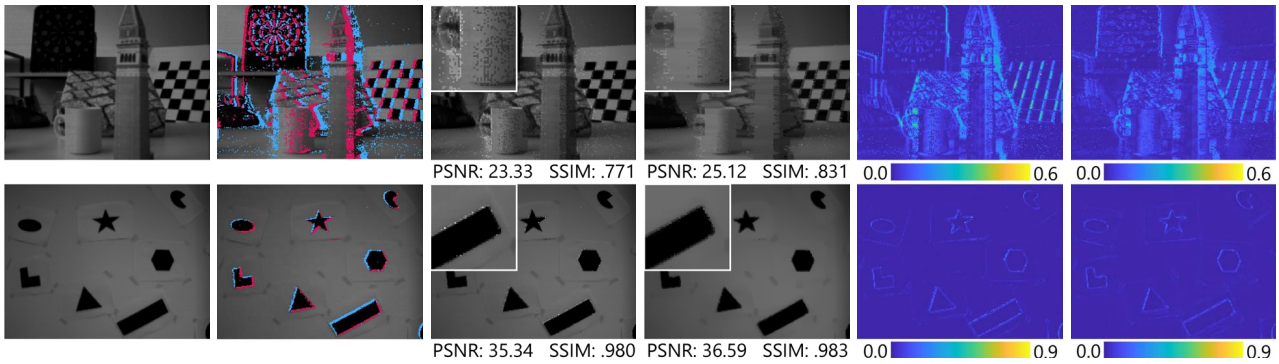
Training and testing. We randomly split the dataset into 89 training classes and 11 testing classes. For augmentation purpose, we perform a random temporal flip and a spatial crop with size 40×40. The sample clip will then experience event frame simulaltion and DMR using a random setting according to Table 1. Note that we enforce generated event frames to contain less than 20% of events. This is according to a statistical analysis of the DAVIS dataset[4]. We generate 100K image pairs of size 40×40 pixels; 80% of the sample dataset are randomly chosen as training samples and the rest 20% are used for validation. We use a batch size of 128, which results in 2K batches per epoch. We use mini-batch stochastic gradient descent with an Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$). Note that the same optimizer has been used in the DMR process, but the optimization of DMR does not involve training. The learning rate is scheduled as 1×10^{-3} for the initial 30 epochs, then 1×10^{-4} for the following 30 epochs and 5×10^{-5} afterwards. We use an NVIDIA TITAN X GPU for parallelization. Each epoch takes approximately 6 minutes training on our machine. We

[4] A statistical analysis is included in the supplementary material



(a) Frame #1 + events (b) Frame #5 (c) Frame #10 (d) Error map #10 (e) Frame #16 (f) Frame #21

Figure 7: Frame interpolation. The start and end frames, as well as in-between events, are used as input. Frame #10 is compared against the ground truth middle frame.



(a) Start frame (b) w/ future events (c) CF [36] (d) Ours (DMR) (e) Error map of CF (f) Error map of ours

Figure 8: Frame prediction. Given a start frame (a) and the future events (b) happened after (a), we predict the end frame (ground truth omitted). Our results using DMR alone outperforms existing algorithm, Complementary Filters (CF) [36].

train our network for 150 epochs. Since our model is fully convolutional, the number of parameters is independent of the image size. This enables us to train on small patches (40×40) and test on the whole image.

Plug & play *vs.* one-time denoising. Since we train our denoiser to establish a mapping function between DMR and its residual towards the ground truth, the first experiment we investigated is how/when to use this denoiser. We compare two frameworks, *i.e.*, the plug & play [39] and the one-time denoising. The plug & play framework decouples the forward physical model and the denoising prior using the ADMM technique [7]. For one time denoising, we apply the residual denoiser once after the DMR has converged. One-time denoising is considered because it is considerably faster than plug & play. Our experimental results show that one-time denoising performs similar or even better than plug & play, shown in Table 2. We reason that this is related to our training process and the initialization of the high-res tensor. Our differentiable model involves a temporal transition process from an existing frame to a future frame. We initialize the high-res tensor with the reference intensity frame. In each DMR iteration, the reconstruction process produces artifacts that are similar to the degradations in the initialized image. However, our denoiser is trained to "recognize" this degradation and remove these artifacts. Therefore, our denoiser is most useful and efficient when applied after the DMR has converged[5].

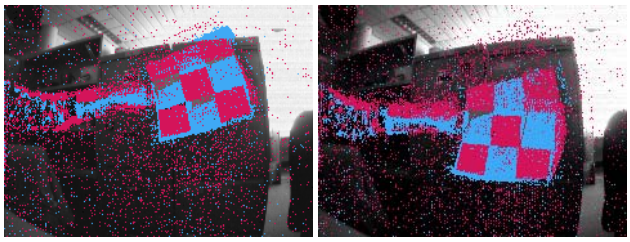
Table 2: Plug & play *vs.* one-time denoising using RD.

clip name	plug &play	one-time
Motorcycle	28.07 / .951	**29.11/.965**
Car race	24.53 / .883	**24.89/.895**
Football Player	29.94 / .935	**32.30/.978**

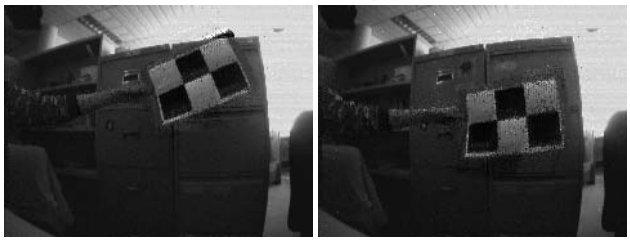
Comparison with Gaussian denoisers. Since we decouple the problem as DMR and RD process, it is interesting to see whether a general denoiser can complete this task. We select several video clips from the testing classes and compare our results with two other denoisers, DnCNN [45] and FFDNet [46]. DnCNN is an end-to-end trainable deep CNN for image denoising with different Gaussian noise levels, *e.g.*, [0, 55]. During our testing of DnCNN we found that the pre-trained weights do not perform well. We re-trained the network using the Need for Speed dataset with Gaussian noise. The FFDNet is a later variant of DnCNN with the inclusion of pre- and post-processing. During our tuning of the FFDNet, we found that smaller noise levels (a tunable parameter for using the model) result in better denoising performance in terms of PSNR and SSIM metrics. For each testing image, we present the best tuned FFDNet result (noise level less than 10) and compare with our pro-

[5]Visual results are included in supplementary material.

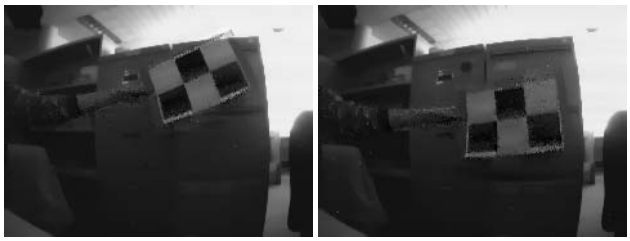
(a) Blurry images (data from [32])



(b) w/ events during exposure time



(c) EDI [32]



(d) Ours (DMR)

Figure 9: Motion deblur. A motion blurred image (a) and the events during exposure time (b) are used to reconstruct a high frame-rate video. Compare to (c) EDI [32], our results (d) preserves spatial features with less noise.

posed denoiser. The results are summarized in Table 3. Partial results[6] with zoom-in figures are presented in Fig. 10.

4.3. Comparison to non-event-based approach

We compared our results for performing multi-frame interpolation with a state-of-the-art approach, SepConv [31]. We present results comparing 3-frame interpolation in Fig. 11. We convert our grayscale testing images to 3 channels (RGB) before applying the SepConv interpolation algorithm. Although the results from SepConv provide better visual experience, they have salient artifacts around large motion regions. Note that performing intensity only frame interpolation produces significant artifacts in the presence of severe occlusions. On the other hand, our event-driven frame interpolation is able to successfully recover image details in occluded regions of interpolated frames[7]. For a quantitive comparison, the SepConv method has an average SSIM of 0.9566 and PSNR of 29.79. Ours have average SSIM of 0.9741 and PSNR of 37.64.

[6]Full results can be seen in the supplementary material.

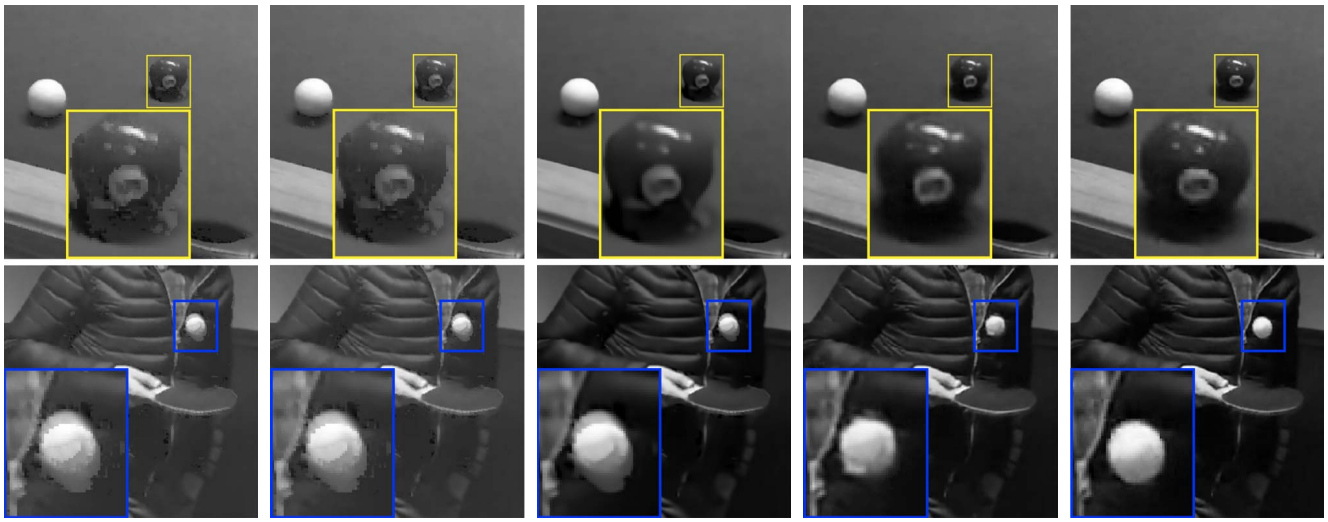
Table 3: Performance comparison for different denoisers.

clip name	metric	DMR	DnCNN	FFDNet	Ours
airplane	PSNR	30.91	31.10	30.92	**31.38**
	SSIM	0.975	0.982	0.976	**0.982**
basketball	PSNR	23.55	24.05	23.47	**24.06**
	SSIM	0.963	0.971	0.964	**0.972**
soccer	PSNR	29.96	31.08	30.13	**31.29**
	SSIM	0.961	0.974	0.962	**0.975**
billiard	PSNR	36.46	35.42	**36.48**	36.46
	SSIM	0.982	0.986	0.983	**0.987**
ping pong	PSNR	32.46	32.26	**32.50**	32.24
	SSIM	0.974	0.978	0.975	**0.979**

5. Concluding remarks

In this paper, we have introduced a novel high frame-rate video synthesis framework by fusing intensity frames with event streams, taking advantages from both ends. Our framework includes two key steps, *i.e.*, DMR and RD. Our DMR is free of training and is capable to unify different fusion settings between the two sensing modalities, which was not considered in previous work such as [32, 36]. We have shown in real data that our DMR performs better than existing algorithms. However, DMR requires tuning parameters, which have large variance across various settings. This was one of the reasons we propose to train an RD. Our strategy is to incorporate a range of DMR parameter settings so as to expose the network with various DMR results, including both the optimal and non-optimal ones. By learning the corresponding residual, our simulation results have shown that a RD can be trained to effectively remove artifacts from DMR. Currently we train an RD from single-frame prediction case. Yet it is interesting to further augment the training samples with *all* the cases, which we will investigate in the future. Applying the RD to real data faces a domain gap due to the resolution (both spatial and temporal) and noise level mismatch. Currently, none of the existing DAVIS datasets contains enough sharp intensity images captured at high speed for training/fine-tuning. We will investigate event simulation using event simulator [33] in our future work.

[7]Please see videos of results in supplementary material.



(a) Ours (DMR) (b) DnCNN [45] (c) FFDNet [46] (d) Ours (RD) (e) ground truth

Figure 10: Comparison of denoising performance. Our learned Residual Denoiser (RD) reconstructs the intermediate frame (1-frame interpolation case) with fewer motion artifacts.

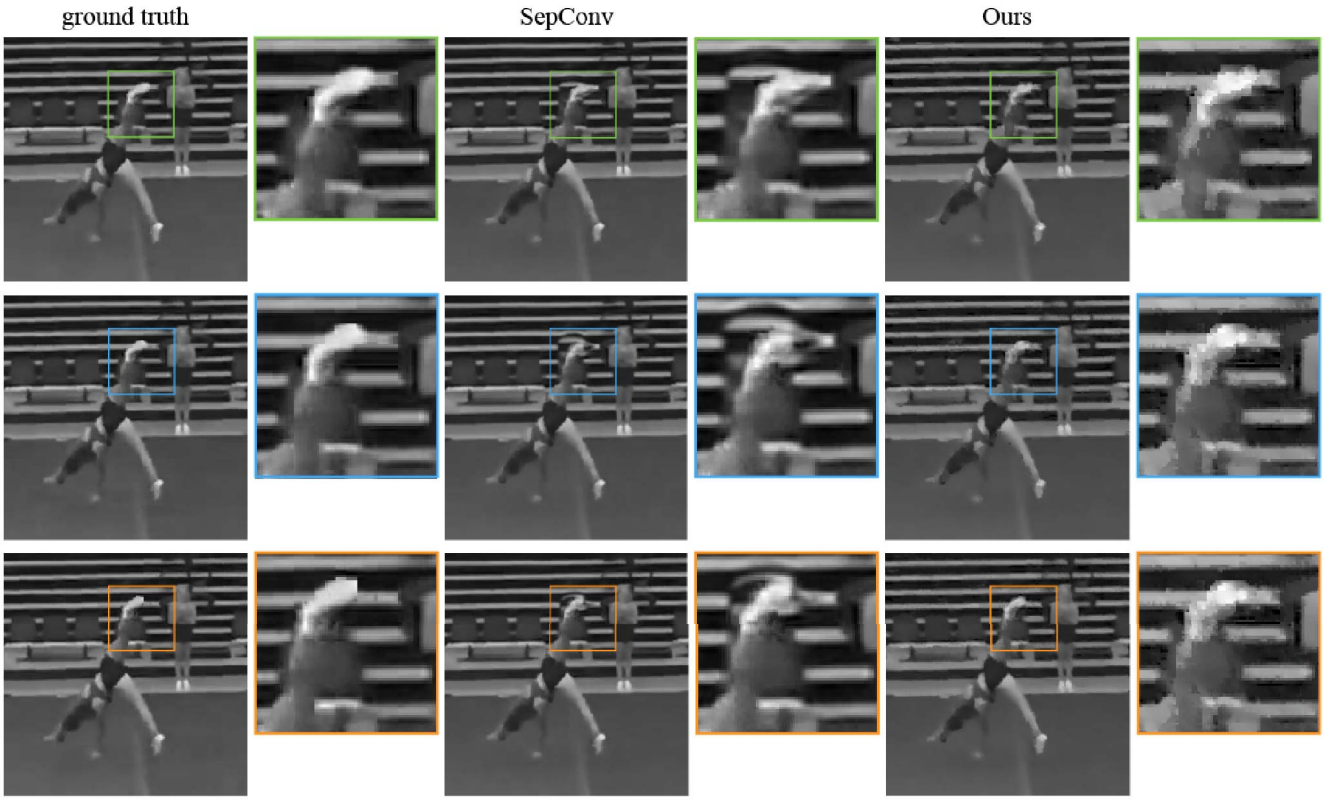


Figure 11: Multi-frame interpolation results, compared with SepConv [31]. Shown are frames #2, #3 and #4. Note that the intensity-only based frame interpolation method (SepConv) produces considerable motion artifacts around occluded areas, while our event-driven frame interpolation succesfully recovers image details in occluded regions.

6. Acknowledgement

This work was supported in part by a DARPA Contract No. HR0011-17-2-0044.

References

[1] R. G. Baraniuk, T. Goldstein, A. C. Sankaranarayanan, C. Studer, A. Veeraraghavan, and M. B. Wakin. Compressive video sensing: algorithms, architectures, and applications. *IEEE Signal Processing Magazine*, 34(1):52–66, 2017. 2

[2] P. Bardow, A. J. Davison, and S. Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016. 2, 4

[3] S. Barua, Y. Miyatani, and A. Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *Proc. of the Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016. 2, 4

[4] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 237–252. Springer, 1992. 2

[5] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, M. Cohen, B. Curless, and S. B. Kang. Using photographs to enhance videos of a static scene. In *Proc. of the 18th Eurographics conference on Rendering Techniques*, pages 327–338. Eurographics Association, 2007. 2

[6] P. Bhattacharjee and S. Das. Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 4271–4280, 2017. 2

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning*, 3(1):1–122, 2011. 6

[8] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240$\times$ 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1

[9] D. Chan, H. Buisman, C. Theobalt, and S. Thrun. A noise-aware filter for real-time depth upsampling. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*, 2008. 2

[10] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(4):407–416, 2007. 2

[11] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 1134–1143, 2017. 5

[12] A. Gupta, P. Bhat, M. Dontcheva, O. Deussen, B. Curless, and M. Cohen. Enhancing and experiencing spacetime resolution with videos and stills. In *Proc. of the IEEE International Conference on Computational Photography (ICCP)*, pages 1–9, 2009. 2

[13] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan. Flexible voxels for motion-aware videography. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 100–114. Springer, 2010. 2

[14] K. He, Z. Wang, X. Huang, X. Wang, S. Yoo, P. Ruiz, I. Gdor, A. Selewa, N. J. Ferrier, N. Scherer, et al. Computational multifocal microscopy. *Biomedical Optics Express*, 9(12):6477–6496, 2018. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[16] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 2

[17] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, 72:9–18, 2018. 2

[18] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018. 2

[19] M. Jin, G. Meishvili, and P. Favaro. Learning to extract a video sequence from a single motion-blurred image. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6334–6342, 2018. 2

[20] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*, volume 26, page 96. ACM, 2007. 2

[21] R. Krishnamurthy, J. W. Woods, and P. Moulin. Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields. *IEEE transactions on circuits and systems for video technology*, 9(5):713–726, 1999. 2

[22] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8183–8192, 2018. 2

[23] Y. Li, J.-B. Huang, A. Narendra, and M.-H. Yang. Deep joint image filtering. In *Proc. of the European Conference on Computer Vision (ECCV)*. Springer, 2016. 2

[24] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In *IEEE International Solid-State Circuits Conference*, pages 2060–2069, 2006. 1

[25] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2017. 2

[26] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady. Coded aperture compressive temporal imaging. *Optics Express*, 21(9):10526–10545, 2013. 2

[27] M. Luessi and A. K. Katsaggelos. Efficient motion compensated frame rate upconversion using multiple interpolations and median filtering. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, pages 373–376, 2009. 2

[28] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 2

[29] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 2, 4, 5

[30] G. Munda, C. Reinbacher, and T. Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, 126(12):1381–1393, 2018. 2

[31] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 261–270, 2017. 2, 5, 7, 8

[32] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5, 7

[33] H. Rebecq, D. Gehrig, and D. Scaramuzza. ESIM: an open event camera simulator. *Conf. on Robotics Learning (CoRL)*, Oct. 2018. 7

[34] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3857–3866, 2019. 2, 4

[35] D. Reddy, A. Veeraraghavan, and R. Chellappa. P2c2: Programmable pixel compressive camera for high speed imaging. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 329–336, 2011. 2

[36] C. Scheerlinck, N. Barnes, and R. Mahony. Continuous-time intensity estimation using event cameras. In *Proc. of the Asian Conference on Computer Vision (ACCV)*, December 2018. 2, 5, 6, 7

[37] F. Schubert and K. Mikolajczyk. Combining high-resolution images with low-quality videos. In *Proc. of the British Machine Vision Conference (BMVC)*, pages 1–10, 2008. 2

[38] V. Stanković, L. Stanković, and S. Cheng. Compressive video sampling. In *European Signal Processing Conference*, pages 1–5. IEEE, 2008. 2

[39] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *Global Conference on Signal and Information Processing (GlobalSIP)*, pages 945–948. IEEE, 2013. 6

[40] L. Wang, Y.-S. Ho, K.-J. Yoon, et al. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[41] Z. Wang, L. Spinoulas, K. He, L. Tian, O. Cossairt, A. K. Katsaggelos, and H. Chen. Compressive holographic video. *Optics Express*, 25(1):250–262, 2017. 2

[42] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality tv-l1 flow with fundamental matrix prior. In *Proc. of the IEEE 23rd International Conference on Image and Vision Computing*, pages 1–6, 2008. 2

[43] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems 29*, pages 91–99. Curran Associates, Inc., 2016. 2

[44] H. Zabrodsky and S. Peleg. Attentive transmission. *Journal of Visual Communication and Image Representation*, 1(2):189–198, 1990. 2

[45] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 5, 6, 8

[46] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 5, 6, 8

[47] X. Zhang, H. Dong, Z. Hu, W.-S. Lai, F. Wang, and M.-H. Yang. Gated fusion network for joint image deblurring and super-resolution. In *Proc. of the British Machine Vision Conference (BMVC)*, 2018. 2