Time Lens: Event-based Video Frame Interpolation

Stepan Tulyakov*,[1] Daniel Gehrig*,[2] Stamatios Georgoulis[1] Julius Erbach[1]
Mathias Gehrig[2] Yuanyou Li[1] Davide Scaramuzza[2]
[1]Huawei Technologies, Zurich Research Center
[2]Dept. of Informatics, Univ. of Zurich and Dept. of Neuroinformatics, Univ. of Zurich and ETH Zurich

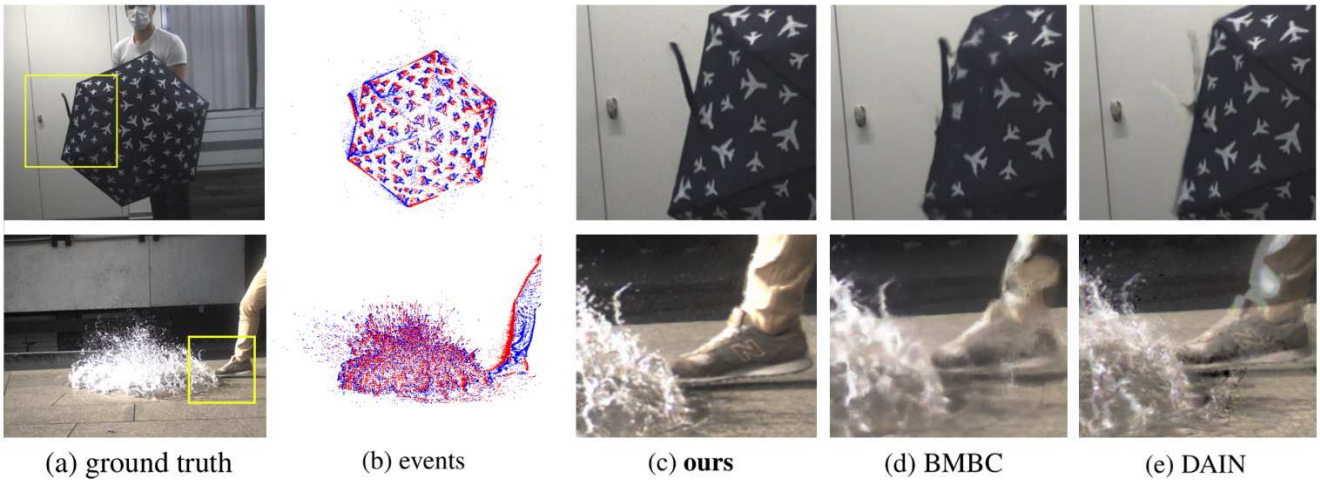


Figure 1: Qualitative results comparing our proposed method, Time Lens, with DAIN [3] and BMBC [28]. Our method can interpolate frames in highly-dynamic scenes, such as while spinning an umbrella (top row) and bursting a balloon (bottom row). It does this by combining events (b) and frames (a).

Abstract

State-of-the-art frame interpolation methods generate intermediate frames by inferring object motions in the image from consecutive key-frames. In the absence of additional information, first-order approximations, i.e. optical flow, must be used, but this choice restricts the types of motions that can be modeled, leading to errors in highly dynamic scenarios. Event cameras are novel sensors that address this limitation by providing auxiliary visual information in the blind-time between frames. They asynchronously measure per-pixel brightness changes and do this with high temporal resolution and low latency. Event-based frame interpolation methods typically adopt a synthesis-based approach, where predicted frame residuals are directly applied to the key-frames. However, while these approaches can capture non-linear motions they suffer from ghosting and perform poorly in low-texture regions with few events. Thus, synthesis-based and flow-based approaches are complementary. In this work, we introduce Time Lens*, a novel method that leverages the advantages of both. We extensively evaluate our method on three synthetic and two real benchmarks where we show an up to 5.21 dB improvement in terms of PSNR over state-of-the-art frame-based and event-based methods. Finally, we release a new large-scale dataset in highly dynamic scenarios, aimed at pushing the limits of existing methods.*

Multimedia Material

The High-Speed Event and RGB (HS-ERGB) dataset and evaluation code can be found at: http://rpg.ifi.uzh.ch/timelens

1. Introduction

Many things in real life can happen in the blink of an eye. A hummingbird flapping its wings, a cheetah accelerating towards its prey, a tricky stunt with the skateboard, or even a baby taking its first steps. Capturing these moments as high-resolution videos with high frame rates typi-

*indicates equal contribution

cally requires professional high-speed cameras, that are inaccessible to casual users. Modern mobile device producers have tried to incorporate more affordable sensors with similar functionalities into their systems, but they still suffer from the large memory requirements and high power consumption associated with these sensors.

Video Frame Interpolation (VFI) addresses this problem, by converting videos with moderate frame rates high frame rate videos in post-processing. In theory, any number of new frames can be generated between two keyframes of the input video. Therefore, VFI is an important problem in video processing with many applications, ranging from super slow motion [10] to video compression [41].

Frame-based interpolation approaches relying solely on input from a conventional frame-based camera that records frames synchronously and at a fixed rate. There are several classes of such methods that we describe below.

Warping-based approaches [20, 10, 43, 21, 28] combine optical flow estimation [8, 16, 35] with image warping [9], to generate intermediate frames in-between two consecutive key frames. More specifically, under the assumptions of linear motion and brightness constancy between frames, these works compute optical flow and warp the input keyframe(s) to the target frame, while leveraging concepts, like contextual information [20], visibility maps [10], spatial transformer networks [43], forward warping [21], or dynamic blending filters [28], to improve the results. While most of these approaches assume linear motion, some recent works assume quadratic [42] or cubic [5] motions. Although these methods can address non-linear motions, they are still limited by their order, failing to capture arbitrary motion.

Kernel-based approaches [22, 23] avoid the explicit motion estimation and warping stages of warping-based approaches. Instead, they model VFI as local convolution over the input keyframes, where the convolutional kernel is estimated from the keyframes. This approach is more robust to motion blur and light changes. Alternatively, *phase-based approaches* [18] pose VFI as a phase shift estimation problem, where a neural network decoder directly estimates the phase decomposition of the intermediate frame. However, while these methods can in theory model arbitrary motion, in practice they do not scale to large motions due to the locality of the convolution kernels.

In general, all frame-based approaches assume simplistic motion models (e.g. linear) due to the absence of visual information in the blind-time between frames, which poses a fundamental limitation of purely frame-based VFI approaches. In particular, the simplifying assumptions rely on brightness and appearance constancy between frames, which limits their applicability in highly dynamic scenarios such as (*i*) for non-linear motions between the input keyframes, (*ii*) when there are changes in illumination or motion blur, and (*iii*) non-rigid motions and new objects appearing in the scene between keyframes.

Multi-camera approaches. To overcome this limitation, some works seek to combine inputs from several frame-based cameras with different spatio-temporal tradeoffs. For example, [1] combined low-resolution video with high resolution still images, whereas [24] fused a low-resolution high frame rate video with a high resolution low frame rate video. Both approaches can recover the missing visual information necessary to reconstruct true object motions, but this comes at the cost of a bulkier form factor, higher power consumption, and a larger memory footprint.

Event-based approaches. Compared to standard frame-based cameras, event cameras [14, 4] do not incur the aforementioned costs. They are novel sensors that only report the per-pixel intensity changes, as opposed to the full intensity images and do this with high temporal resolution and low latency on the order of microseconds. The resulting output is an asynchronous stream of binary "events" which can be considered a compressed representation of the true visual signal. These properties render them useful for VFI under highly dynamic scenarios (e.g. high-speed non-linear motion, or challenging illumination).

Events-only approaches reconstruct high frame rate videos directly from the stream of incoming events using GANs [37], RNNs [31, 32, 33], or even self-supervised CNNs [27], and can be thought of as a proxy to the VFI task. However, since the integration of intensity gradients into an intensity frame is an ill-posed problem, the global contrast of the interpolated frames is usually miscalculated. Moreover, as in event cameras intensity edges are only exposed when they move, the interpolation results are also dependent on the motion.

Events-plus-frames approaches. As certain event cameras such as the Dynamic and Active VIsion Sensor (DAVIS) [4] can simultaneously output the event stream and intensity images – the latter at low frame rates and prone to the same issues as frame-based cameras (e.g. motion blur) – several works [25, 40, 11, 36] use both streams of information. Typically, these works tackle VFI in conjunction with de-blurring, de-noising, super-resolution, or other relevant tasks. They synthesize intermediate frames by accumulating temporal brightness changes, represented by events, from the input keyframes and applying them to the key frames. While these methods can handle illumination changes and non-linear motion they still perform poorly compared to the frame-based methods (please see § 3.2), as due to the inherent instability of the contrast threshold and sensor noise, not all brightness changes are accurately registered as events.

Our contributions are as follows

1. We address the limitations of all aforementioned methods by introducing a CNN framework, named *Time Lens*, that marries the advantages of warping-

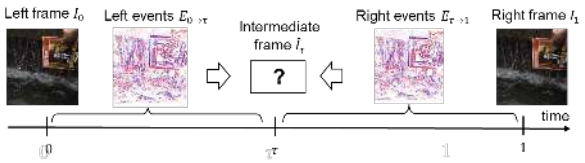


Figure 2: Proposed event-based VFI approach.

and synthesis-based interpolation approaches. In our framework, we use a synthesis-based approach to ground and refine results of high-quality warping-based approach and provide the ability to handle illumination changes and new objects appearing between keyframes (refer Fig. 7),

2. We introduce a new warping-based interpolation approach that estimates motion from events, rather than frames and thus has several advantages: it is more robust to motion blur and can estimate non-linear motion between frames. Moreover, the proposed method provides a higher quality interpolation compared to synthesis-based methods that use events when event information is not sufficient or noisy.
3. We empirically show that the proposed Time Lens greatly outperforms state-of-the-art frame-based and event-based methods, published over recent months, on three synthetic and two real benchmarks where we show an up to 5.21 dB improvement in terms of PSNR.

2. Method

Problem formulation. Let us assume an event-based VFI setting, where we are given as input the left I_0 and right I_1 RGB key frames, as well as the left $E_{0\to\tau}$ and right $E_{\tau\to1}$ *event sequences*, and we aim to interpolate (one or more) new frames $\hat{I}_\tau$ at random timesteps τ in-between the key frames. Note that, the event sequences ($E_{0\to\tau}$, $E_{\tau\to1}$) contain all asynchronous events that are triggered from the moment the respective (left I_0 or right I_1) key RGB frame is synchronously sampled, till the timestep τ at which we want to interpolate a new frame $\hat{I}_\tau$. Fig. 2 illustrates the proposed event-based VFI setting.

System overview. To tackle the problem under consideration we propose a learning-based framework, namely *Time Lens*, that consists of four dedicated modules that serve complementary interpolation schemes, i.e. warping-based and synthesis-based interpolation. In particular, (1) the *warping-based interpolation* module estimates a new frame by warping the boundary RGB keyframes using optical flow estimated from the respective event sequence; (2) the *warping refinement* module aims to improve this estimate by computing residual flow; (3) the *interpolation by synthesis* module estimates a new frame by directly fusing the input information from the boundary keyframes and the event sequences; finally (4) the *attention-based averaging* module aims to optimally combine the warping-based and synthesis-based results. In doing so, Time Lens marries the advantages of warping- and synthesis-based interpolation techniques, allowing us to generate new frames with color and high textural details while handling non-linear motion, light changes, and motion blur. The workflow of our method is shown in Fig. 3a.

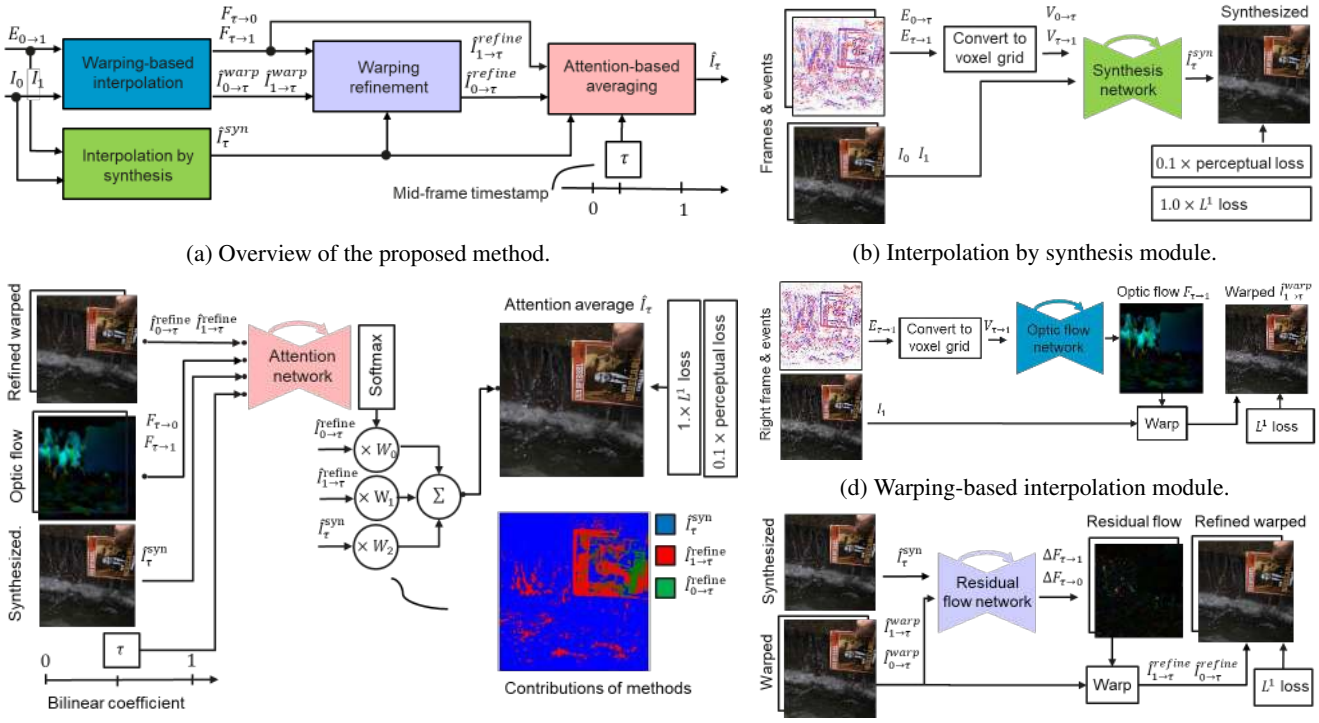
All modules of the proposed method use the same backbone architecture, which is an hourglass network with skip connections between the contracting and expanding parts, similar to [10]. The backbone architecture is described in more detail in the supplementary materials. Regarding the learning representation [7] used to encode the event sequences, all modules use the *voxel grid* representation. Specifically, for event sequence $E_{\tau_0\to\tau_{end}}$ we compute a voxel grid $V_{\tau_0\to\tau_{end}}$ following the procedure described in [45]. In the following paragraphs, we analyze each module and its scope within the overall framework.

Interpolation by synthesis, as shown in Fig. 3b, directly regresses a new frame $\hat{I}^{\text{syn}}$ given the left I_0 and right I_1 RGB keyframes and events sequences $E_{0\to\tau}$ and $E_{\tau\to1}$ respectively. The merits of this interpolation scheme lie in its ability to handle changes in lighting, such as water reflections in Fig. 6 and a sudden appearance of new objects in the scene, because unlike warping-based method, it does not rely on the brightness constancy assumption. Its main drawback is the distortion of image edges and textures when event information is noisy or insufficient because of high contrast thresholds, e.g. triggered by the book in Fig. 6.

Warping-based interpolation, shown in Fig. 3d, first estimates the optical flow $F_{\tau\to0}$ and $F_{\tau\to1}$ between a latent new frame $\hat{I}_\tau$ and boundary keyframes I_0 and I_1 using events $E_{\tau\to0}$ and $E_{\tau\to1}$ respectively. We compute $E_{\tau\to0}$, by reversing the event sequence $E_{0\to\tau}$, as shown in Fig. 4. Then our method uses computed optical flow to warp the boundary keyframes in timestep τ using differentiable interpolation [9], which in turn produces two new frame estimates $\hat{I}^{\text{warp}}_{0\to\tau}$ and $\hat{I}^{\text{warp}}_{1\to\tau}$.

The major difference of our approach from the traditional warping-based interpolation methods [20, 10, 21, 42], is that the latter compute optical flow between keyframes using the frames themselves and then approximate optical flow between the latent middle frame and boundary by using a linear motion assumption. This approach does not work when motion between frames is non-linear and keyframes suffer from motion blur. By contrast, our approach computes the optical flow from the events, and thus can naturally handle blur and non-linear motion. Although events are sparse, the resulting flow is sufficiently dense as shown in Fig. 3d, especially in textured areas with dominant mostion, which is most important for interpolation.

Moreover, the warping-based interpolation approach relying on events also works better than synthesis-based method in the scenarios when event data is noisy or not



(a) Overview of the proposed method.

(b) Interpolation by synthesis module.

(c) Attention-based averaging module.

(d) Warping-based interpolation module.

(e) Warping refinement module.

Figure 3: Structure of the proposed method. The overall workflow of the method is shown in Fig. 3a and individual modules are shown in Fig. 3d, 3b, 3e and 3c. In the figures we also show loss function that we use to train each module. We show similar modules in the same color across the figures.

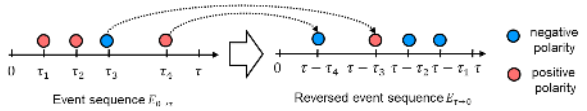


Figure 4: Example of an event sequence reversal.

sufficient due to high contrast thresholds, e.g. the book in Fig. 6. On the down side, this method still relies on the brightness constancy assumption for optical flow estimation and thus can not handle brightness changes and new objects appearing between keyframes, e.g. water reflections in Fig. 6.

Warping refinement module computes refined interpolated frames, $\hat{I}^{\text{refine}}_{0\to\tau}$ and $\hat{I}^{\text{refine}}_{1\to\tau}$, by estimating residual optical flow, $\Delta F_{\tau\to 0}$ and $\Delta F_{\tau\to 1}$ respectively, between the warping-based interpolation results, $\hat{I}^{\text{warp}}_{0\to\tau}$ and $\hat{I}^{\text{warp}}_{1\to\tau}$, and the synthesis result $\hat{I}^{\text{syn}}_{\tau}$. It then proceeds by warping $\hat{I}^{\text{warp}}_{0\to\tau}$ and $\hat{I}^{\text{warp}}_{1\to\tau}$ for a second time using the estimated residual optical flow, as shown in Fig. 3e. The refinement module draws inspiration from the success of optical flow and disparity refinement modules in [8, 26], and also by our observation that the synthesis interpolation results are usually perfectly aligned with the ground-truth new frame. Besides computing residual flow, the warping refinement module also performs inpainting of the occluded areas, by filling them with values from nearby regions.

Finally, the **attention averaging** module, shown in Fig. 3c, blends in a pixel-wise manner the results of synthesis $\hat{I}^{\text{syn}}_{\tau}$ and warping-based interpolation $\hat{I}^{\text{refine}}_{0\to\tau}$ and $\hat{I}^{\text{refine}}_{1\to\tau}$ to achieve final interpolation result $\hat{I}_{\tau}$. This module leverages the complementarity of the warping- and synthesis-based interpolation methods and produces a final result, which is better than the results of both methods by 1.73 dB in PSNR as shown in Tab. 1 and illustrated in Fig. 6.

A similar strategy was used in [21, 10], however these works only blended the warping-based interpolation results to fill the occluded regions, while we blend both warping and synthesis-based results, and thus can also handle light changes. We estimate the blending coefficients using an attention network that takes as an input the interpolation results, $\hat{I}^{\text{refine}}_{0\to\tau}$, $\hat{I}^{\text{refine}}_{1\to\tau}$ and $\hat{I}^{\text{syn}}$, the optical flow results $F_{\tau\to 0}$ and $F_{\tau\to 1}$ and bi-linear coefficient τ, that depends on the position of the new frame as a channel with constant value.

2.1. High Speed Events-RGB (HS-ERGB) dataset

Due to the lack of available datasets that combine synchronized, high-resolution event cameras and standard RGB cameras, we build a hardware synchronized hybrid sensor which combines a high-resolution event camera with

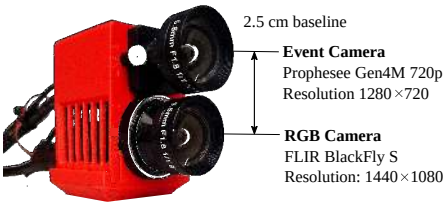


Figure 5: Illustration of the dual camera setup. It comprises a Prophesee Gen4 720p monochrome event camera (top) and a FLIR BlackFly S RGB camera (bottom). Both cameras are hardware synchronized with a baseline of 2.5 cm

.

a high resolution and high-speed color camera. We use this hybrid sensor to record a new large-scale dataset which we term the High-Speed Events and RGB (HS-ERGB) dataset which we use to validate our video frame interpolation approach. The hybrid camera setup is illustrated in Fig. 5.

It features a Prophesee Gen4 (1280×720) event camera (Fig. 5 top) and a FLIR BlackFly S global shutter RGB camera (1440×1080) (Fig. 5 bottom), separated by a baseline of 2.5 cm. Both cameras are hardware synchronized and share a similar field of view (FoV). We provide a detailed comparison of our setup against the commercially available DAVIS 346 [4] and the recently introduced setup [39] in the appendix.Compared to both [4] and [39] our setup is able to record events at much higher resolution (1280×720 vs. 240×180 or 346×260) and standard frames at much higher framerate (225 FPS vs. 40 FPS or 35 FPS) and with a higher dynamic range (71.45 dB vs. 55 dB or 60 dB). Moreover, standard frames have a higher resolution compared to the DAVIS sensor (1440×1080 vs. 240×180) and provide color. The higher dynamic range and frame rate, enable us to more accurately compare event cameras with standard cameras in highly dynamic scenarios and high dynamic range. Both cameras are hardware synchronized and aligned via rectification and global alignment. For more synchronization and alignment details see the appendix.

We record data in a variety of conditions, both indoors and outdoors. Sequences were recorded outdoors with exposure times as low as 100 µs or indoors with exposure times up to 1000 µs. The dataset features frame rates of 160 FPS, which is much higher than previous datasets, enabling larger frame skips with ground truth color frames. The dataset includes highly dynamic close scenes with non-linear motions and far-away scenes featuring mainly camera ego-motion. For far-away scenes, stereo rectification is sufficient for good per-pixel alignment. For each sequence, alignment is performed depending on the depth either by stereo rectification or using feature-based homography estimation.To this end, we perform standard stereo calibration between RGB images and E2VID [31] reconstructions and rectify the images and events accordingly. For the dynamic close scenes, we additionally estimate a global homography by matching SIFT features [17] between these two images. Note that for feature-based alignment to work well, the camera must be static and objects of interest should only move in a fronto-parallel plane at a predetermined depth. While recording we made sure to follow these constraints.

For a more detailed dataset overview we refer to the supplementary material.

3. Experiments

All experiments in this work are done using the PyTorch framework [29]. For training, we use the Adam optimizer[12] with standard settings, batches of size 4 and learning rate 10^4, which we decrease by a factor of 10 every 12 epoch. We train each module for 27 epoch. For the training, we use large dataset with synthetic events generated from *Vimeo90k* septuplet dataset [43] using the video to events method [6], based on the event simulator from [30].

We train the network by adding and training modules one by one, while freezing the weights of all previously trained modules. We train modules in the following order: synthesis-based interpolation, warping-based interpolation, warping refinement, and attention averaging modules. We adopted this training because end-to-end training from scratch does not converge, and fine-tuning of the entire network after pretraining only marginally improved the results. We supervise our network with perceptual [44] and L^1 losses as shown in Fig. 3b, 3d, 3e and 3c. We fine-tune our network on real data module-by-module in the order of training. To measure the quality of interpolated images we use structural similarity (SSIM) [38] and peak signal to noise ratio (PSNR) metrics.

Note, that the computational complexity of our interpolation method is among the best: on our machine for image resolutions of 640×480, a single interpolation on the GPU takes 878 ms for DAIN [3], 404 ms for BMBC [28], 138 ms for ours, 84 ms for RRIN [13], 73 ms for Super SloMo [10] and 33 ms for LEDVDI [15] methods.

3.1. Ablation study

To study the contribution of every module of the proposed method to the final interpolation, we investigate the interpolation quality after each module in Fig. 3a, and report their results in Tab. 1. The table shows two notable results. First, it shows that adding a warping refinement block after the simple warping block significantly improves the interpolation result. Second, it shows that by attention averaging synthesis-based and warping-based results, the interpolations are improved by 1.7 dB in terms of PSNR. This is because the attention averaging module combines the advantages of both methods. To highlight this further, we illustrate example reconstructions from these two modules in Fig. 6. As can be seen, the warping-based module excels at reconstructing textures in non-occluded areas (fourth column) while the synthesis module performs better in regions

with difficult lighting conditions (fifth column). The attention module successfully combines the best parts of both modules (first column).

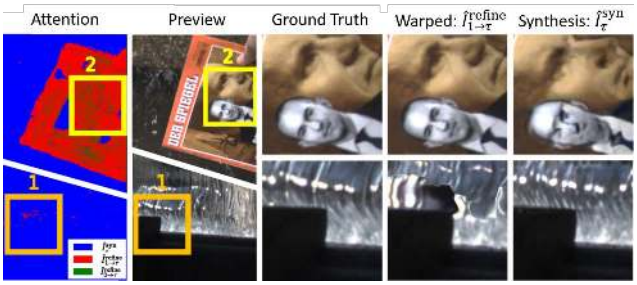


Figure 6: Complementarity of warping- and synthesis-based interpolation.

Table 1: Quality of interpolation after each module on Vimeo90k (denoising) validation set. For SSIM and PSNR we show mean and one standard deviation. The best result is highlighted.

Module	PSNR	SSIM
Warping interpolation	26.68±3.68	0.926±0.041
Interpolation by synthesis	34.10±3.98	0.964±0.029
Warping refinement	33.02±3.76	0.963±0.026
Attention averaging (ours)	**35.83±3.70**	**0.976±0.019**

3.2. Benchmarking

Synthetic datasets. We compare the proposed method, which we call *Time Lens*, to four state-of-the-art frame-based interpolation methods *DAIN* [3], *RRIN* [13], *BMBC* [28], *SuperSloMo* [10], event-based video reconstruction method *E2VID* [32] and two event and frame-based methods EDI [25] and LEDVDI [15] on popular video interpolation benchmark datasets, such as *Vimeo90k (interpolation)* [43], *Middlebury* [2]. During the evaluation, we take original video sequence, skip 1 or 3 frames respectively, reconstruct them using interpolation method and compare to ground truth skipped frames. Events for event-based methods we simulate using [6] from the skipped frames. We do not fine-tune the methods for each dataset but simply use pre-trained models provided by the authors. We summarise the results in Tab. 2.

As we can see, the proposed method outperforms other method across datasets in terms of average PSNR (up to 8.82 dB improvement) and SSIM scores (up to 0.192 improvement). As before these improvements stem from the use of auxiliary events during the prediction stage which allow our method to perform accurate frame interpolation, event for very large non-linear motions. Also, it has significantly lower standard deviation of the PSNR (2.53 dB vs. 4.96 dB) and SSIM (0.025 vs. 0.112) scores, which suggests more consistent performance across examples. Also, we can see that PSNR and SSIM scores of the proposed method degrades to much lesser degree than scores of the frame-based methods (up to 1.6 dB vs. up to 5.4 dB), as we skip and attempt to reconstruct more frames. This suggests that our method is more robust to non-linear motion than frame-based methods.

High Quality Frames (HQF) dataset. We also evaluate our method on *High Quality Frames (HQF)* dataset [34] collected using DAVIS240 event camera that consists of video sequences without blur and saturation. During evaluation, we use the same methodology as for the synthetic datasets, with the only difference that in this case we use real events. In the evaluation, we consider two versions of our method: *Time Lens-syn*, which we trained only on synthetic data, and *Time Lens-real*, which we trained on synthetic data and fine-tuned on real event data from our own DAVIS346 camera. We summarise our results in Tab. 3.

The results on the dataset are consistent with the results on the synthetic datasets: the proposed method outperforms state-of-the-art frame-based methods and produces more consistent results over examples. As we increase the number of frames that we skip, the performance gap between the proposed method and the other methods widens from 2.53 dB to 4.25 dB, also the results of other methods become less consistent which is reflected in higher deviation of PSNR and SSIM scores. For a more detailed discussion about the impact of frame skip length and performance, see the appendix. Interestingly, fine-tuning of the proposed method on real event data, captured by another camera, greatly boosts the performance of our method by an average of 1.94 dB. This suggest that existence of large domain gap between synthetic and real event data.

High Speed Event-RGB dataset. Finally, we evaluate our method on our dataset introduced in § 2.1. As clear from Tab. 4, our method, again significantly outperforms frame-based and frame-plus-event-based competitors. In Fig. 7 we show several examples from the HS-ERGB test set which show that, compared to competing frame-based method, our method can interpolate frames in the case of nonlinear ("Umbrella" sequence) and non-rigid motion ("Water Bomb"), and also handle illumination changes ("Fountain Schaffhauserplatz" and "Fountain Bellevue").

4. Conclusion

In this work, we introduce Time Lens, a method that can show us what happens in the blind-time between two intensity frames using high temporal resolution information from an event camera. It works by leveraging the advantages of synthesis-based approaches, which can handle changing illumination conditions and non-rigid motions, and flow-based approach, relying on motion estimation from events. It is therefore robust to motion blur and non-linear motions. The proposed method achieves

Table 2: Results on standard video interpolation benchmarks such as *Middlebury* [2], *Vimeo90k* (interpolation) [43] and *GoPro* [19]. In all cases, we use a test subset of the datasets. To compute SSIM and PSNR, we downsample the original video and reconstruct the skipped frames. For Middlebury and Vimeo90k (interpolation), we skip 1 and 3 frames, and for GoPro we skip 7 and 15 frames due its its high frame rate of 240 FPS. *Uses frames* and *Uses events* indicate if a method uses frames and events for interpolation. For event-based methods we generate events from the skipped frames using the event simulator [6]. *Color* indicates if a method works with color frames. For SSIM and PSNR we show mean and one standard deviation. Note, that we can not produce results with 3 skips on the Vimeo90k dataset, since it consists of frame triplet. We show the best result in each column in bold and the second-best using underscore text.

Method	**Uses frames**	**Uses events**	**Color**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Middlebury [2]				**1 frame skip**		**3 frames skips**	
DAIN [3]	✔	✘	✔	30.87±5.38	0.899±0.110	26.67±4.53	0.838±0.130
SuperSloMo [10]	✔	✘	✔	29.75±5.35	0.880±0.112	26.43±5.30	0.823±0.141
RRIN [13]	✔	✘	✔	31.08±5.55	0.896±0.112	27.18±5.57	0.837±0.142
BMBC [28]	✔	✘	✔	30.83±6.01	0.897±0.111	26.86±5.82	0.834±0.144
E2VID [31]	✘	✔	✘	11.26±2.82	0.427±0.184	26.86±5.82	0.834±0.144
EDI [25]	✔	✔	✘	19.72±2.95	0.725±0.155	18.44±2.52	0.669±0.173
Time Lens (ours)	✔	✔	✔	**33.27±3.11**	**0.929±0.027**	**32.13±2.81**	**0.908±0.039**
Vimeo90k (interpolation) [43]				**1 frame skip**		**3 frames skips**	
DAIN [3]	✔	✘	✔	34.20±4.43	0.962±0.023	-	-
SuperSloMo [10]	✔	✘	✔	32.93±4.23	0.948±0.035	-	-
RRIN [13]	✔	✘	✔	34.72±4.40	0.962±0.029	-	-
BMBC [28]	✔	✘	✔	34.56±4.40	0.962±0.024	-	-
E2VID [31]	✘	✔	✘	10.08±2.89	0.395±0.141	-	-
EDI [25]	✔	✔	✘	20.74±3.31	0.748±0.140	-	-
Time Lens (ours)	✔	✔	✔	**36.31±3.11**	**0.962±0.024**	-	-
GoPro [19]				**7 frames skip**		**15 frames skips**	
DAIN [3]	✔	✘	✔	28.81±4.20	0.876±0.117	24.39±4.69	0.736±0.173
SuperSloMo [10]	✔	✘	✔	28.98±4.30	0.875±0.118	24.38±4.78	0.747±0.177
RRIN [13]	✔	✘	✔	28.96±4.38	0.876±0.119	24.32±4.80	0.749±0.175
BMBC [28]	✔	✘	✔	29.08±4.58	0.875±0.120	23.68±4.69	0.736±0.174
E2VID [31]	✘	✔	✘	9.74±2.11	0.549±0.094	9.75±2.11	0.549±0.094
EDI [25]	✔	✔	✘	18.79±2.03	0.670±0.144	17.45±2.23	0.603±0.149
Time Lens (ours)	✔	✔	✔	**34.81±1.63**	**0.959±0.012**	**33.21±2.00**	**0.942±0.023**

Table 3: Benchmarking on the High Quality Frames (HQF) DAVIS240 dataset. We do not fine-tune our method and other methods and use models provided by the authors. We evaluate methods on all sequences of the dataset. To compute SSIM and PSNR, we downsample the original video by skip 1 and 3 frames, reconstruct these frames and compare them to the skipped frames. In *Uses frames* and *Uses events* columns we specify if a method uses frames and events for interpolation. In the *Color* column, we indicate if a method works with color frames. In the table, we present two versions of our method: *Time Lens-syn*, which we trained only on synthetic data, and *Time Lens-real*, which we trained on synthetic data and fine-tuned on real event data from our own DAVIS346 camera. For SSIM and PSNR, we show mean and one standard deviation. We show the best result in each column in bold and the second-best using underscore text.

Method	**Uses frames**	**Uses events**	**Color**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
				1 frame skip		**3 frames skips**	
DAIN [3]	✔	✘	✔	29.82±6.91	0.875±0.124	26.10±7.52	0.782±0.185
SuperSloMo [10]	✔	✘	✔	28.76±6.13	0.861±0.132	25.54±7.13	0.761±0.204
RRIN [13]	✔	✘	✔	29.76±7.15	0.874±0.132	26.11±7.84	0.778±0.200
BMBC [28]	✔	✘	✔	29.96±7.00	0.875±0.126	26.32±7.78	0.781±0.193
E2VID [31]	✘	✔	✘	6.70±2.19	0.315±0.124	6.70±2.20	0.315±0.124
EDI [25]	✔	✔	✘	18.7±6.53	0.574±0.244	18.8±6.88	0.579±0.274
Time Lens-syn (our)	✔	✔	✔	30.57±5.01	0.903±0.067	28.98±5.09	0.873±0.086
Time Lens-real (ours)	✔	✔	✔	**32.49±4.60**	**0.927±0.048**	**30.57±5.08**	**0.900±0.069**

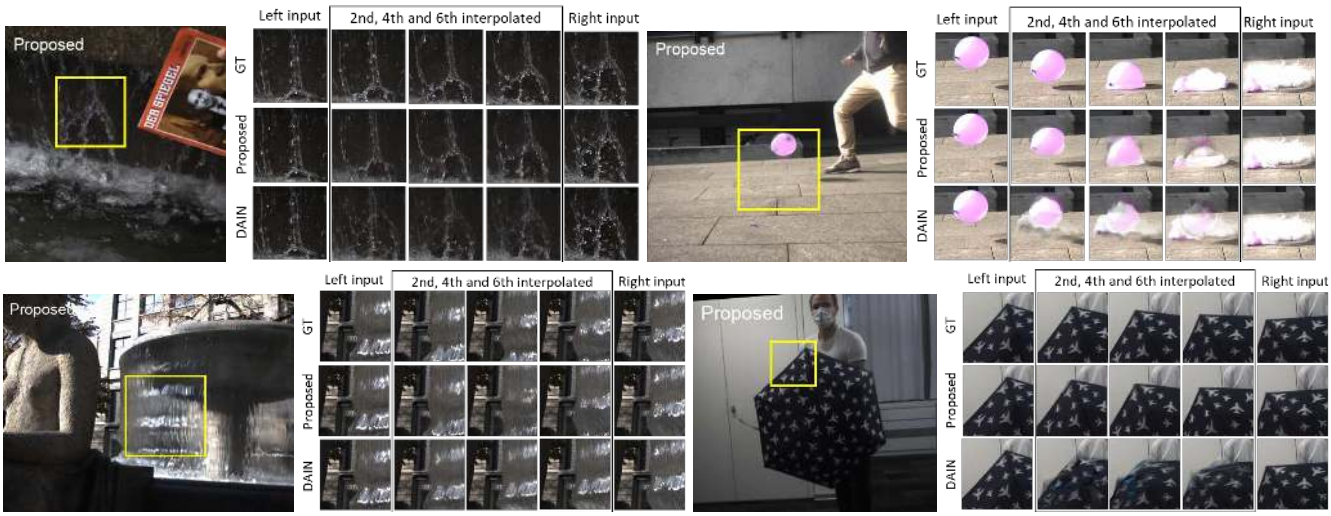


Figure 7: Qualitative results for the proposed method and its closes competitor DAIN [3] on our Dual Event and Color Camera Dataset test sequences: "Fountain Schaffhauserplatz" (top-left), "Fountain Bellevue" (bottom-left) "Water bomb" (top-right) and "Umbrella" (bottom-right). For each sequence, the figure shows interpolation results on the left (the animation can be viewed in Acrobat Reader) and close-up interpolation results on the right. The close-ups, show input left and right frame and intermediate interpolated frames.

Table 4: Benchmarking on the test set of the High Speed Event and RGB camera (HS-ERGB) dataset. We report PSNR and SSIM for all sequences by skipping 5 and 7 frames respectively, and reconstructing the missing frames with each method. By design LEDVDI [15] can interpolate only 5 frames. *Uses frames* and *Uses events* indicate if a method uses frames or events respectively. *Color* indicates whether a method works with color frames. For SSIM and PSNR the scores are averaged over the sequences. Best results are shown in bold and the second best are underlined.

Method	**Uses frames**	**Uses events**	**Color**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Far-away sequences				**5 frame skip**		**7 frames skips**	
DAIN [3]	✔	✘	✔	27.92±1.55	0.780±0.141	27.13±1.75	0.748±0.151
SuperSloMo [10]	✔	✘	✔	25.66±6.24	0.727±0.221	24.16±5.20	0.692±0.199
RRIN [13]	✔	✘	✔	25.26±5.81	0.738±0.196	23.73±4.74	0.703±0.170
BMBC [28]	✔	✘	✔	25.62±6.13	0.742±0.202	24.13±4.99	0.710±0.175
LEDVDI [15]	✔	✔	✘	12.50±1.74	0.393±0.174	n/a	n/a
Time Lens (ours)	✔	✔	✔	**33.13±2.10**	**0.877±0.092**	**32.31±2.27**	**0.869±0.110**
Close planar sequences				**5 frame skip**		**7 frames skips**	
DAIN [3]	✔	✘	✔	29.03±4.47	0.807±0.093	28.50±4.54	0.801 ± 0.096
SuperSloMo [10]	✔	✘	✔	28.35±4.26	0.788±0.098	27.27±4.26	0.775 ± 0.099
RRIN [13]	✔	✘	✔	28.69±4.17	0.813±0.083	27.46±4.24	0.800±0.084
BMBC [28]	✔	✘	✔	29.22±4.45	0.820±0.085	27.99±4.55	0.808±0.084
LEDVDI [15]	✔	✔	✘	19.46±4.09	0.602±0.164	n/a	n/a
Time Lens (ours)	✔	✔	✔	**32.19±4.19**	**0.839±0.090**	**31.68±4.18**	**0.835±0.091**

an up to 5.21 dB improvement over state-of-the-art frame-based and event-plus-frames-based methods on both synthetic and real datasets. In addition, we release the first High Speed Event and RGB (HS-ERGB) dataset, which aims at pushing the limits of existing interpolation approaches by establishing a new benchmark for both event- and frame-based video frame interpolation methods.

5. Acknowledgement

This work was supported by Huawei Zurich Research Center; by the National Centre of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation (SNSF); the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 864042).

References

[1] Enhancing and experiencing spacetime resolution with videos and stills. In *ICCP*, pages 1–9. IEEE, 2009. 2

[2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011. 6, 7

[3] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, pages 3703–3712, 2019. 1, 5, 6, 7, 8

[4] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *JSSC*, 49(10):2333–2341, 2014. 2, 5

[5] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. *arXiv preprint arXiv:2007.11762*, 2020. 2

[6] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *CVPR*, June 2020. 5, 6, 7

[7] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. 2019. 3

[8] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 2462–2470, 2017. 2, 4

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 2, 3

[10] 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 *CVPR*, pages 9000–9008, 2018. 2, 3, 4, 5, 6, 7, 8

[11] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *CVPR*, pages 3320–3329, 2020. 2

[12] Diederik P. Kingma and Jimmy L. Ba. Adam: A method for stochastic optimization. 2015. 5

[13] Haopeng Li, Yuan Yuan, and Qi Wang. Video frame interpolation via residue refinement. In *ICASSP 2020*, pages 2613–2617. IEEE, 2020. 5, 6, 7, 8

[14] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 latency asynchronous temporal contrast vision sensor. *JSSC*, 43(2):566–576, 2008. 2

[15] Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy Ren. Learning event-driven video deblurring and interpolation. *ECCV*, 2020. 5, 6, 8

[16] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pages 4463–4471, 2017. 2

[17] David G. Lowe. Distinctive image features from scale-invariant keypoints. 60, 2004. 5

[18] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *CVPR*, 2018. 2

[19] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, July 2017. 7

[20] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *CVPR*, pages 1701–1710, 2018. 2, 3

[21] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, pages 5437–5446, 2020. 2, 3, 4

[22] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017. 2

[23] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017. 2

[24] Avinash Paliwal and Nima Khademi Kalantari. Deep slow motion video reconstruction with hybrid imaging system. *PAMI*, 2020. 2

[25] 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 *CVPR*, pages 6820–6829, 2019. 2, 6, 7

[26] Jiahao Pang, Wenxiu Sun, JS Ren, Chengxi Yang, and Qiong Yan. Cascade Residual Learning: A Two-stage Convolutional Neural Network for Stereo Matching. In *ICCV*, pages 887–895, 2017. 4

[27] Federico Paredes-Vallés and Guido CHE de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. *CoRR*, 2020. 2

[28] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. *ECCV*, 2020. 1, 2, 5, 6, 7, 8

[29] Pytorch web site. http://http://pytorch.org/ Accessed: 08 March 2019. 5

[30] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. 2018. 5

[31] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, pages 3857–3866, 2019. 2, 5, 7

[32] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *TPAMI*, 2019. 2, 6

[33] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *WACV*, pages 156–163, 2020. 2

[34] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *ECCV*, 2020. 6

[35] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 2

[36] Bishan Wang, Jingwei He, Lei Yu, Gui-Song Xia, and Wen Yang. Event enhanced high-quality image recovery. *ECCV*, 2020. 2

[37] 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 *CVPR*, pages 10081–10090, 2019. 2

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[39] Zihao 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 *CVPR*, 2020. 5

[40] Zihao W Wang, Weixin Jiang, Kuan He, Boxin Shi, Aggelos Katsaggelos, and Oliver Cossairt. Event-driven video frame synthesis. In *ICCV Workshops*, pages 0–0, 2019. 2

[41] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *ECCV*, pages 416–431, 2018. 2

[42] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, pages 1647–1656, 2019. 2, 3

[43] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 127(8):1106–1125, 2019. 2, 5, 6, 7

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[45] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based optical flow using motion compensation. In *ECCV*, pages 0–0, 2018. 3