Supplementary Material - Time Lens: Event-based Video Frame Interpolation

Anonymous CVPR submission

Paper ID 3301

1. Video Demonstration

This PDF is accompanied with a video showing advantages of the proposed method compared to state-of-the-art frame-based methods published over recent months, as well as potential practical applications of the method.

2. Backbone network architecture

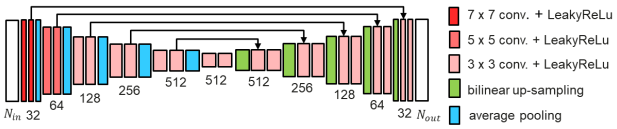


Figure 1: Backbone hourglass network that we use in all modules of the proposed method.

For all modules in the proposed method, we use the same backbone architecture which is an *hourglass network* with shortcut connections between the contracting and the expanding parts similar to [2] which we show in Fig. 1.

3. Additional Ablation Experiments

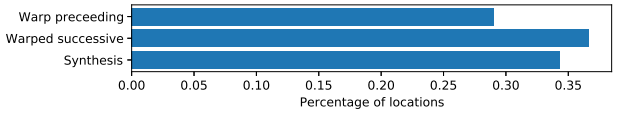


Figure 2: Percentage of pixels each interpolation method contributes on average to the final interpolation result for Vimeo90k (denoising) validation set. Note, that all methods contribute almost equally to the final result and thus are equally important.

Importance of inter-frame events. To study the importance of additional information provided by events, we skip every second frame of the original video and attempt to reconstruct it using two versions of the proposed method. One version has access to the events synthesized from the skipped frame and another version does not have inter-frame information. As we can see from the Tab. 1, the former significantly outperforms the later by a margin of 4.24dB. Indeed this large improvements can be explained by the fact that the method with inter-frame events has implicit access to the ground truth image it tries to reconstruct, albeit in the form of asynchronous events. This highlights that our network is able to efficiently decode the asynchronous intermediate events to recover the missing frame. Moreover, this shows that the addition of events has a significant impact on the final task performance, proving the usefulness of an event camera as an auxiliary sensor.

Table 1: Importance of inter-frame events on Middlebury test set. To compute SSIM and PSNR, we skip one frame of the original video, reconstruct it and compare to the skipped frame. One version of the proposed method has access to the events synthesized from the skipped frame and another version does not have inter-frame information. We also show performance of frame-based SuperSloMo method [2], that is used in event simulator for reference. We highlight the best performing method.

Method	PSNR	SSIM
With inter-frame events (ours)	**33.27±3.11**	**0.929±0.027**
Without inter-frame events	29.03±4.85	0.866±0.111
SuperSloMo [2]	29.75±5.35	0.880±0.112

Importance of each interpolation method. To study relative importance of *synthesis-based* and *warping-based* interpolation methods, we compute the percentage of pixels that each method contribute on average to the final interpolation result for the Vimeo90k (denoising) validation dataset and show the result in Fig. 2. As it is clear from the figure, all the methods contribute almost equally to the final result and thus are all equally important.

"Rope" plot. To study how the interpolation quality decreases with the distance to the input frames, we skip all but every 7th frame in the input videos from the High Quality Frames dataset, restore them using our method and compare to the original frames. For each skipped frame position, we compute average PSNR of the restored frame over entire dataset and show results in Fig. 3. As clear from the figure, the proposed method has the highest PSNR. Also, its PSNR decreases much slower than PSNR of the competing

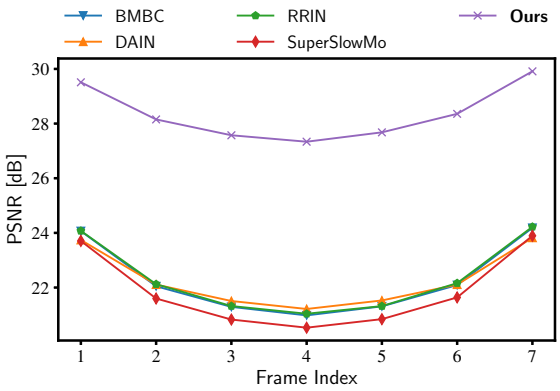


Figure 3: "Rope plot" showing interpolation quality as a function of distance from input boundary frames on High Quality Frames dataset. We skip all but every 7th frame and restore them using events and remaining frames. For each skip position, we compute average PSNR of the restored frame over entire dataset. We do not fine-tune the proposed and competing methods on the HQF dataset and simply use pre-trained models provided by the authors. Note, that the proposed method have the highest PSNR. Also, its PSNR decreases much slower than PSNR of other methods we move away from the input boundary frames.

methods as we move away from the boundary frames.

4. Additional Benchmarking Results

To makes sure that the fine-tuning does not affect our general conclusions, we fine-tuned the proposed method and RRIN method [4] on subset of High Quality Frames dataset and test them on the remaining part ("poster_pillar_1", "slow_and_fast_desk", "bike_bay_hdr" and "desk" sequences). We choose RRIN method for this experiment, because it showed good performance across synthetic and real datasets and it is fairly simple. As clear from the Tab. 2, after the fine-tuning, performance of the proposed method remained very strong compared to the RRIN method.

5. High Speed Events and RGB Dataset

In this section we describe the sequences in the High-Speed Event and RGB (HS-ERGB) dataset. The commercially available DAVIS 346 [1] already allows the simultaneous recording of events and grayscale frames, which are temporally and spatially synchronized. However, it has some shortcomings as the relatively low resolution of only 346×260 pixels of both frames and events. This is far below the resolution of typical frame based consumer cameras. Additionally, the DAVIS 346 has a very limited dynamic range of 55 db and a maximum frame of 40 FPS. Those properties render it not ideal for many event based methods, which aim to outperform traditional frame based cameras in certain applications. The setup described in [9] shows improvements in the resolution of frames and dynamic range, but has a reduced event resolution instead. The lack of publicly available high resolution event and color frame datasets and of the shelf hardware motivated the development of our dual camera setup. It features high resolution, high frame rate, high dynamic range color frames combined with high resolution events. A comparison of our setup with the DAVIS346[1] and the setup with beam splitter in [9] is shown in 3. With this new setup we collect new High Speed Events and RGB (HS-ERGB) Dataset that we summarize in Tab. 4. We show several fragments from the dataset in Fig. 5. In the following paragraphs we describe temporal synchronization and spatial alignment of frame and event data that we performed for our dataset.

Synchronization In our setup, two cameras are hardware synchronized through the use of external triggers. Each time the standard camera starts and ends exposure, a trigger is sent to the event camera which records an *external trigger event* with precise timestamp information. This information allows us to assign accurate timestamps to the standard frames, as well as group events during exposure or between consecutive frames.

Alignment In our setup event and RGB cameras are arranged in stereo configuration, therefore event and frame data in addition to temporal, require spatial alignment. We perform the alignment in three steps: *(i)* stereo calibration, *(ii)* rectification and *(iii)* feature-based global alignment. We first calibrate the cameras using a standard checkerboard pattern. The recorded asynchronous events are converted to temporally aligned video reconstructions using E2VID[6, 7]. Finally, we find the intrinsic and extrinsics by applying the stereo calibration tool Kalibr[3] to the video reconstructions and the standard frames recorded by the color camera. We then use the found intrinsics and extrinsics to rectify the events and frames.

Due to the small baseline and similar fields of view (FoV), stereo rectification is usually sufficient to align the output of both sensors for scenes with a large average depth ($>40\,\mathrm{m}$). This is illustrated in Fig. 4 (a).

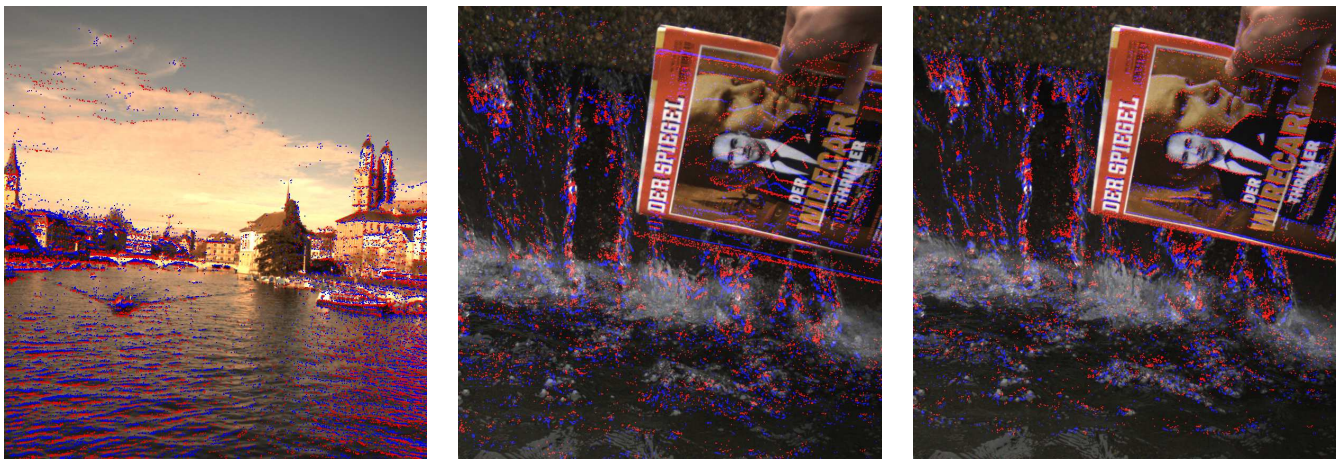
For close scenes, however, events and frames are misaligned (Fig. 4 (b)). For this reason we perform the second step of global alignment using a homography which we estimate by matching SIFT features [5] extracted on the standard frames and video reconstructions. The homography estimation also utilizes RANSAC to eliminate false matches. When the cameras are static, and the objects of interest move within a plane, this yields accurate alignment between the two sensors (Fig.4 (c)).

Table 2: Results on High Quality Frames [8] with fine-tuning. Due to the time limitations, we only fine-tuned the proposed method and RRIN [4] method, that performed well across synthetic and real datasets. For evaluation, we used "poster_pillar_1", "slow_and_fast_desk", "bike_bay_hdr" and "desk" sequences of the set and other sequences we used for the fine-tuning. For SSIM and PSNR, we show mean and one standard deviation across frames of all sequences.

Method	**1 skip**		**3 skips**	
	PSNR	**SSIM**	**PSNR**	**SSIM**
RRIN [4]	28.62±5.51	0.839±0.132	25.36±5.70	0.750±0.173
Time Lens (Ours)	**33.42±3.18**	**0.934±0.041**	**32.27±3.44**	**0.917±0.054**

Table 3: Comparison of our HS-ERGB dataset against publicly available High Quality Frames (HQF) dataset, acquired by DAVIS 346 [1] and Guided Event Filtering (GEF) dataset, acquired by setup with DAVIS240 and RGB camera mounted with beam splitter [9]. Note, that in contrast to the previous datasets, the proposed dataset has high resolution of event data, and high frame rate. Also, it is the first dataset acquired by dual system with event and frame sensors arranged in stereo configuration.

	Frames				Events			
	FPS	Dynamic Range, [dB]	Resolution	Color	Dynamic Range, dB	Resolution	Sync.	Aligned
DAVIS 346 [1]	40	55	346 × 260	✘	120	346 × 260	✔	✔
GEF[9]	35	60	**2480 × 2048**	✔	120	240 × 180	✔	✔
HS-ERGB (Ours)	**226**	**71.45**	1440 × 1080	✔	120	**720 × 1280**	✔	✔



(a) far away scenes (b) misaligned close scenes (c) after global alignment

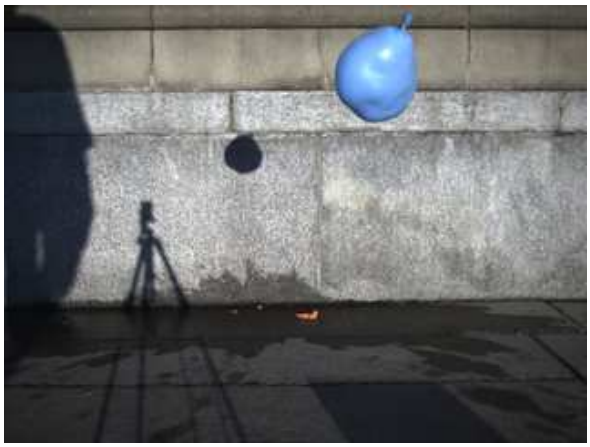
Figure 4: Alignment of standard frames with events. Aggregated events (blue positive, red negative) are overlain with the standard frame. For scenes with sufficient depth (more than $40\,\mathrm{m}$) stereo rectification of both outputs yields accurate per-pixel alignment (a). However, for close scenes (b) events and frames are misaligned. In the absence of camera motion and motion in a plane, the views can be aligned with a global homography (c).

Table 4: Overview of all sequences of the High Speed Event-RGB (HS-ERGB) dataset.

Sequence Name	Subset	Camera Settings	Description
Close planar sequences			
Water bomb air (Fig. 5a)	Train	163 FPS, 1080 µs exposure, 1065 frames	accelerating object, water splash
Lighting match		150 FPS, 2972 µs exposure, 666 frames	illumination change, fire
Fountain Schaffhauserplatz 1		150 FPS, 977 µs exposure, 1038 frames	illumination change, fire
Water bomb ETH 2 (Fig. 5c)		163 FPS, 323 µs exposure, 3494 frames	accelerating object, water splash
Waving arms		163 FPS, 3476 µs exposure, 762 frames	non-linear motion
Popping air balloon	Test	150 FPS, 2972 µs exposure, 335 frames	non-linear motion, object disappearance
Confetti (Fig. 5e		150 FPS, 2972 µs exposure, 832 frames	non-linear motion, periodic motion
Spinning plate		150 FPS, 2971 µs exposure, 1789 frames	non-linear motion, periodic motion
Spinning umbrella		163 FPS, 3479 µs exposure, 763 frames	non-linear motion
Water bomb floor 1 (Fig. 5d)		160 FPS, 628 µs exposure, 686 frames	accelerating object, water splash
Fountain Schaffhauserplatz 2		150 FPS, 977 µs exposure, 1205 frames	non-linear motion, water
Fountain Bellevue 2 (Fig. 5b)		160 FPS, 480 µs exposure, 1329 frames	non-linear motion, water, periodic movement
Water bomb ETH 1		163 FPS, 323 µs exposure, 3700 frames	accelerating object, water splash
Candle (Fig. 5f)		160 FPS, 478 µs exposure, 804 frames	illumination change, non-linear motion
Far-away sequences			
Kornhausbruecke letten x 1	Train	163 FPS, 266 µs exposure, 831 frames	fast camera rotation around z-axis
Kornhausbruecke rot x 5		163 FPS, 266 µs exposure, 834 frames	fast camera rotation around x-axis
Kornhausbruecke rot x 6		163 FPS, 266 µs exposure, 834 frames	fast camera rotation around x-axis
Kornhausbruecke rot y 3		163 FPS, 266 µs exposure, 833 frames	fast camera rotation around y-axis
Kornhausbruecke rot y 4		163 FPS, 266 µs exposure, 833 frames	fast camera rotation around y-axis
Kornhausbruecke rot z 1		163 FPS, 266 µs exposure, 857 frames	fast camera rotation around z-axis
Kornhausbruecke rot z 2		163 FPS, 266 µs exposure, 833 frames	fast camera rotation around z-axis
Sihl 4		163 FPS, 426 µs exposure, 833 frames	fast camera rotation around z-axis
Tree 3		163 FPS, 978 µs exposure, 832 frames	camera rotation around z-axis
Lake 4		163 FPS, 334 µs exposure, 833 frames	camera rotation around z-axis
Lake 5		163 FPS, 275 µs exposure, 833 frames	camera rotation around z-axis
Lake 7		163 FPS, 274 µs exposure, 833 frames	camera rotation around z-axis
Lake 8		163 FPS, 274 µs exposure, 832 frames	camera rotation around z-axis
Lake 9		163 FPS, 274 µs exposure, 832 frames	camera rotation around z-axis
Bridge lake 4		163 FPS, 236 µs exposure, 836 frames	camera rotation around z-axis
Bridge lake 5		163 FPS, 236 µs exposure, 834 frames	camera rotation around z-axis
Bridge lake 6		163 FPS, 235 µs exposure, 832 frames	camera rotation around z-axis
Bridge lake 7		163 FPS, 235 µs exposure, 832 frames	camera rotation around z-axis
Bridge lake 8		163 FPS, 235 µs exposure, 834 frames	camera rotation around z-axis
Kornhausbruecke letten random 4	Test	163 FPS, 266 µs exposure, 834 frames	random camera movement
Sihl 03		163 FPS, 426 µs exposure, 834 frames	camera rotation around z-axis
Lake 01		163 FPS, 335 µs exposure, 784 frames	camera rotation around z-axis
Lake 03		163 FPS, 334 µs exposure, 833 frames	camera rotation around z-axis
Bridge lake 1		163 FPS, 237 µs exposure, 833 frames	camera rotation around z-axis
Bridge lake 3		163 FPS, 236 µs exposure, 834 frames	camera rotation around z-axis

(a) Water bomb air

(b) Fountain Bellevue



(c) Water bomb ETH 2

(d) Water bomb floor 1

(e) Confetti

(f) Candle

Figure 5: Example sequences of the HS-ERGB dataset. This figure contains animation that can be viewed in Acrobat Reader.

References

[1] 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, 3

[2] 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. 1

[3] L. Kneip R. Siegwart L. Oth, P. Furgale. Rolling shutter camera calibration. In *CVPR*, 2013. 2

[4] Haopeng Li, Yuan Yuan, and Qi Wang. Video frame interpolation via residue refinement. In *ICASSP 2020*, pages 2613–2617. IEEE, 2020. 2, 3

[5] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, Nov. 2004. 2

[6] 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

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

[8] 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. 3

[9] 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. 2, 3