EFI-Net: Video Frame Interpolation from Fusion of Events and Frames

Genady Paikin Yotam Ater Roy Shaul Evgeny Soloveichik
Samsung Israel R&D Center
Tel Aviv
{genady.p, yotam.ater, roy.s, evgeny.s}@samsung.com

Abstract

Event cameras are sensors with pixels that respond independently and asynchronously to changes in scene illumination. Event cameras have a number of advantages when compared to conventional cameras: low-latency, high temporal resolution, high dynamic range, low power and sparse data output. However, existing event cameras also suffer from comparatively low spatial resolution and are sensitive to noise. Recently, it has been shown that it is possible to reconstruct an intensity frame stream from an event stream. These reconstructions preserve the high temporal rate of the event stream, but tend to suffer from significant artifacts and low image quality due to the shortcomings of event cameras. In this work we demonstrate that it is possible to combine the best of both worlds, by fusing a color frame stream at low temporal resolution and high spatial resolution with an event stream at high temporal resolution and low spatial resolution to generate a video stream with both high temporal and spatial resolutions while preserving the original color information. We utilize a novel event frame interpolation network (EFI-Net), a multi-phase convolutional neural network which fuses the frame and event streams. EFI-Net is trained using only simulated data and generalizes exceptionally well to real-world experimental data. We show that our method is able to interpolate frames where traditional video interpolation approaches fail, while also outperforming event-only reconstructions. We further contribute a new dataset, containing event camera data synchronized with high speed video. This work opens the door to a new application for event cameras, enabling high fidelity fusion with frame based image streams for generation of high-quality high-speed video. The dataset is available at https://drive.google.com/file/d/1UIGVBqNER_5KguYPAu5y7TVg-JlNhz3-/view?usp=sharing

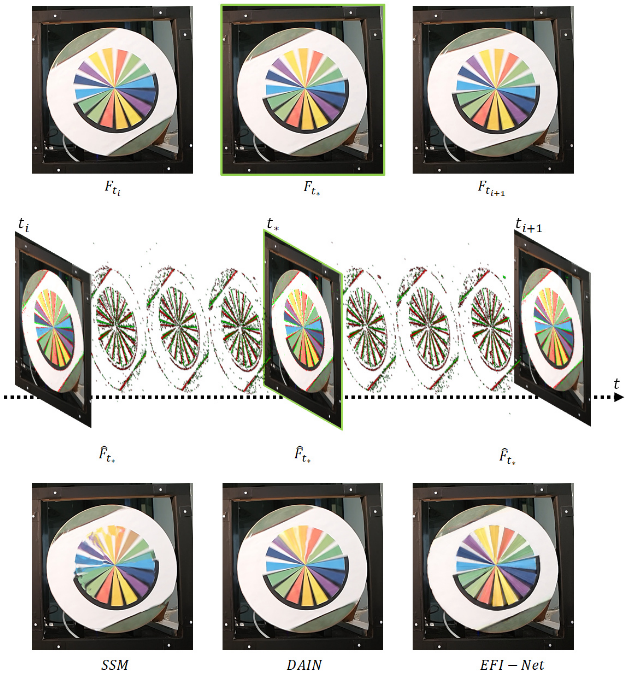

Figure 1. Comparison of our proposed method with state-of-the-art VFI methods: Super Slomo (SSM) [22] and DAIN [2]. Top row: frames captured using a high speed video camera. Mid row: illustration of frames and intermediate events grouped into temporal bins. Bottom row: Interpolated frame at time t^*, using key frames t_i and t_{i+1} synthesized by SSM (left), DAIN (center) and our proposed method (right).

1. Introduction

Event cameras are a novel class of sensors, with pixels which respond asynchronously to changes in illumination. The output of an event camera, an event stream, is a sequence of datums which contain information about the timing of the event, the polarity of the illumination change and the spatial location of the source pixel. Event cameras, such as the Dynamic Vision Sensor (DVS) which was first introduced by [31], often have a dynamic range in excess of 100 dB, temporal time-stamp resolutions on the order of 10us and sub-millisecond latency [57, 11, 50].

These properties have made them promising candidates for challenging machine vision tasks. These include, object detection [40, 29], gesture recognition [1, 49], feature tracking [13, 28, 71], visual odometry and SLAM [27, 53, 74, 52, 63, 73], optical flow [73, 4, 3, 60, 72, 60, 47, 32] and video deblurring [23, 67]. Recent works have shown that it is possible to reconstruct intensity images from the event stream [54, 64, 42, 3, 27, 55]. The most recent works [7, 65], also performed super resolution in order to enhance the comparatively low spatial resolution of event cameras. These works have demonstrated that the event stream contains visual information necessary for intensity frame reconstruction, however they display significant artifacts. Common artifacts include accumulated noise, blurriness and inconsistent video frames.

Classical video frame interpolation (VFI) is a well-known problem in the field of video processing. The goal of VFI is to synthesize non-existent frames, by interpolating over a set of sequential frames in the video stream. A typical VFI application is frame rate upconversion [5, 6, 21, 25, 30]. Other applications include frame recovery in video coding and streaming [17, 18], slow motion effects [22] and novel view synthesis [10, 59]. Conventional approaches to VFI typically consist of the following steps: bi-directional motion estimation, motion interpolation and occlusion reasoning, and motion-compensated frame interpolation. Such approaches are prone to various artifacts, such as halos, ghosts and break-ups due to insufficient quality of any of the components mentioned above.

Deep learning, and specifically convolutional neural networks (CNNs) have emerged as the leading method for numerous image processing and computer vision tasks. Many works [22, 33, 35, 37, 38, 43, 44, 62, 68, 51, 45] have attempted to replace all or some of the steps in the VFI algorithmic flow with CNNs. Despite the significant progress achieved by recent CNN based methods, existing approaches are still limited in their performance, with strong motions and wide occlusions being particular weaknesses. Recently, [45] has suggested alleviating some of these issues by performing fusion of two video streams as input: a primary video stream with high spatial resolution and an auxiliary video stream with high frame rate and low spatial resolution. In this work, it was shown that the auxiliary video stream allowed the CNN to reduce artifacts caused by temporal aliasing. In an analogous manner to [45], the event stream can provide information at a high temporal resolution and relatively low bit rate.

In this paper we propose a novel method for sensor fusion between conventional cameras and event cameras, which capitalizes on the best properties of each of the two sensors. Our method generates videos with the spatial resolution of the conventional camera and the temporal resolution of the event camera. An example of our interpolation algorithm working on experimental data is seen in Fig. 1.

This work differs from previous approaches which utilized event cameras to generate images [54, 64, 42, 3, 27, 55] by virtue of the fusion we perform with a conventional frame based sensor. Previous works [58, 48, 46] performed fusion between events and frames, but they did not addressed the problems of spatial resolution mismatch, temporal resolution mismatch and preservation of color. To the best of our knowledge, ours is the first work which systematically resolves all of these common disparities between conventional and event based sensors. Likewise, our work differs from previous VFI works by the addition of high temporal resolution data from an event camera.

Our algorithm is a full CNN solution, performed in three phases. Phase I fuses the data from the intensity images and the event stream into an intensity estimate of the interpolated frame at the spatial resolution of the event camera. Phase II up-scales the output of Phase I to the spatial resolution of the conventional camera. Finally, Phase III colorizes the intensity image to output a final color frame. The way that we control the output allows us to interpolate any intermediate frame up to the time resolution of the event camera. Our network is trained using simulated event data that is generated from affine transforms of still images. This allows us to generate large amounts of data, and our experimental results show that our method generalizes well to real data. We validate our results on an experimental dataset consisting of high frame rate video spatio-temporally aligned with the output of an event camera, which allows us to rigorously test our method against high-quality ground truth.

Our primary contributions are:

- A novel method of performing VFI utilizing an event-camera to reduce aliasing and strong motion artifacts.
- A novel three phase CNN architecture that fuses a conventional frame stream with the output of an event camera, while simultaneously fusing data with spatial and temporal resolution differences and preserving the color information.
- A dataset of event camera and high frame rate video synchronized in time and calibrated in space.

2. Related Works

2.1. Intensity Reconstruction from Events

The first works on image reconstruction from event cameras were made by [8, 26] where a scene mosaic was reconstructed. Next [27], used a probabilistic filtering formulation to jointly estimate scene structure and perform gray-

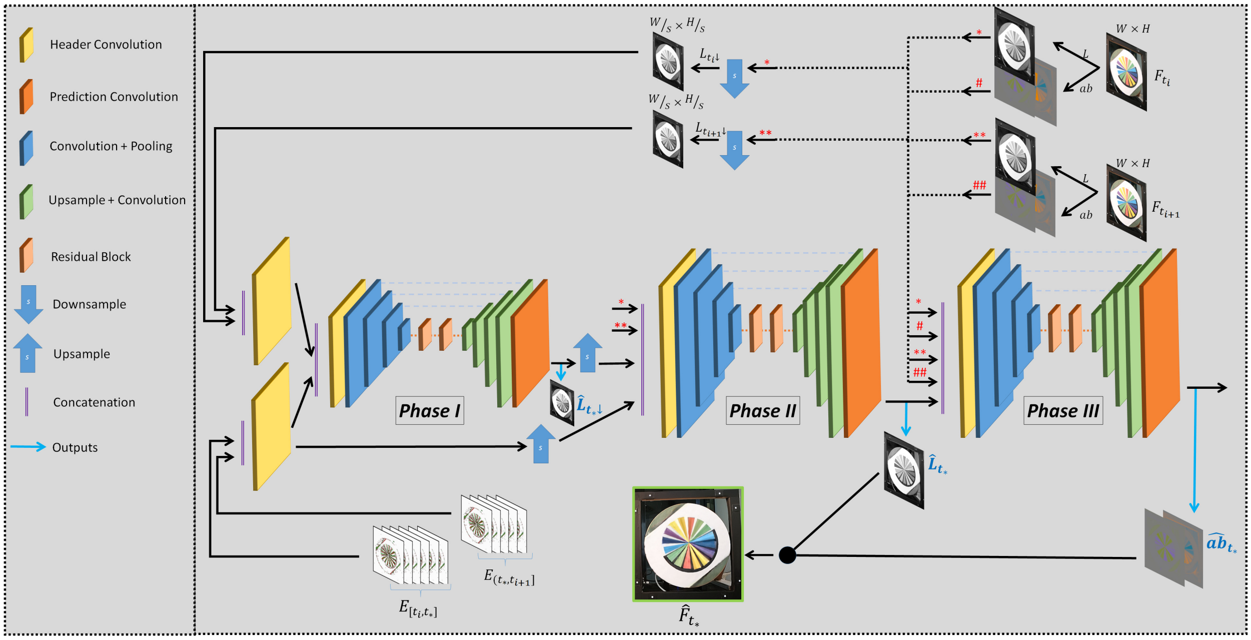

Figure 2. An illustration of the proposed EFI-Net data flow and architecture. The network consists of three phases: low resolution intensity interpolation (Phase I), high resolution intensity interpolation (Phase II), and re-colorization (Phase III).

scale intensity reconstruction. [3] proposed to estimate optical flow and intensity changes simultaneously by minimizing a variational energy functional. Similarly, [42] regarded image reconstruction as an energy minimization problem defined on manifolds induced by event timestamps.

Recently, the application of deep learning based neural networks have made much progress in intensity image and video reconstruction. First, [54] exploited recurrent neural networks to reconstruct video, followed by [64] which used a generative adversarial network to perform high resolution reconstruction. [65] suggested a three step pipeline: reconstruction of low resolution images from event streams, image quality enhancement, and up-sampling of the enhanced images. [7] reconstructed high resolution images by taking a sequence of event stacks near the timestamp of interest as input, and then used a recurrent neural network to generate a super-resolved image, followed by a mixer network for final reconstruction.

Works [58, 48, 46] demonstrated methods of fusion between event cameras and traditional cameras. [58] utilized an asynchronous complementary filter to perform real-time intensity image reconstruction using either cameras fusion or solely event data and is able to generate frames at the rate of the events. Papers [48, 46] utilized an optimization based framework to deblur conventional frames using event data, [46] also showed that this method can be used for high frame rate video generation. None of these fusion methods address the challenges of resolution mismatch and colorization when the event camera differs from the conventional camera.

2.2. VFI with CNNs

Inspired by the success of CNNs, the early works on CNN based VFI [35] and video frame prediction [37] adopted an approach of direct estimation of pixels. However, this typically led to blurred outputs and unsatisfactory image quality. To overcome the weaknesses of these initial attempts, later approaches suggested more structured neural networks. In the AdaConv [43] and SepConv [44] methods CNNs are used to estimate adaptive filters for every pair of corresponding patches in consecutive input frames. Other works revisited the classical VFI algorithmic flow and focused on replacing some of its steps by one or more CNNs such as in [34, 62, 69]. Recent publications [22, 43] demonstrate the ability to produce arbitrarily interpolated intermediate frames by using bi-directional motion estimation, and refinement steps such as: motion estimation refinement, occlusion reasoning, and frame synthesis. Other recent works suggested utilizing a per-pixel phase-based motion representation for VFI [38, 39]. In [15], a method which uses structure-guided interpolation and texture refinement was shown. [45] argue that auxiliary data contributes to VFI. They demonstrated a hybrid system with an auxiliary high frame rate, low spatial resolution video camera in addition to the main low frame rate, high spatial resolution camera. Our work also introduces an auxiliary high temporal resolution source, however, the sparsity of the event stream utilizes less bandwidth for any given sampling frequency.

3. Proposed Method

3.1. Input Data

Our goal is to synthesize an accurate interpolated color frame at any time in-between two sequential video key frames by utilizing auxiliary information from the event stream. We denote the two sequential key frames as F_{t_i} and $F_{t_{i+1}}$ respectively captured at instants t_i and t_{i+1}. The frame we want to synthesize at time $t^* \in (t_i, t_{i+1})$ is denoted $\hat{F_{t^*}}$. Finally, we denote the set of all events which occurred between two arbitrary points in time $t_a, t_b \in [t_i, t_{i+1}]$ as $\mathbf{e}_{[t_a,t_b]}$. We process the input data using a three phase CNN pipeline which we denote "EFI-Net". Our pipeline is described in Fig. 2 and detailed in the following sections. The supplementary material describes the full architecture and training details of the CNN.

3.1.1 Event Tensor

Given the set $\mathbf{e}$ of N input events $\{(x_j, y_j, t_j, p_j) | j \in [0, N-1]\}$ between key frames F_{t_i} and $F_{t_{i+1}}$, where x, y are pixel coordinates, t is the timestamp of the event, and $p = \pm 1$ is the polarity indicating the sign of brightness change. We partition the set $\mathbf{e}$ by time and polarity into the subsets $\mathbf{e}^+_{[t_i,t^*]}$, $\mathbf{e}^+_{(t^*,t_{i+1}]}$, $\mathbf{e}^-_{[t_i,t^*]}$ and $\mathbf{e}^-_{(t^*,t_{i+1}]}$. For example $\mathbf{e}^+_{[t_i,t^*]}$ is the subset of events with positive polarity and $t_j \in [t_i, t^*]$. Each of these subsets is used to construct a tensor with fixed dimensions $W \times H \times B$, where W and H are the spatial dimensions of the event camera and B is a parameter of used to determine the number of temporal bins. Note that B is constant for all tensors. We construct these tensors $\mathbf{E}^+_{[t_i,t^*]}$, $\mathbf{E}^+_{(t^*,t_{i+1}]}$, $\mathbf{E}^-_{[t_i,t^*]}$ and $\mathbf{E}^-_{(t^*,t_{i+1}]}$ as spatio-temporal voxel grids, similarly to [64, 73, 54]. Recall that each event contributes to the two temporally closest bins proportionally to its distance in time from their centers. The tensor are concatenated to form input event tensor $\mathbf{E}$ of dimensions $W \times H \times 4B$

3.1.2 Key Frame Tensor

The event stream only adds information to the intensity component of the image. For this reason, we address the intensity and color channels of the key frames separately. We first convert the key frames from the RGB color space to the CIELAB color space [36], commonly used in colorization networks [19, 16, 20, 70]. We denote the intensity and color channels of the CIELAB color space as L and ab. Each of the key frames F_t are separated into their components as $F_t \rightarrow L_t, ab_t$. For Phase I we additionally downsample the L_t components to the resolution of the event camera, denoted as $L_{t\downarrow}$.

3.1.3 Training Data

To train the network a large amount of groundtruth (GT) data is required. However, collection of a suitably large dataset containing both event data and high-speed GT frames is challenging. While previous works [9, 12] have addressed this challenge by devising methods to generate simulated events from video sequences, we opted for a simpler approach. We take a single image as input, from which we generate a sequence of images $F_1, F_2, ..., F_N$, by applying small random tranformations: translation, resize and rotation. $L_{1\downarrow}, L_{2\downarrow}, ..., L_{N\downarrow}$ are generated from the sequence of images and are the downscaled intensity components of the images. Simulated event data is generated by the following equation:

$$E = \begin{cases} 1 & \frac{L_{n+1\downarrow}+c}{L_{n\downarrow}+c} > \tau \\ -1 & \frac{L_{n+1\downarrow}+c}{L_{n\downarrow}+c} < 1/\tau \\ 0 & else \end{cases} \tag{1}$$

Where c is a positive constant to avoid excessive noise in dark regions, and τ is a threshold simulating the sensitivity of the event camera. τ is randomized during training to simulate varying levels of sensitivity and noise. The use of ratios mimics the logarithmic nature of the event camera. This is the only type of data we use in training our network.

3.2. EFI-Net

3.2.1 Network Architecture

As described in Fig. 2 our basic block for all three phases is a U-Net with skip connections [56]. In each phase, all layers except the prediction layer are followed by a ReLU activation. $\hat{F_{t^*}}$, the final output of the network is taken by combining the outputs of Phase II and Phase III.

3.2.2 Phase I

Phase I estimates an event camera resolution luminance image $\hat{L}_{t^*\downarrow}$ from inputs $E_{[t_i,t^*]}$, $E_{(t^*,t_{i+1}]}$, $L_{t_i\downarrow}$ and $L_{t_{t+1}\downarrow}$. The L and E input tensors are passed through separate header layers to extend the number of channels and then concatenated to a single extended tensor. The prediction layer of this phase is followed by a sigmoid activation.

3.2.3 Phase II

Phase II estimates the full resolution luminance component $\hat{L}_{t^*}$ from inputs L_{t_i}, $L_{t_{i+1}}$, $\hat{L}_{t^*\downarrow}$, and a generalized tensor $E_{Phase\ I}$ which is the output of the event header convolution of Phase I. The inputs $\hat{L}_{t^*\downarrow}$ and $E_{Phase\ I}$ are first spatially upsampled using bi-linear interpolation and then concatenated channel-wise with L_{t_i}, $L_{t_{i+1}}$. The prediction layer of this phase is followed by a sigmoid activation.

3.2.4 Phase III

Phase III estimates the full resolution color components $\hat{ab}_{t^*}$ from inputs L_{t_i}, $L_{t_{i+1}}$, ab_{t_i}, $ab_{t_{i+1}}$ and $\hat{L}_{t^*}$. Additionally, we calculate an optical flow prior [61] for the ab components by using optical flow interpolation, as suggested by [45]. The prediction layer is followed by a hyperbolic-tangent activation to approximate the natural non-linearity of the ab channels.

3.3. Training and Loss Functions

We start by training only Phase I and Phase II and after several epochs we add Phase III to the training. For Phase I and Phase II three losses are applied: for perceptual reconstruction we apply a perceptual loss based on VGG19 (ϕ) similar to [24],

$$\mathcal{L}_P = ||\phi(\hat{L_j}) - \phi(L_j)||_2^2 \tag{2}$$

an $\mathcal{L}_1$ loss to refine the intensity level,

$$\mathcal{L}_I = ||\hat{L_j} - (L_j)||_1 \tag{3}$$

and gradient maximization loss to encourage deblurring by the event stream,

$$\mathcal{L}_G = -(||\nabla_x(\hat{L}_j)| + |\nabla_y(\hat{L}_j)||_1) \tag{4}$$

where ∇ is a directional gradient. For Phase III we use smooth $\mathcal{L}_1$ loss as proposed by [14],

$$\mathcal{L}_C = \mathcal{L}_{1smooth}(\hat{ab}_j - ab_j). \tag{5}$$

Finally, to encourage temporal consistency we use,

$$\mathcal{L}_S = ||(L_{j+1} - L_j) - (\hat{L}_{j+1} - \hat{L}_j)||_1. \tag{6}$$

This loss encourages the overall VFI output to have a smoother and more natural appearance. For all equations above j is an arbitrary frame in the sequence. We combine all losses together for every key frame pair and a random number of interpolated frames at random intermediate times.

$$\begin{aligned}\mathcal{L}_{total} = \lambda_P(\mathcal{L}_P \downarrow + \mathcal{L}_P) + \lambda_I(\mathcal{L}_I \downarrow + \mathcal{L}_I) + \\ \lambda_G(\mathcal{L}_G \downarrow + \mathcal{L}_G) + \lambda_S(\mathcal{L}_S \downarrow + \mathcal{L}_S) + \lambda_C \cdot \mathcal{L}_C\end{aligned} \tag{7}$$

where $\mathcal{L} \downarrow$ refers to losses of Phase I.

4. Experiments and Evaluation

In this section, we demonstrate the potential of our approach as a bridge between both the high spatial resolution of conventional cameras and the the high temporal resolution of event cameras. No previous works have performed such fusion, so we test our method on a variety of datasets and compare to various VFI and image reconstruction methods. We also introduce a novel dataset of event camera data synchronized with high speed video. By performing a thorough evaluation of our proposed method, we show that by fusing both domains we are indeed utilizing the gained information for better interpolation results, even in cases where traditional VFI methods fail. Moreover, we show that our method has comparable performance to non-traditional VFI methods which also utilize high temporal resolution data. Our experiments are performed on two different datasets, with data from two very different event cameras and different qualities of conventional camera. The results show that while our model was trained on simulated data, it generalizes well to different real-world sensors.

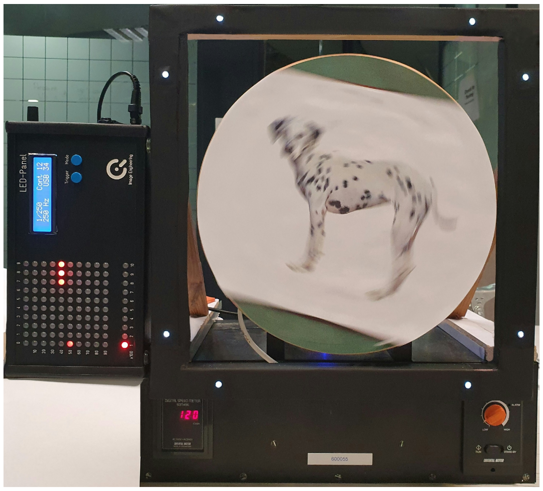

Figure 3. Our experimental setup. The eight white LEDs are co-planar with a controlled spinning wheel for spatial alignment of cameras. The LED panel (on left) is used for temporal synchronization.

4.1. Our Dataset

To the best of our knowledge, there is no publicly available dataset with spatio-temporally synchronized event data and high resolution high speed video from a conventional camera. Therefore, to test our method we created such a dataset. The dataset is captured using a Samsung GEN3 640 × 480 DVS event camera and a Samsung Galaxy S10+. The S10+ natively captures video at a resolution of 1280 × 960 and 240 FPS. To allow accurate spatio-temporal registration between the event camera and conventional camera, the test setup is placed on a planar spinning wheel bounded by co-planar blinking LED lights. The wheel was spun at 100, 150 and 200 RPM and the image on the wheel was replaced to create different scenarios. The blinking LED lights are necessary for the event camera which only responds to illumination changes. Spatial alignment is performed by locating the blinking LEDs

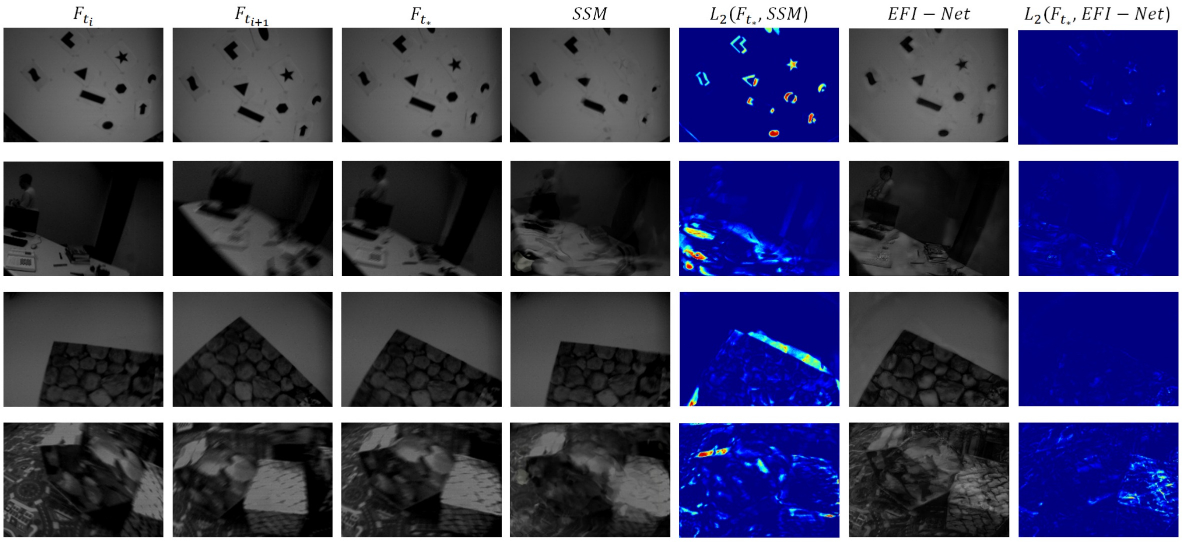

Figure 4. Interpolated frames of EFI-Net and SSM [22] from the dataset of [41]. The L_2 heatmaps emphasize the spatial mismatch error.

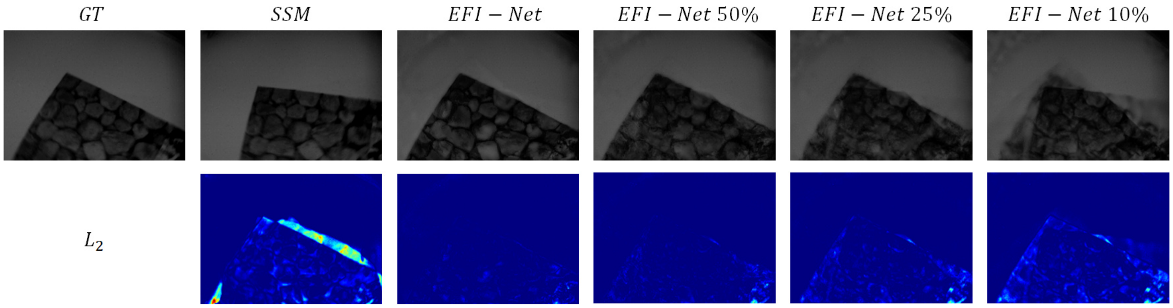

Figure 5. Interpolated frames of EFI-Net with subsampled event stream and SSM [22] on the dataset of [41]. Subsampling events gradually increases error.

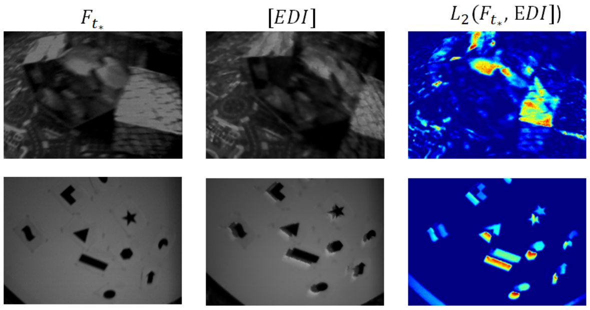

Figure 6. Interpolated frames EDI [46] from the dataset of [41]. The L_2 heatmaps emphasize the spatial mismatch error.

in both camera frames and calculating a planar homography between the two perspectives. Temporal alignment is achieved by using a precisely timed LED panel. Both can be seen in Fig. 3. This dataset will be made publicly available.

4.2. Comparison with Event Only Reconstruction

The goal of this experiment is to validate that event data alone, while having information, is insufficient for accurate frame reconstruction. To show this, we test our algorithms' performance on the dataset introduced in [41], which is captured with a DAVIS240 event camera. The DAVIS240 event camera simultaneously captures frames and events on the same pixel array, resulting in spatio-temporally aligned data. We compare EFI-Net frame interpolation, using both the intermediate events and the key frames, with the methods of [54, 64, 7] which utilize only event data. The comparison is shown in Table 1. The results highlight the gap in image quality between the two approaches where the fused approach, is on average about 25% better in SSIM, and at least 10 times better in MSE. A plausible reason to explain such a gain in quality, is due to the event camera being analogous to a differential operator on the illumination signal while event only reconstruction methods can be

Sequence	SSIM							MSE						
	EV [54]	EG [64]	SR [7]	SSM [22]	DAIN [2]	EDI [46]	EFI-Net	EV [54]	EG [64]	SR [7]	SSM [22]	DAIN [2]	[46]	EFI-Net
dynamic 6dof	0.46	0.44	0.48	0.79	0.8	0.64	0.78	0.14	0.05	0.03	0.006	0.006	0.01	0.002
boxes 6dof	0.62	0.61	0.45	0.59	0.61	0.39	0.71	0.04	0.02	0.03	0.009	0.009	0.017	0.003
poster 6dof	0.62	0.63	0.61	0.58	0.6	0.42	0.74	0.06	0.02	0.01	0.009	0.009	0.014	0.002
shapes 6dof	0.8	0.79	0.56	0.84	0.84	0.75	0.86	0.04	0.01	0.03	0.006	0.006	0.011	0.001
office zigzag	0.54	0.68	0.67	0.83	0.82	0.64	0.79	0.03	0.01	0.01	0.006	0.007	0.011	0.002
slider depth	0.58	0.59	0.54	0.91	0.91	0.61	0.8	0.05	0.02	0.02	0.005	0.005	0.012	0.002
calibration	0.7	0.71	0.67	0.91	0.9	0.73	0.85	0.02	0.01	0.01	0.005	0.006	0.011	0.002
Mean	0.617	0.636	0.569	0.778	0.783	0.597	0.79	0.054	0.02	0.02	0.007	0.008	0.013	0.002

Table 1. Comparison of EFI-Net with event reconstruction works [54, 64, 7] , with SSM [22] and DAIN [2] on the dataset of [41]. Our method outperforms all other compared methods on this dataset. For [54, 64, 7] we report metrics from [7].

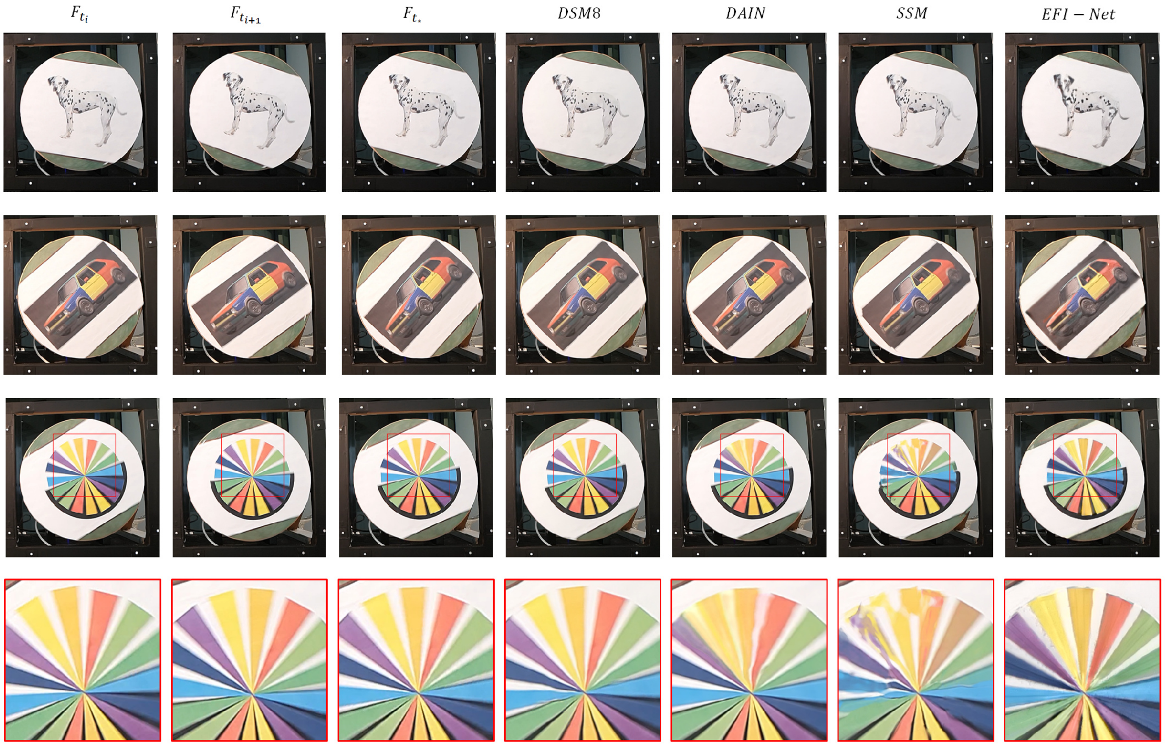

Figure 7. Interpolated frames of EFI-Net, SSM [22], DSM8 [45], and DAIN [2] on our dataset. The sequences from top are: car 100 RPM, dog 100 RPM, star 150 RPM. Bottom row is zoomed in on star 150 RPM, highlighting the artifacts of the different methods.

	SSM [22]	EFI-Net 100%	EFI-Net 50%	EFI-Net 25%	EFI-Net 10%
	SSIM				
Mean	0.779	0.790	0.769	0.743	0.713
	PSNR				
Mean	21.753	27.245	26.607	25.663	24.567

Table 2. Comparison of EFI-Net with SSM [22] on the dataset of [41], as the event stream is subsampled.

considered as a form of integration over noisy input without well defined initial conditions. In our method the keyframes serve as boundary conditions, and the temporal support of the integration is limited, thus reducing the amount of noise which is accumulated.

4.3. Comparison with Traditional VFI

Next, we aim to validate that the event data augment the key frames for VFI. To do so, we compare ourselves with Super Slomo [22] (as implemented by [17]), which uses only key frames for VFI.

Table 2 and Fig. 4 show the results on dataset [41]. The results show that while EFI-Net and [22] have a comparable score on the SSIM metric, our method is over 5 [dB] better in terms of PSNR. A suggested explanation to these scores may be found in [66]: the MSE (equivalently, PSNR) is highly vulnerable to spatial shifts. As can be seen in the L2 error heatmaps of Fig. 4, the EFI-Net frame is able to more faithfully reproduce the motion in the blind time between key frames. As seen in Table 2, reducing the number of events reduces the interpolation quality. Even so, sampling

Sequence	SSIM								PSNR							
	SSM [22]		DSM8 [45]		DAIN [2]		EFI-Net		SSM [22]		DSM8 [45]		DAIN [2]		EFI-Net	
Input\Output FPS	60\240	30\120	60\240	30\120	60\240	30\120	60\240	30\120	60\240	30\120	60\240	30\120	60\240	30\120	60\240	30\120
star 100 RPM	0.93	0.86	0.94	0.92	0.88	0.85	0.92	0.91	29.19	20.49	30.89	29.2	25.48	20.09	29.03	26.62
star 150 RPM	0.91	0.86	0.94	0.91	0.91	0.79	0.92	0.91	23.9	20.61	31	28.85	26.05	19.31	28.18	25.83
star 200 RPM	0.89	0.85	0.94	0.91	0.88	0.8	0.91	0.90	22.3	19.86	30.66	28.29	21.97	19.4	26.57	23.99
dog 100 RPM	0.94	0.89	0.94	0.92	0.94	0.9	0.92	0.92	30.91	25.06	32.09	30	31.31	27.26	28.45	27.95
car 100 RPM	0.93	0.87	0.94	0.92	0.93	0.83	0.91	0.91	29.86	23.61	31.29	29.47	30.42	23.14	28.95	27.94
Mean	0.92	0.866	0.94	0.916	0.91	0.83	0.916	0.91	27.232	21.926	31.186	29.162	27.05	21.84	28.236	26.466

Table 3. Comparison of EFI-Net, SSM [22], DSM8 [45] and DAIN [2] on our dataset. The key-frame input rates are 30 and 60, the output rates are 120 and 240, respectively.

only 10% of events still improves PSNR compared to frame only interpolation.

In Table 3 we compare Super Slomo with EFI-Net on our dataset in various configurations. The configurations include varying speed of scene motion (100, 150, 200 RPM) and varying input key frame rate of 30 and 60 FPS. The outputs of these configurations are shown in Fig. 7. The results show that as the 2D velocity of the scene increases or the key frame FPS is reduced, EFI-Net remains robust while frame only interpolation results decrease significantly (e.g. 2.5 [dB] PSNR loss for EFI-Net vs. 7 [dB] loss for SSM in transition from ”star 100” to ”star 200” at 60 FPS).

4.4. Comparison with Fusion VFI

Results presented in previous sections showed that event data enhances VFI. As a final test, we show that using event data as an auxiliary source can have comparable performance to fusion of multiple frame streams. We compare EFI-Net with Deep Slow Motion [45] (DSM), which fuses a high resolution low FPS frame stream with a low resolution high FPS frame stream. In Table 3 and Fig. 7 we compare our algorithm with DSM [45] using auxiliary streams with spatial resolution of $1/8$ of the primary stream (denoted DSM8). With EFI-Net it is possible to interpolate at the time resolution of the event stream ˜1 [ms], while equivalent FPS with DSM would require more data.

4.5. Comparison with Event Fusion VFI

Finally, we seek to compare ourselves with the closest work in terms of input structure. [46] uses both events and frames primarily for deblurring and may also be used for VFI. The method of [46] differs from ours in that it performs VFI by extrapolating from a single frame using event data, rather than interpolating between two frames using the event data as with our method. Therefore, we tested [46] using their publicly available code by inputting F_{t_i} and the event data $E_{[t_i,t]}$ to estimate the frame at time t, F_{t^*}. EFI-Net outperforms [46] by 8.2 [dB] in terms of PSNR and by 0.18 in terms of SSIM on the dataset of [41]. Fig. 6 a visual comparison of the results of the two algorithms and Table 1 gives a quantitative comparison. This result leads us to believe that EFI-Net generalizes better to experimental event data than the event generation and noise models assumed in [46].

4.6. Limitations

The fixed number of channels in the input tensor forces temporal binning of the event stream, which can lead to data loss and sub-optimal interpolation. Additionally, and similarly to other VFI methods, the output video stream is not always temporally consistent with a noticeable difference between key frames and interpolated frames.

Furthermore, our dataset may not sufficiently represent real world data, as it lacks real world elements (e.g. occlusions, multiple depths, varying motion types, etc.). Since no hybrid device capturing synchronized event stream and high speed video is available, our dataset is limited by the requirement of using a planar scene to allow accurate registration between the two cameras.

5. Conclusion and Future Work

In this paper, we proposed EFI-Net, a novel method for performing VFI by fusion of data from conventional frame camera and an event camera. To our knowledge, this work is the first to propose such a full fusion pipeline. We demonstrated that a relatively simple method of generating training data generalizes well across multiple datasets and event camera models. We thoroughly tested our method on existing public datasets and our own dataset. Our tests show, that our method is significantly better than event only reconstructions and succeeds in cases where traditional frame based VFI methods fail. Additionally, we showed that our method has comparable performance to the only other VFI method we are aware of which utilizes auxiliary high temporal resolution input. Finally, we contribute a novel dataset with spatio-temporal alignment of high speed video and event camera data. We believe that this work opens the door to a new application for event cameras. Future work can explore utilizing event cameras with color information, and research optimal architectures and training methods.

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252, 2017.

[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.

[3] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 884–892, 2016.

[4] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE transactions on neural networks and learning systems*, 25(2):407–417, 2013.

[5] Roberto Castagno, Petri Haavisto, and Giovanni Ramponi. A method for motion adaptive frame rate up-conversion. *IEEE Transactions on circuits and Systems for Video Technology*, 6(5):436–446, 1996.

[6] Byeong-Doo Choi, Jong-Woo Han, Chang-Su Kim, and Sung-Jea 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.

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

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

[9] Tobi Delbruck, Yuhuang Hu, and Zhe He. V2e: From video frames to realistic dvs event camera streams, 2020.

[10] John Flynn, Keith Snavely, Ivan Neulander, and James Philbin. Deepstereo: learning to predict new views from real world imagery, Mar. 13 2018. US Patent 9,916,679.

[11] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

[12] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras, 2020.

[13] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Asynchronous, photometric feature tracking using events and frames. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–765, 2018.

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[15] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14004–14013, 2020.

[16] Mingming He, Dongdong Chen, Jing Liao, Pedro V Sander, and Lu Yuan. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4):1–16, 2018.

[17] Xin Huang and Søren Forchhammer. Cross-band noise model refinement for transform domain wyner–ziv video coding. *Signal Processing: Image Communication*, 27(1):16–30, 2012.

[18] Xin Huang, Lars Lau Rakêt, Huynh Van Luong, Mads Nielsen, François Lauze, and Søren Forchhammer. Multi-hypothesis transform domain wyner-ziv video coding including optical flow. In *2011 IEEE 13th International Workshop on Multimedia Signal Processing*, pages 1–6. IEEE, 2011.

[19] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016.

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[21] Bo-Won Jeon, Gun-Ill Lee, Sung-Hee Lee, and Rae-Hong Park. Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure. *IEEE Transactions on Consumer Electronics*, 49(3):499–508, 2003.

[22] 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.

[23] 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.

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[25] Suk-Ju Kang, Kyoung-Rok Cho, and Young Hwan Kim. Motion compensated frame rate up-conversion using extended bilateral motion estimation. *IEEE Transactions on Consumer Electronics*, 53(4):1759–1767, 2007.

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

[27] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016.

[28] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 16–23. IEEE, 2016.

[29] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359, 2016.

[30] Sung-Hee Lee, Ohjae Kwon, and Rae-Hong Park. Weighted-adaptive motion-compensated frame rate up-conversion. *IEEE Transactions on Consumer Electronics*, 49(3):485–492, 2003.

[31] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[32] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally optimal contrast maximisation for event-based motion estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6349–6358, 2020.

[33] Ziwei Liu, Raymond A Yeh, Xiaoou 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.

[34] Ziwei Liu, Raymond A Yeh, Xiaoou 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.

[35] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. Learning image matching by simply watching video. In *European Conference on Computer Vision*, pages 434–450. Springer, 2016.

[36] Ming Ronnier Luo. *CIELAB*, pages 1–7. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[37] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[38] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 498–507, 2018.

[39] 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.

[40] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[41] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide 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.

[42] 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.

[43] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018.

[44] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017.

[45] Avinash Paliwal and Nima Khademi Kalantari. Deep slow motion video reconstruction with hybrid imaging system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[46] Liyuan Pan, Richard Hartley, Cedric Scheerlinck, Miaomiao Liu, Xin Yu, and Yuchao Dai. High frame rate video reconstruction based on an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[47] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. *arXiv preprint arXiv:2004.00347*, 2020.

[48] 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.

[49] P. K. J. Park, B. H. Cho, J. M. Park, K. Lee, H. Y. Kim, H. A. Kang, H. G. Lee, J. Woo, Y. Roh, W. J. Lee, C. Shin, Q. Wang, and H. 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.

[50] P. K. J. Park, J. Seok Kim, C. W. Shin, H. Lee, W. Liu, Q. Wang, Y. Roh, J. Kim, Y. Ater, E. Soloveichik, and H. E. Ryu. Low-latency interactive sensing for machine vision. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 10.6.1–10.6.4.

[51] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2398–2407, 2019.

[52] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. 2017.

[53] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2016.

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

[55] Christian Reinbacher, Gottfried Munda, and Thomas Pock. Real-time panoramic tracking for event cameras. In *2017 IEEE International Conference on Computational Photography (ICCP)*, pages 1–9. IEEE, 2017.

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmen-

tation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[57] Hyunsurk Eric Ryu. Industrial dvs design: Key features and applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

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

[59] Timo Stich, Christian Linz, Georgia Albuquerque, and Marcus Magnor. View and time interpolation in image space. In *Computer Graphics Forum*, volume 27, pages 1781–1787. Wiley Online Library, 2008.

[60] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7244–7253, 2019.

[61] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[62] Joost van Amersfoort, Wenzhe Shi, Alejandro Acosta, Francisco Massa, Johannes Totz, Zehan Wang, and Jose Caballero. Frame interpolation with multi-scale deep loss functions and generative adversarial networks. *arXiv preprint arXiv:1711.06045*, 2017.

[63] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[64] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10081–10090, 2019.

[65] Lin Wang, Tae-Kyun Kim, and Kuk-Jin Yoon. Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8315–8325, 2020.

[66] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[67] Zihao W Wang, Peiqi Duan, Oliver Cossairt, Aggelos Katsaggelos, Tiejun Huang, and Boxin Shi. Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1609–1619, 2020.

[68] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

[69] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

[70] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[71] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based feature tracking with probabilistic data association. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4465–4470. IEEE, 2017.

[72] 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.

[73] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019.

[74] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2017.