Unsupervised Event-Based Video Reconstruction

Gereon Fox
MPI Informatik
Saarland Informatics Campus
gfox@mpi-inf.mpg.de

Xingang Pan
Nanyang Technological University
xingang.pan@ntu.edu.sg

Ayush Tewari
MIT
ayusht@mit.edu

Mohamed Elgharib
MPI Informatik
Saarland Informatics Campus
elgharib@mpi-inf.mpg.de

Christian Theobalt
MPI Informatik
Saarland Informatics Campus
theobalt@mpi-inf.mpg.de

Abstract

Event cameras report events whenever an individual pixel changes brightness. The discrete and asynchronous nature of events makes recovering pixel brightness signals a challenging task, even if conventional brightness frames are recorded along with events. Recent works have addressed this task with neural networks, which tend to be biased towards their training distribution. All methods need to deal with noise in the events to produce very high output framerates. We introduce a new approach to event-based reconstruction, not learning-based: Our model assigns each event an explicit confidence weight to account for the uncertainty arising from noise. We also introduce a novel loss term to balance confidences against each other and show that interpolation of brightness signals between events can benefit from Bézier curves. We demonstrate that allowing brightness changes between exposures can improve reconstruction quality. Our evaluation shows that our method improves the state of the art in the tasks of event-based deblurring and event-based frame interpolation.

1. Introduction

Classic frame-based cameras synchronously expose their pixels to incoming brightness, for a nonzero exposure time. The resulting frames indicate the average brightness flowing into each pixel during exposure. This averaging is problematic for fast motion: Either a high-end camera with very short exposure time and high framerate is used, which requires large amounts of memory for storing frames and consumes considerable amounts of power, or a low-end camera with rather long exposure time and low framerate is used, where the averaging leads to motion blur in the frames.

Event cameras [3, 6, 9, 12, 26] have their pixels *asynchronously* report so-called "events": A pixel emits an event as soon as the brightness it measures deviates from a reference value by a sufficient margin c. The reference value is usually the level of brightness measured at the previous event. Event cameras measure brightness at a far higher rate than conventional cameras, which greatly reduces motion blur and also makes them suitable for low-light conditions, in which classic cameras tend to produce very strong noise. In addition, the fact that pixels produce data only when they measure a *change* in brightness makes event cameras encode sequences in a much more compact format than frame-based cameras. In this work, we focus on event cameras that not only record events, but also low-framerate, long exposure brightness frames through the same pixel matrix, see Fig. 1. Given a recording of such a camera it is desirable to reconstruct a brightness signal that could explain both the events and the frames: In doing so, one can obtain deblurred versions of the recorded frames, or interpolate frames in-between exposures. This allows one to record sequences

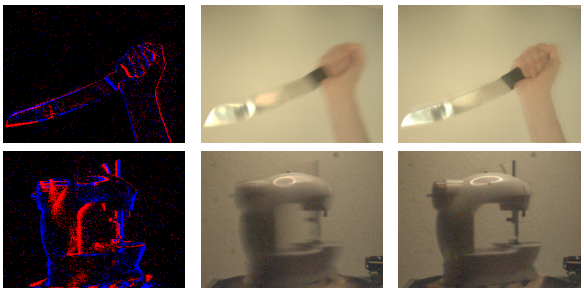

Figure 1. Given an event stream and a sequence of long-exposure RGB frames, we optimize a model of a continuous brightness signal that plausibly explains the input. We can query our model at arbitrary exposure times, for example to obtain deblurred output.

at lower memory bandwidth and power consumption than with a classic high-framerate camera, while capturing more temporal detail than a low-framerate camera would.

Many approaches to the task of event-based video reconstruction [7, 10, 17, 19, 20, 25, 28, 31, 32, 34, 38] are based on neural networks, which learn priors that enable them to account for noise in the input. However, their training not only requires a sufficiently large dataset, which can be expensive and difficult to obtain, but also tends to limit their performance on scenes that are not in line with the training distribution. In contrast, other methods [15, 33] do not rely on prior training, but exploit the semantic properties of events in a principled way. Our method belongs into this category, but our contributions make it improve the SOTA:

We define a family of brightness signals that incorporate the input events by construction, while compatibility with brightness frames is formulated as an energy that we minimize by gradient descent. Our method shares a loss term with mEDI [15], but our model construction and optimization strategy are completely different. Beyond that, we contribute several novel notions: Since cameras tend to produce spurious events, we allow our model to ignore events, by assigning each event a "confidence weight". Also, additional degrees of freedom allow our model to reproduce brightness frames in regions in which brightness changes are too small to trigger events. Lastly, our signals are defined as piecewise Bezier functions, instead of piecewise-constant functions, which improves reconstruction accuracy.

In summary, our contributions are:

- A new method to reconstruct high-frequency brightness signals that explain a stream of input events and a sequence of input frames with long exposure time.
- Our method does not require any training and hence no training *data* that it would be biased towards.
- Per-event a confidence weights, regularized by a novel loss term, are adjusted during optimization.
- Exposure-based control points help produce smooth signals when brightness changes did not trigger events.
- Bézier interpolation in-between events leads to higher reconstruction accuracy.

2. Related Work

Event cameras have attracted the attention of the research community for some time now, with several applications in 3D reconstruction [23, 36], feature point tracking [5, 27], spatial super-resolution [8, 11], video frame interpolation [10, 15, 33, 38], SLAM [18, 40] and many more [3]. We focus on methods for event-based video reconstruction that allow for temporal interpolation. Most current solutions use neural networks [7, 10, 17, 19, 20, 25, 28–32, 34, 35, 37, 38, 41] while some do not [1, 2, 13, 15, 16, 24, 33, 39].

Neural network-based approaches. Many works have addressed event-based video reconstruction via neural networks: The idea is to train a convolutional neural network that takes events (and sometimes also brightness frames) as input and predicts interpolated frames with high temporal resolution. To achieve this, researchers have proposed different designs, including recurrent neural networks [7, 19, 20, 25, 41], conditional GANs [28], direct supervised learning [32, 34], complex modularized pipelines [10, 29–31, 35, 37], and combinations with the mathematical formulation of event-based deblurring [38]. Most of them need training on synthetic high-FPS data, which could lead to domain gap issues when applied to real sequences. Recently, Zhang *et al*. [38] and Valles *et al*. [17] presented self-supervised learning methods that get rid of the need for ground truth training data, but still need pretraining [17, 38]. Since our method is not learning-based, it does not require any training data, but still achieves SOTA results.

Non-neural network approaches. Other works attempt to reconstruct signals without neural networks, often inferring the high framerate brightness signal purely from the physical meaning of events [1, 2, 13, 15, 16, 24, 33]. Some of them focus on reducing the effects of noisy events [24, 33, 39], which is a well known issue of event cameras. Other approaches [1, 2, 13] are "events-only", *i.e*. they do not use brightness frames and thus cannot reconstruct videos with accurate brightness. The most related work to ours is mEDI [15], which, like our method, solves an optimization problem over the entire duration of the sequence. However, our signal representation is more general as it introduces event-associated confidence score to handle noisy events, and is based on Bezier functions that offer more flexibility in representing the signals. We demonstrate the advantages of our design over mEDI and other baselines in Sec. 4.

3. Method

The input to our method is a recording from an event camera that also captures long-exposure brightness frames. Our method models a brightness signal that should be compatible with the input data: The events are used directly in the definition of the model, whereas the frames are used to formulate a loss term. We minimize this loss by gradient descent, to find values for the free parameters of the model.

We model brightness frames of resolution $w \times h$ with d color channels as functions $B_i : [0 : w] \times [0 : h] \times [0 : d] \rightarrow [0; 1]$. The camera logs exposure times $t_i^{\text{open}}, t_i^{\text{close}} \in [T_0; T_1]$, *i.e*. times at which the shutter opens/closes, which all lie within the time bounds T_0, T_1 of the sequence. Color undergoes a Bayer filter, see supplemental document.

Events are modelled as tuples $(t_j, x_j, y_j, z_j, p_j) \in [T_0; T_1] \times [0 : w] \times [0 : h] \times [0 : d] \times \{-1, +1\}$. The

emission of events is based on two numbers, c_{+1} and c_{-1}, that we refer to as the positive and negative **logarithmic brightness thresholds**, explained in Eqs. (2) and (3).

Our model treats each pixel (x, y, z) in isolation. To simplify notation, we will, in much of the remainder of this section, assume one arbitrary, but fixed pixel identity (x, y, z) and omit indices (x, y, z) and tuple components (x, y, z).

We assume that both frames B_i and events (t_j, p_j) were obtained from a brightness signal $b(t)$, i.e. for all B_i:

$$B_i = \int_{t_i^{\text{open}}}^{t_i^{\text{close}}} b(t)\, dt \tag{1}$$

and, as in previous work [16, 23, 38], the existence of two consecutive events $(t_j, p_j), (t_{j+1}, p_{j+1})$ is equivalent to the conjunction of Eqs. (2) and (3):

$$p_{j+1} \cdot (\tilde{b}(t_{j+1}) - \tilde{b}(t_j)) \geq c_{p_{j+1}} \tag{2}$$

$$\forall t \in [t_j; t_{j+1}] : -c_{-1} < \tilde{b}(t) - \tilde{b}(t_j) < c_{+1} \tag{3}$$

where $\tilde{b}(t) := \log(b(t) + \epsilon)$ for a small positive constant ϵ. Once we have found an approximation b^* of b we can compute integrals over much shorter exposures $[t'^{\text{open}}; t'^{\text{close}}]$ to re-render the sequence at arbitrary temporal resolution.

3.1. Model

Our model $M(t)$ represents not $b^*(t)$, but its integral:

$$b^*(t) := M'(t) \tag{4}$$

We choose this formulation because our loss term (see Sec. 3.2) is expressed in terms of integrals under b^*. If we were to model b^* directly, we would have to approximate these integrals by numeric integration, subject to a trade-off between accuracy and computation time. Instead, Eq. (4) allows us to compute integrals by evaluating M (see Eq. (10)), while b^* can be computed accurately and efficiently by automatic differentiation, or even analytically.

Eqs. (2) and (3) constitute a strong prior on the set of admissible functions b^*. We utilize this prior by representing each pixel signal of $M(t)$ as an interpolation between carefully defined control points (CP, see Fig. 2) $P_k = (t_k, y_k, g_k, w_k^{\text{left}}, w_k^{\text{right}}) \in \mathbb{R}^5$, that enforce, for all k:

$$M(t_k) = y_k \qquad M'(t_k) = g_k \tag{5}$$

while $w_k^{\text{left}}, w_k^{\text{right}}$ govern the interpolation between P_k and its neighbors. The P_k are subject to the following rules:

- Each P_k belongs to one of two types:

  An **event-based** CP represents an event and its t_k is fixed to the time of the event. An **exposure-based** CP represents the transition from one brightness frame B_i to its successor and we fix $t_k := 0.5(t_i^{\text{close}} + t_{i+1}^{\text{open}})$

- $y_k \geq 0$ is a free parameter of the model.
- g_k is defined by the event semantics (see below).
- The **gradient weights** $w_k^{\text{left}}, w_k^{\text{right}}$ are free parameters.

To make our model compatible with the events by construction we have to define the gradient parameters g_k in accordance with Eqs. (2) and (3). We thus equip the pixel signal with one *confidence weight* γ_j per event and one overall parameter $\bar{b}$. γ_j represents the confidence our model has in the validity of event j. Modelling such confidence is necessary because our assumption of only two threshold values c_{+1}, c_{-1} is a strong simplification: The physical properties of the camera circuit make the thresholds rather fuzzy, leading to an entire distribution of thresholds that could have caused an event. The confidence weights account for this uncertainty. The parameter $\bar{b}$ is left free and represents the average brightness our model assigns to the pixel over the entire sequence duration, see Fig. 2. We transform the confidence weights and multiply them with the thresholds, to obtain the **effective** logarithmic thresholds c_j for each event:

$$c_j := c_{p_j} \cdot \text{sigmoid}(\gamma_j \cdot \omega_{p_j} + \beta_{p_j}) \tag{6}$$

where the scales $\omega_p \in \{\omega_{+1}, \omega_{-1}\}$ and biases $\beta_p \in \{\beta_{+1}, \beta_{-1}\}$ are shared by all pixels.

Given $\bar{b}$, chaining Eq. (2) in the form $p_{j+1} \cdot (\tilde{b}^*(t_{j+1}) - \tilde{b}^*(t_j)) = c_j$ admits only one possible valuation for those g_k that are *event*-based , if one assumes that brightness is constant between events (see Fig. 2). For more details, please see our supplemental document. For the remaining, *exposure*-based control points P_k, we consider the latest event j that occurs before t_k. Since there exists an event-based control point $P_{k'}$ with gradient $g_{k'}$ for this event, we can set

$$g_k := \exp(\delta_k \cdot c_{\text{sign}(\delta_k)} \cdot \frac{c_j}{c_{p_j}}) \cdot g_{k'} \tag{7}$$

where $\delta_k \in [-1; 1]$ is a free model parameter, allowing log-brightness values at exposure-based control points to deviate from the value at the beginning of the event interval they lie in by at most c_{+1} or c_{-1}, satisfying Eq. (3).

We have now determined a set of control points P_k that make M consistent with Eqs. (2) and (3), on the basis of the parameter $\bar{b}$, the parameters γ_j for all events and the parameters δ_k for all exposure-based control points k. Each pixel has its own set of these parameters. The only parameters shared by the pixels are $\omega_{+1}, \omega_{-1}, \beta_{+1}, \beta_{-1}$, and c_{+1}, c_{-1}. To define M *between* the control points, we could use straight lines (which by Eq. (4) would translate into piecewise-constant brightness signals) or parabolas (leading to piecewise-linear signals), but these methods require further constraints, because they cannot comply with arbitrary combinations of control point parameters. Also, Sec. 4.2 shows that the usage of simple Bézier curves can

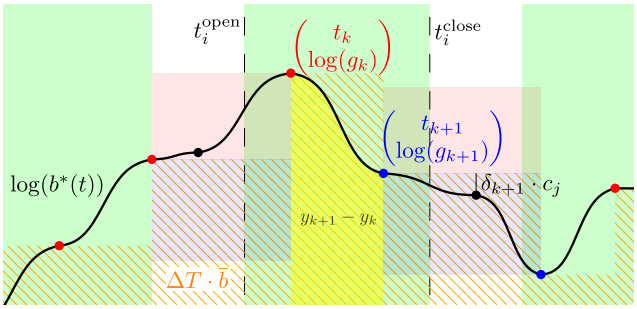

Figure 2. Our brightness signal visualized in the log domain: The green rectangles represent input exposures, with significant gaps in-between them. Based on $\bar{b}$ and the event times and polarities, the orange-hatched area can be constructed ($\Delta T := T_1 - T_0$), determining the brightness levels at each event (red for positive polarity, blue for negative). Black points are exposure-based and must lie within either the reddish or blueish rectangle that we depict them in here (Eq. (7)). Based on the g_k, y_k and other control point parameters, we construct M as a piece-wise Bézier curve (*not* depicted here) and thus $M' = b^*$, depicted as the black curve.

lead to higher accuracy. For a Bézier curve between points P_k, P_{k+1}, a general solution to Eq. (5) requires a cubic Bézier spline and thus two helper points, which must lie on certain lines determinted by the control points. The control point parameters $w_k^{\text{right}}, w_{k+1}^{\text{left}}$ tell us which helper points on these lines to choose, controlling how quickly b^* transitions from the value g_k to the value g_{k+1}. Our supplemental material contains the technical details of this construction, including the ranges we constrain our parameters to.

3.2. Optimization

M is differentiable with respect to its parameters, so we can minimize the following losses via gradient descent:

The **exposure loss** forces the model to reproduce the input brightness frames:

$$\mathcal{L}_{\text{exposure}} := \sum_{\forall i,x,y,z} \frac{\text{err}_{i,x,y,z} \left(\int_{t_i^{\text{open}}}^{t_i^{\text{close}}} b^*_{x,y,z}(t)\, dt \right)^2}{t_i^{\text{close}} - t_i^{\text{open}}} \tag{8}$$

$$\text{err}_{i,x,y,z}(b) := \begin{cases} B_i(x,y,z) - b : B_i(x,y,z) < 1 \\ \max\left(0, 1 - b\right) : B_i(x,y,z) = 1 \end{cases} \tag{9}$$

and according to Eq. (4) we can compute the integral as

$$\int_{t_i^{\text{open}}}^{t_i^{\text{close}}} b^*_{x,y,z}(t)\, dt = M_{x,y,z}(t_i^{\text{close}}) - M_{x,y,z}(t_i^{\text{open}}) \tag{10}$$

$\mathcal{L}_{\text{exposure}}$ is strictly necessary, since it is the only component of our method that informs our model about absolute levels of brightness recorded by the camera: Without it the model (in particular the parameters $\bar{b}$) could converge to arbitrary multiples of the brightness values recorded in the frames and each pixel could do so independently from the others. In addition, this term helps suppress noise that may be present in the event data. The second case in Eq. (9) is necessary for pixels that are completely saturated.

The **confidence loss** uses the helper variables ${m_{x,y,z,j}}^+$ and ${m_{x,y,z,j}}^-$ to drive all confidence weights γ_j up, such that the sigmoid term in Eq. (6) approaches 1:

$$\mathcal{L}_{\text{confidence}} := \sum_{(t_j,x,y,z,p_j)\in\mathbb{E}} \frac{\left(\frac{1}{2}{m_{x,y,z,j}}^+ + \frac{1}{2}{m_{x,y,z,j}}^-\right)^2}{\Delta T} \tag{11}$$

where $\mathbb{E}$ is the set of all events and ${m_{x,y,z,j}}^+$ and ${m_{x,y,z,j}}^-$ are penalties on the the amount of integral mass under the brightness signal that would be gained/lost by making the sigmoid term in in Eq. (6) equal to 1:

Let (x,y,z) be a pixel and (t_j,x,y,z,p_j) an event. We use the identity function SG to indicate where our implementation stops gradient backpropagation, to make $\mathcal{L}_{\text{confidence}}$ affect no parameters other than the confidence weights. For the l-th event in the pixel, there exists the event-based control point P_k and we define the "idealized" integral mass m_l under the brightness signal between the events l and $l+1$ as $m_l := \text{SG}(g_k) \cdot \Delta t_l$. The number ρ_j is the factor by which the brightness level between events j and $j+1$ would change if the confidence for event j could make the sigmoid term in Eq. (6) equal to 1:

$$\rho_j := \exp\left(\text{SG}(c_{p_j}) \cdot \left(1 - \frac{c_j}{c_{p_j}}\right)\right) \tag{12}$$

ρ_j is also the factor by which all later event intervals would increase their m_l, which is why ${m_{x,y,z,j}}^+$ equals the absolute amount of integral mass that these intervals would gain:

$${m_{x,y,z,j}}^+ := \|\rho_j - 1\| \cdot \sum_{j+1}^{n-1} m_l \tag{13}$$

Penalizing only *later* intervals would make it very cheap for the last intervals in the pixel to have low confidence. Thus ${m_{x,y,z,j}}^-$ penalizes *earlier* intervals:

$${m_{x,y,z,j}}^- := \|1 - \frac{1}{\rho_j}\| \cdot \sum_{l=0}^{j} m_l \tag{14}$$

$\mathcal{L}_{\text{confidence}}$ makes sure that we take every event as "serious" as possible. Without it some events may be needlessly assigned a low confidence, leading to high-frequency information being ignored, and thus more motion blur.

The **linearity** regularizer encourages our Bézier curves to have linear derivatives and thus our brightness signal to be piecewise linear in areas where other losses do not determine a specific shape. We achieve this between P_k and

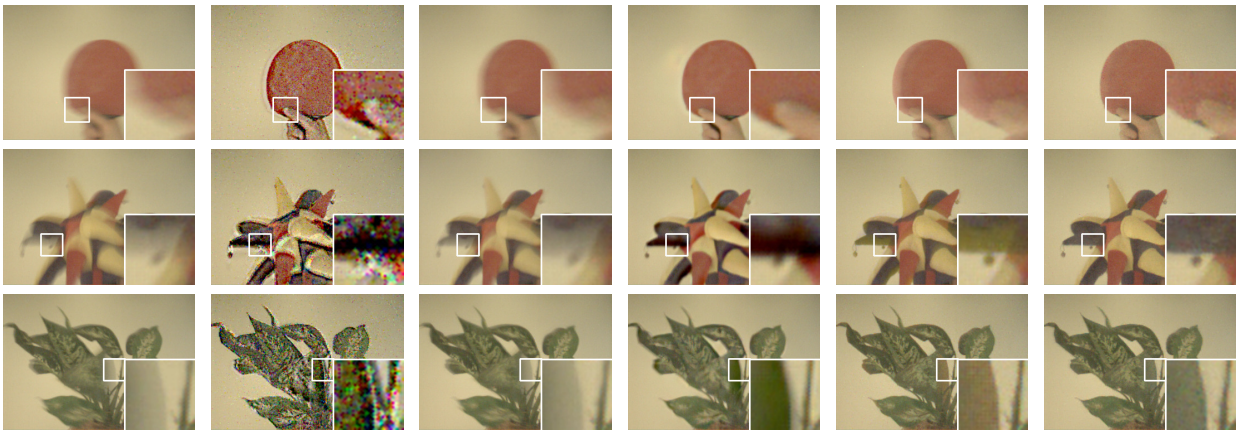

Figure 3. We compare our method to AKF [33], EVDI [38], LEDVDI [10] and mEDI [15]. We omit the event input in this figure. Input exposure time was approximately 0.2 seconds, but output exposure time for all methods was 0.002 seconds. Our method manages to reduce motion blur most effectively. Our supplemental video results give a much better impression than these still frames.

P_{k+1} by penalizing the surface area $A_{x,y,z,k}$ of the triangle between the points (t_k, g_k), (t_{k+1}, g_{k+1}) and

$$\begin{pmatrix} t_{k,k+1} \\ g_{k,k+1} \end{pmatrix} := \begin{pmatrix} \frac{1}{2}(t_k + t_{k+1}) \\ b_{x,y,z}^{*}(\frac{1}{2}(t_k + t_{k+1})) \end{pmatrix}$$

which yields the loss formulation

$$\mathcal{L}_{\text{linearity}} := \sum_{\substack{\forall k,x,y,z: \\ P_k, P_{k+1} \in \mathbb{P}(x,y,z)}} \frac{A_{x,y,z,k}^2}{\Delta t_{x,y,z,k}} \tag{15}$$

where $\mathbb{P}(x, y, z)$ is the set of CP for pixel (x, y, z) and $\Delta t_{x,y,z,k} := t_{k+1} - t_k$. Sec. 4.2 shows that this loss gives better results that enforcing linearity by construction.

We minimize our overall loss

$$\mathcal{L} := 1 \cdot \mathcal{L}_{\text{exposure}} + 0.2 \cdot \mathcal{L}_{\text{confidence}} + 0.1 \cdot \mathcal{L}_{\text{linearity}}$$

by gradient descent, updating the model parameters $\bar{b}$ for all pixels (x, y, z), the parameters γ_j for all events, the parameters y_k, w_k^{left}, and w_k^{right} for all control points, the parameters δ_k for all exposure-based control points, and the global parameters c_{+1}, c_{-1}, ω_{+1}, ω_{-1}, β_{+1}, β_{-1}.

4. Results

We recorded sequences with a DAVIS 346C, at exposure $0.2s$, resulting in 4.5FPS - 5FPS, due to the shutter remaining closed for significant durations between frames. For the DAVIS 346C, the duration of these so-called "exposure gaps" (visualized as black input frames in supplemental video) remains constant as exposure time is decreased, leading to exposures covering less and less sequence time. The long exposure of $0.2s$ was chosen in order to keep this coverage high (92%). If one were to reduce exposure time and thus increase framerates, frames would contain *less* information about the scene instead of more: For example, exposure time $0.01s$ would result in 35FPS, covering only about 35% of sequence time. Even shorter exposures and higher framerates would make frames even less informative, because they would represent only very short time intervals, with large gaps in-between. We thus constrain the evaluation to a setting where the exposures cover most of the sequence duration. Ground truth signals for quantitative evaluation was obtained in 2 ways: First, we dropped every second frame in our recordings, allowing us to evaluate how well a reconstruction method is able to compute the missing long-exposure frames. Second, similarly to previous work [38], we synthesized events from high-framerate RGB sequences: We temporally upsampled 30 REDS sequences [14] to 800Hz using FILM [21, 22] and synthesized events using ESIM [4] (threshold $c = 0.2$). We also derived a low-FPS, long-exposure version from each latent image sequence, at 10Hz, without exposure gaps, to be used as input. We cannot show results on the Color Event Camera Dataset [26], because it lacks exposure time information.

4.1. Comparison to previous work

We compare our method to 4 previous approaches to event-based reconstruction that all take events and long-exposure brightness frames as input: EVDI [38] is a recent learning-based method that is self-supervised, but needs to be pre-trained. Likewise, LEDVDI [10] is learning-based, but needs ground-truth supervision and thus is trained on synthetic data. It increases temporal frequency by a factor fixed at training time. Since our method does not require any pre-training at all, we use the official checkpoints of

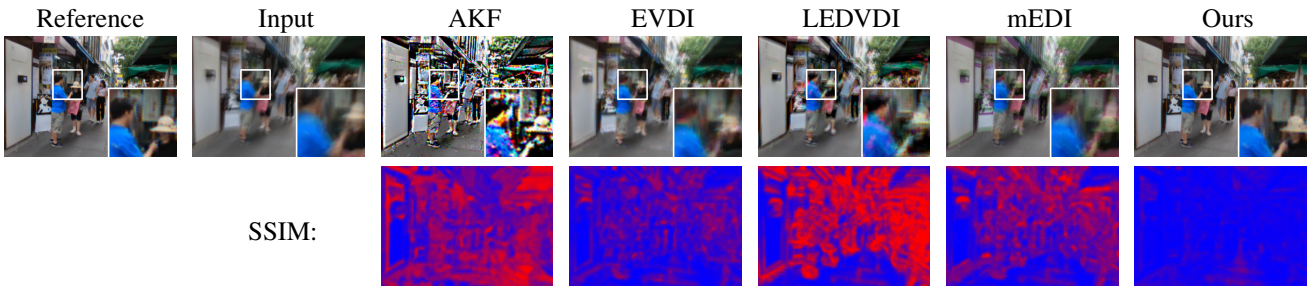

Figure 4. On our synthetic dataset, we can compare outputs to a pseudo-ground truth reference and hence evaluate results quantitatively. Input exposure time was 0.1s, output exposure time was 0.002s. SSIM in particular often shows the superiority of our method.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
AKF	14.42dB	0.4861	0.3675
EVDI (pretrained)	23.32dB	0.7356	0.2504
LEDVDI (pretrained)	20.66dB	0.6491	0.2000
mEDI	24.68dB	0.7888	0.1831
Ours	**29.69dB**	**0.9039**	**0.0739**

Table 1. We evaluated all methods quantitatively on our synthetic dataset based on REDs [14]. The scores in this table are averaged over the 30 sequences in the REDS validation set.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
AKF	22.64dB	0.5033	0.4505
EVDI (pretrained)	33.92dB	**0.9498**	**0.0651**
LEDVDI (pretrained)	30.74dB	0.9120	0.0739
mEDI	34.91dB	0.9114	0.0876
Ours	**37.91dB**	0.9251	0.0882

Table 2. Evaluation of our frame drop experiment. Exposure for both reference and input was 0.2s. Spurious events in the scene background make it hard for our method to keep background brightness constant, hence our scores do not beat those of EVDI.

Variant	Ref. exp. 0.1s PSNR	Ref. exp. 0.1s SSIM	Ref. exp. 0.002s PSNR	Ref. exp. 0.002s SSIM
Lin. interp.	40.72dB	0.987	29.50dB	0.901
Parab. interp.	39.21dB	0.980	28.95dB	0.889
No confid.	43.96dB	0.993	**30.25dB**	**0.911**
No exp.-CP	43.69dB	0.993	29.52dB	0.901
No $\mathcal{L}_{\text{confidence}}$	**45.80dB**	**0.996**	27.70dB	0.868
No $\mathcal{L}_{\text{linearity}}$	42.61dB	0.992	20.92dB	0.668
Ours (full)	45.17dB	0.995	29.69dB	0.904

Table 3. Ablation study on synthetic data, comparing outputs to input frames (exposure $0.1s$) and to pseudo ground truth (exposure $0.002s$). Since this dataset does not contain the real-world noise, the event confidences in our method, as well as $\mathcal{L}_{\text{confidence}}$ are not improving performance. However, our full method ranks second best more often than any other method ranks best. Both linear interpolation and parabolic interpolation lead to the input frames being reproduced far less faithfully.

EVDI ("GoPro" checkpoint) and LEDVDI (frequency factor 6). EVDI is self-supervised and does not require ground truth, so we overfit it to the input for 10 epochs, helping overcome the domain gap between the input and the training data. Like our method, AKF [33] and mEDI [15] do not require pre-training. A comparison to TimeLens(++) [30, 31] is out of the scope, because TimeLens requires very short exposures with as little motion blur as possible, whereas our goal is to deal with long exposures that do contain blur, see supplemental. We have each method produce outputs at 500FPS. We have extended all previous methods to processing coloured data, by applying the to each colour channel individually. This is necessary because official checkpoints have been trained on single channel data only.

Fig. 3 shows a comparison on multiple recordings: Methods are expected to turn inputs with exposure 0.2s into output with exposure 0.002s, which should reduce motion blur. AKF produces strong spatial noise. Both EVDI and mEDI give results with considerable motion blur. Surprisingly, LEDVDI, which can only produce output at exposure time 0.033s, manages to reduce blur considerably for the racket, but suffers from a "pulsing" artifact (see supplemental video). Our method deblurs best, to be seen, for example, in the third row, where EVDI and mEDI struggle with clearly resolving the right edge of the foreground leaf.

We evaluated methods on our synthetic data, comparing outputs to pseudo ground truth. Fig. 4 confirms many observations from the recordings, with the exception of LEDVDI, which now also gives blurry results, possibly due to the bias of LEDVDI towards its training distribution. In fact, Tab. 1 lists rather weak scores for LEDVDI and AKF, while our method consistently outperforms the others.

For quantitative comparison on real inputs, we modified our recordings by dropping every second frame and using the original as a blurry long-exposure reference (Sec. 4).

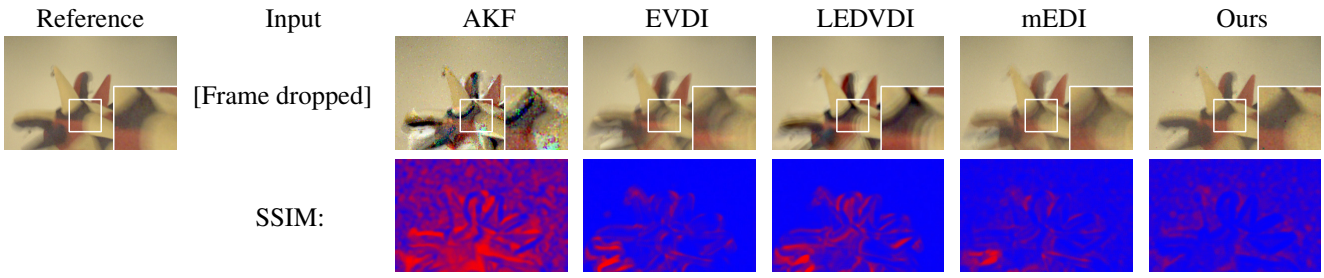

Figure 5. After dropping every second recorded frame (Sec. 4), we can use the remaining frames and the events as input, while the original recording can serve as a long-exposure reference. All methods struggle in this setting, but ours performs best.

All methods struggle in this setting (Fig. 5). As expected, those frames that were still remaining in the input were reproduced more faithfully than those that were dropped, confirming that event-based reconstruction greatly benefits from the availability of long-exposure brightness frames. Tab. 2 averages scores over a number of recordings: In both SSIM and LPIPS our method is outperformed by EVDI. The reason for this seems to be that the spurious events that the camera reports for the scene background (*i.e.* exactly in those areas where *no* motion is happening) make it hard for our method to keep the brightness of the background pixels constant. Normally $\mathcal{L}_{\text{exposure}}$ (see Sec. 3.2) would correct that, but without the dropped frames our method has no way of knowing whether these events are legitimate or not. It is here, where the learned priors of EVDI prove to be an advantage. Error maps and scores in supplemental video.

4.2. Ablation study

Bézier interpolation. Sec. 3 specifies that the interpolation in-between the control points of our model is computed as Bézier splines, which means that the function M, *i.e.* the integral under the brightness signal (*not* the brightness signal itself) consists of Bézier curves. Since we also regularize these curves to become parabolas, by having $\mathcal{L}_{\text{linearity}}$ encourage the brightness signal to become piecewise linear, we have to investigate whether these choices really give better results than making the interpolation use parabolas already by definition, or even linear functions (in which case the brightness signal would be piecewise constant). However, both these "simplifications" require the addition of further constraints to our model, because piecewise parabolas or lines cannot satisfy Eq. (5) in all cases, since they admit fewer degrees of freedom than Bézier curves do. Technically this means that we cannot leave the y coordinates y_k free, and instead have to compute them from other parameters in a way that allows parabolas or lines to be used. These additional constraints change the dynamics of our signal representation during optimization. Tab. 3 shows that this leads to significant degradations in quality, especially with regard to the ability of our method to reconstruct the input frames. Furthermore, the second row of Fig. 7 shows more spatial noise for the simplified interpolation methods.

Exposure-based control points. Similarly, Tab. 3 shows that omitting exposure-based control points harms the fidelity with which input frames are reconstructed. Fig. 6 gives a possible explanation: Pixels in the sky region change their brightness only very slightly as the camera is panning. The logarithmic brightness threshold of 0.2 that we used to synthesize events for the REDS sequences is apparently too large to trigger frequent events for such subtle changes. Therefore, the control points in these pixels are very sparse and cannot satisfy $\mathcal{L}_{\text{exposure}}$, which leads to the sky region having bad SSIM scores. Only exposure-based control points add the necessary degrees of freedom here.

Confidence weights. Tab. 3 shows our full method outperform all ablations except those that are missing confidences or $\mathcal{L}_{\text{confidence}}$. The reason why omitting these seems to even slightly improve the metrics in this ablation study is that since we did not simulate event camera noise, almost all events in the synthetic dataset are legitimate (up to corner cases due to quantisation), so allowing confidences to deviate from 1 cannot do much good. Note that the PSNR metric (given only because previous work gives it) is unreliable in the case of Tab. 3: Not only is SSIM a more advanced metric for perceived quality, but also does PSNR diverge towards infinity as images become more similar to the reference, making PSNR differences less and less significant. We therefore suggest to focus on SSIM values especially in the first half of Tab. 3, where similarity is very high. On real data however (Fig. 7), many events are not legitimate at all and especially in-between exposure periods the lack of confidence weights can lead to severe artifacts.

Confidence loss. SSIM scores in Tab. 3 show that omission of $\mathcal{L}_{\text{confidence}}$ leads to a significant loss of quality on synthetic data at short exposure times, suggesting that confidences tend to needlessly deviate from 1. The third row of Fig. 7 confirms this as well, with the omission of $\mathcal{L}_{\text{confidence}}$ leading to strong artifacts in the orange tiles of the cube.

Linearity regularizer. Using Bézier interpolation without having $\mathcal{L}_{\text{linearity}}$ regularize those parameters that are not

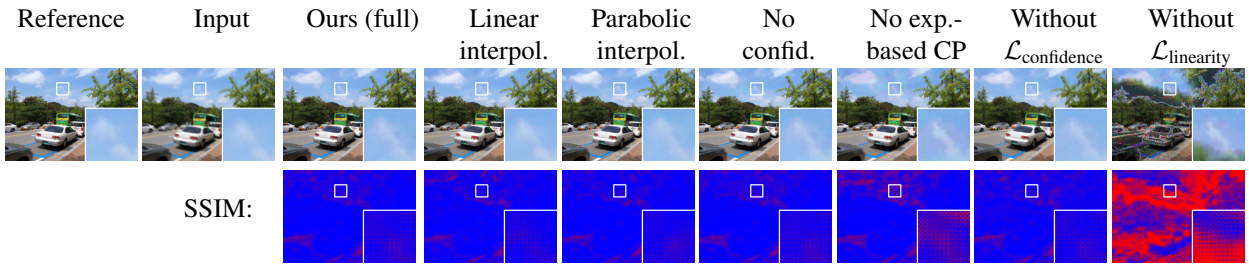

Figure 6. Ablation on synthetic data. We observe that the sky region leads to significant SSIM error if no exposure-based control points are used. This is because the pixels in this region change their brightness very slightly, without triggering events. Enabling exposure-based control points allows $\mathcal{L}_{\text{exposure}}$ to correct the resulting brightness errors. The checkerboard patterns visible in the error maps are due to the Bayer filter of the camera and must appear regardless of the method used. Please see our supplemental video for details.

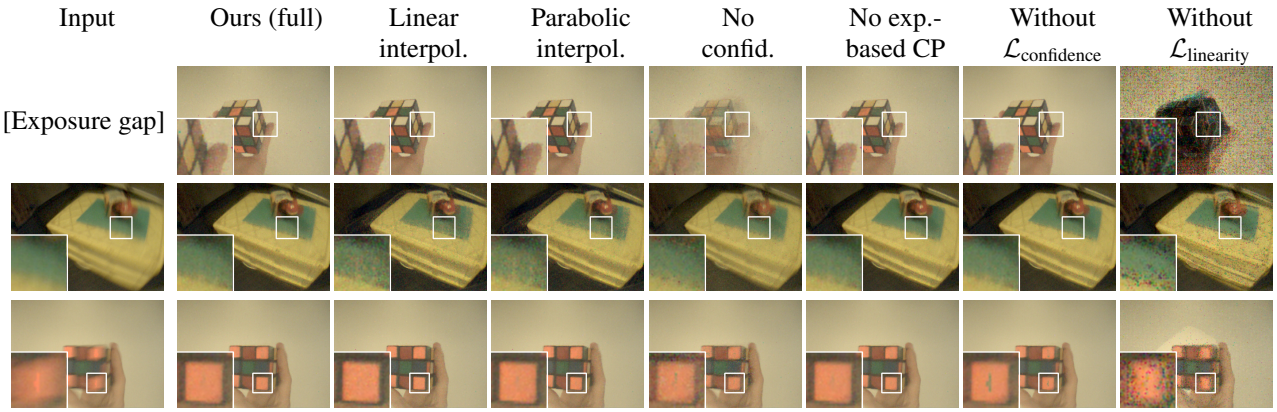

Figure 7. Qualitative ablation on real recordings, at output exposure time 0.002s. The first row was captured in-between two input exposure periods, hence we cannot show an input frame. In the second row, we observe that both linear and parabolic interpolation lead to increased spatial noise in the green mat on the yellow box. In all three rows, the result without confidences shows significantly more blur and the third one shows strong artifacts in the orange tiles of the cube if $\mathcal{L}_{\text{confidence}}$ is not used. Once more, disabling $\mathcal{L}_{\text{linearity}}$ proves harmful.

strictly derived from the event data leads to very strong artifacts, as evidenced by the poor scores for short exposures in Tab. 3 and the visual results in Figs. 6 and 7.

5. Limitations

$\mathcal{L}_{\text{exposure}}$ requires knowledge of exposure time stamps. Another limitation of our method is its resource usage: Representing a sequence in memory requires significant GPU capacity. We experimented with applying our method only to pairs of consecutive brightness frames and stitching results together. While this leads to qualitatively comparable results with less memory demand, it does take more time, because frame pairs need to overlap and because optimizing for one frame pair does not fully make use of GPU parallelism. We chose to value computation time higher than memory consumption and thus reported results for global optimization only. Nevertheless, depending on scene characteristics (duration, number of events), our run time on an RTX 3090 can range from tens of minutes to several hours. For an extended discussion see the supplemental document.

6. Conclusion

We have presented a method for event-based video reconstruction. Instead of learning from a training set that our method would then be biased to, we have exploited the semantic model of events in a principled manner. On this basis we introduced per-event confidence weights as a novel way of dealing with event camera noise, which required a novel regularization loss. In addition, we showed that equipping the brightness signal with new degrees of freedom in-between exposures and even, by the use of Bézier interpolation, in-between single events, helps improve output quality to the point where we outperform state of the art methods at a temporal resolution 100 times as high as that of the input. In contrast to methods like TimeLens [31] we manage to do so given very long exposure input frames.

Acknowledgements. We thank Kartik Teotia, for helping with the evaluation of related methods, and Viktor Rudnev, for supplying some of the sequences used in the evaluation.

References

[1] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016. 2

[2] Souptik Barua, Yoshitaka Miyatani, and Ashok Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE, 2016. 2

[3] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. 1, 2

[4] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2020. 5

[5] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. EKLT: Asynchronous, photometric feature tracking using events and frames. *Int. J. Comput. Vis. (IJCV)*, 2019. 2

[6] Giacomo Indiveri, Bernabé Linares-Barranco, Tara Julia Hamilton, André van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih-Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, et al. Neuromorphic silicon neuron circuits. *Frontiers in neuroscience*, 5:73, 2011. 1

[7] 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 (CVPR)*, June 2020. 2

[8] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Turning frequency to resolution: Video super-resolution via event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7772–7781, June 2021. 2

[9] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15μ s latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008. 1

[10] Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy Ren. Learning event-driven video deblurring and interpolation. *ECCV*, 2020. 2, 5

[11] Sayed Mohammad Mostafavi Isfahani, Yeongwoo Nam, Jonghyun Choi, and Kuk-Jin Yoon. E2sri: Learning to super-resolve intensity images from events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2

[12] 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. 1

[13] 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. 2

[14] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 5, 6

[15] 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*, pages 1–1, 2020. 2, 5, 6

[16] 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 (CVPR)*, June 2019. 2, 3

[17] Federico Paredes-Vallés and Guido C. H. E. de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3445–3454, 2021. 2

[18] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. EMVS: Event-based multi-view stereo–3d reconstruction with an event camera in real-time. *Int. J. Comput. Vision*, 126(12):1394–1414, dec 2018. 2

[19] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3852–3861, 2019. 2

[20] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(6):1964–1980, 2021. 2

[21] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. FILM: Frame interpolation for large motion. In *The European Conference on Computer Vision (ECCV)*, 2022. 5

[22] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Tensorflow 2 implementation of "FILM: Frame Interpolation for Large Motion". https://github.com/google-research/frame-interpolation, 2022. 5

[23] Viktor Rudnev, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt. EventHands: Real-time neural 3d hand pose estimation from an event stream. In *International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[24] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 308–324, Cham, 2019. Springer International Publishing. 2

[25] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert E. Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 156–163, 2020. 2

[26] Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1684–1693. IEEE, June 2019. 1, 5

[27] Hochang Seok and Jongwoo Lim. Robust feature tracking in dvs event stream using bézier mapping. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1647–1656. IEEE, Mar. 2020. 2

[28] 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*, pages 534–549, Cham, 2020. Springer International Publishing. 2

[29] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 412–428. Springer, 2022. 2

[30] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17755–17764, 2022. 2, 6

[31] 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. 2, 6, 8

[32] Bishan Wang, Jingwei He, Lei Yu, Gui-Song Xia, and Wen Yang. Event enhanced high-quality image recovery. In *European Conference on Computer Vision*. Springer, 2020. 2

[33] Ziwei Wang, Yonhon Ng, Cedric Scheerlinck, and Robert Mahony. An asynchronous kalman filter for hybrid event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 448–457, October 2021. 2, 5, 6

[34] Zihao W. Wang, Weixin Jiang, Kuan He, Boxin Shi, Aggelos Katsaggelos, and Oliver Cossairt. Event-driven video frame synthesis. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4320–4329, 2019. 2

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

[36] Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt. Eventcap: Monocular 3d capture of high-speed human motions using an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 2

[37] Zhiyang Yu, Yu Zhang, Deyuan Liu, Dongqing Zou, Xijun Chen, Yebin Liu, and Jimmy S Ren. Training weakly supervised video frame interpolation with events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14589–14598, 2021. 2

[38] Xiang Zhang and Lei Yu. Unifying motion deblurring and frame interpolation with events. In *CVPR*, 2022. 2, 3, 5

[39] Zelin Zhang, Anthony Yezzi, and Guillermo Gallego. Formulating event-based image reconstruction as a linear inverse problem with deep regularization using optical flow. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–18, 2022. 2

[40] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza. Semi-dense 3d reconstruction with a stereo event camera. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, pages 242–258, Berlin, Heidelberg, 2018. Springer-Verlag. 2

[41] Lin Zhu, Xiao Wang, Yi Chang, Jianing Li, Tiejun Huang, and Yonghong Tian. Event-based video reconstruction via potential-assisted spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3594–3604, 2022. 2