Supplementary Material — Time Lens++: Event-based Frame Interpolation with Parametric Non-linear Flow and Multi-scale Fusion

Stepan Tulyakov[1] Alfredo Bochicchio[1] Daniel Gehrig[2] Stamatios Georgoulis[1]
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

1. Dataset Licenses

In this work we use the HS-ERGB dataset [22] which is published under the "TimeLens Evaluation License" and can be found under the URL http://rpg.ifi.uzh.ch/timelens. The Vimeo90k dataset [26] is published under the following URL http://toflow.csail.mit.edu/.

2. Network structure

In this section, we provide a detailed description of our network. As explained in the paper, our network consist of *spline motion estimator* and *multi-scale feature fusion* modules. The spline motion estimator consists of 2 encoders and one joint decoder and the multi-scale fusion network consists of 4 encoders and one joint decoder. In both networks, encoders and decoders have the same structure shown in Fig. 2 and Fig. 1.

All decoder and encoder are defined by three parameters: depth D, the maximum number of features C_{max} and the base number of features C. We provide these parameters for the motion estimator module and multi-scale fusion module in Tab. 1.

Module	D	C_{max}	C
Fusion encoders & decoder	3	128	32
Motion estimator encoders & decoder	4	256	64

Table 1. Parameters of encoder and decoder for motion estimator and multi-scale fusion modules.

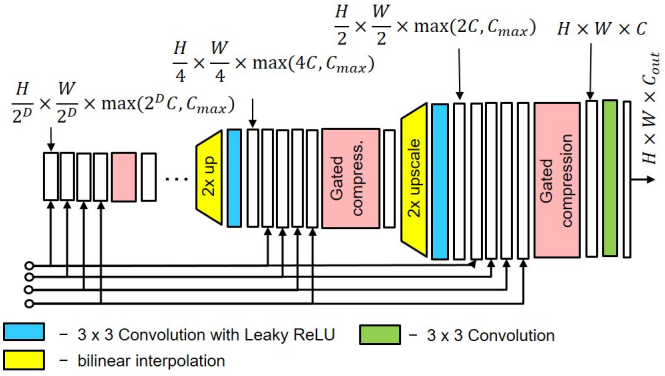

Figure 1. Architecture of joint decoder.

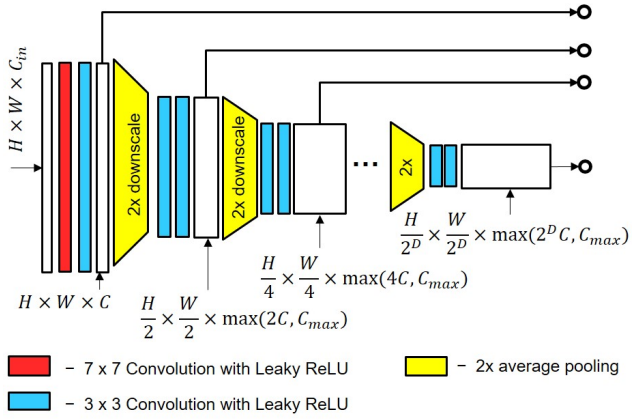

Figure 2. Architecture overview of encoder.

3. Training Details

In this section, we provide additional details about the training procedure.

Thresholded losses. As we mentioned in the paper, we use SSIM and L^1 losses to train the motion estimator. We found that it is beneficial to use these losses only in the areas where they are beyond a certain threshold (0.06 for L^1 and 0.4 for SSIM). Intuitively, this helps to exclude occluded areas and areas with brightness constancy violation from the loss computation.

Multi-stage training. We developed a multi-stage training procedure that improves the performance of the motion estimator (see Tab. 2) as well as the multi-scale fusion module (see Tab. 3). We use the same procedure for the motion encoder and the multi-stage fusion module since they share a similar high-level network structure. Firstly, we train the image and event-based encoders separately, each with its own "dummy" decoder. Secondly, we select sim-

ilar performing checkpoints for each encoder, we remove the dummy encoders and train joint decoder while freezing parameters of the encoders. Finally, we unfreeze all weights and train the entire network. This procedure helps when encoders have different training speeds and forces the network to avoid a local minimum where the decoder uses features only from the "faster" training encoder, ignoring the other. For example, the event-based motion encoder in the motion estimation module trains much faster than the image-based motion encoder, and thus using a single-stage training procedure would learn to simply ignore images and only use events. By contrast, our multi-stage training procedure forces the network to use both inputs.

4. Additional Ablations

Here we present ablations that we omitted from the main paper due to space limitations.

Method	SSIM	PSNR [dB]
Importance of Double Encoder		
U-Net	0.858	27.13
Double Enc. U-Net (ours)	**0.863**	**27.41**
Importance of Multistage Training		
Single stage	0.863	27.41
Multistage (ours)	**0.879**	**28.10**

Table 2. Additional ablation for motion estimation module.

Method	SSIM	PSNR [dB]
Importance of shared synthesis encoder		
Joined	0.911	31.74
Shared (ours)	**0.912**	**31.87**
Importance of Multistage Training		
Single stage	0.912	31.87
Multistage (ours)	**0.919**	**32.73**

Table 3. Additional ablation studies for fusion module.

Motion estimator. Here we summarize additional ablations for the motion estimator in Table 2. We found that using two separate encoders in the motion module improves results by 0.23 dB, by leaving the encoders more freedom to compute separate features for events and frames. Additionally, we found that by conducting multi-stage training, we can further improve the performance of our module by 0.69 dB. Multistage training consists of firstly training each encoder separately, when freezing encoders and train decoder, and finally training the entire motion module.

Multi-scale feature fusion. We found that, by using shared synthesis encoders, we can get a 0.13 dB improvement, and, by applying multi-stage training, we can further boost performance by 1.06 dB in Tab. 3. Additionally, in Fig. 3 we show the average attention weights predicted by our fusion network for synthesis and warping interpolation features on each scale. It is clear that the fusion network uses both synthesis and warping interpolation features, but prefers synthesis features on coarse scales and warping features on fine scales. This is consistent with observations in Time Lens [27].

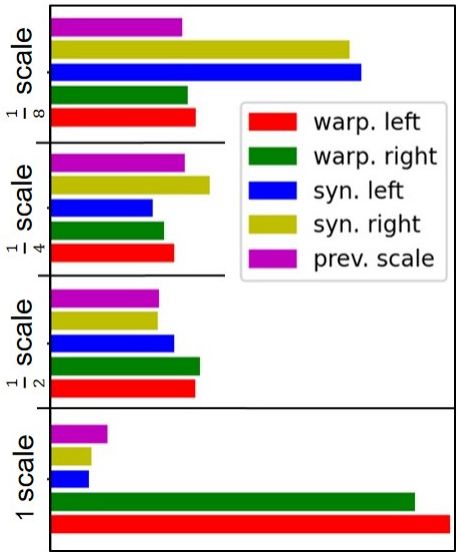

Figure 3. Average attention weights predicted by fusion network for synthesis and warping interpolation features on each scale.

5. Beamsplitter Setup and Dataset

We built an experimental setup with a global shutter RGB Flir 4096×2196 camera and a Prophesee Gen4M 1280×720 event camera [1] arranged with the beam splitter as shown in Fig. 5.

In our setup, the 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.

We use a standard stereo rectification procedure using images and event reconstructions from E2VID [2] to physically align the event and frame cameras and achieve a baseline of around 0.6mm. Next, we set the same focus for the event and frame camera and match the resolution of the RGB camera to the event camera by downsampling the images to a resolution of 1280×720. Next, we calibrate each camera separately to estimate the lens distortion parameters and focal lengths. After removing the lens distortion, we estimate a homography that compensates for the misalignment between events and images. While this procedure removes misalignment for most of the scenes, small pixel misalignment still occurs for very close scenes due to the residual baseline. We automatically compensate for

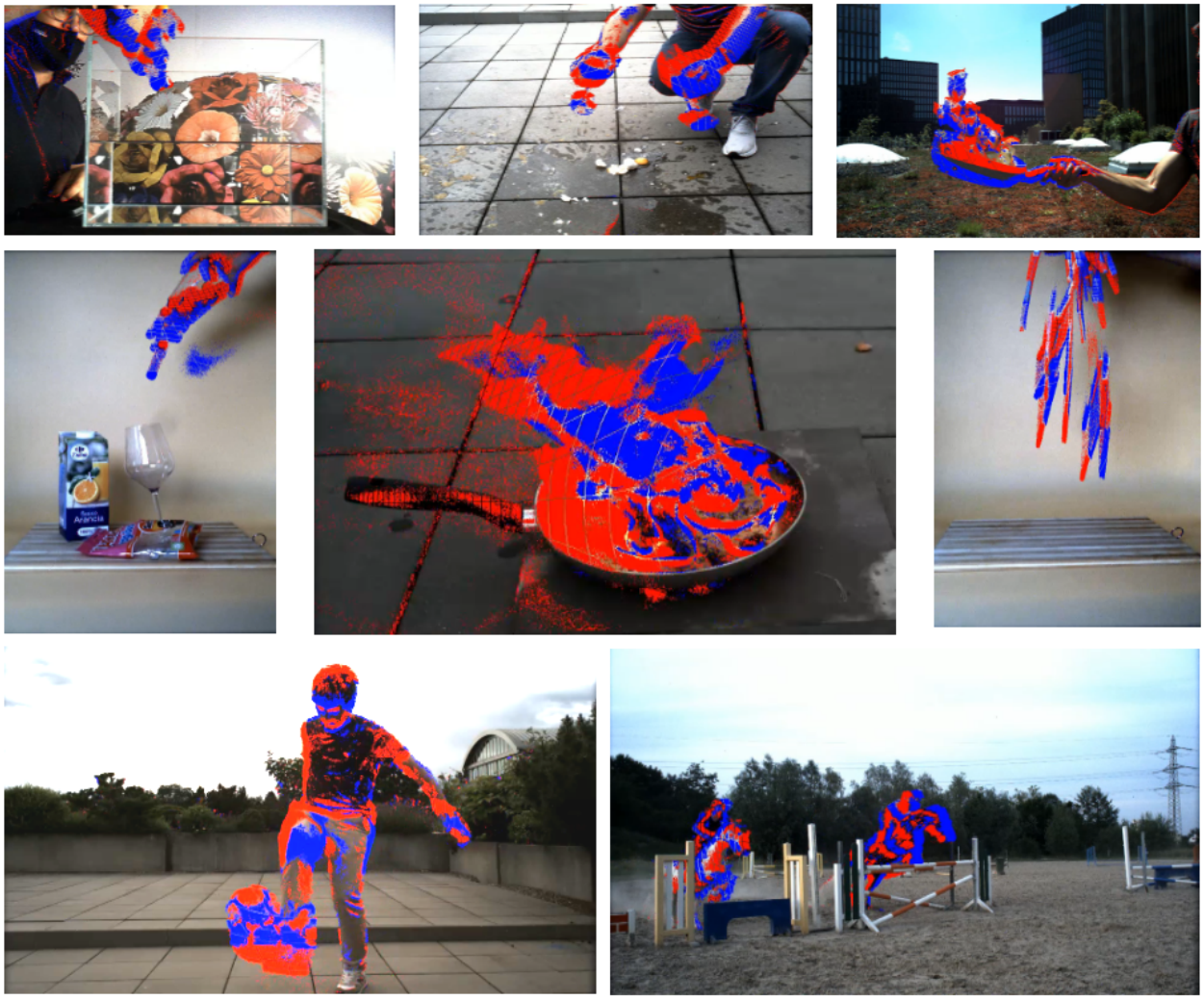

Figure 4. The dataset includes challenging scene with water, color liquid, objects and liquid, small colorful objects, fire, thin colorful objects, rotating objects on a moving background, eye catching scene (from top-left to bottom-right)

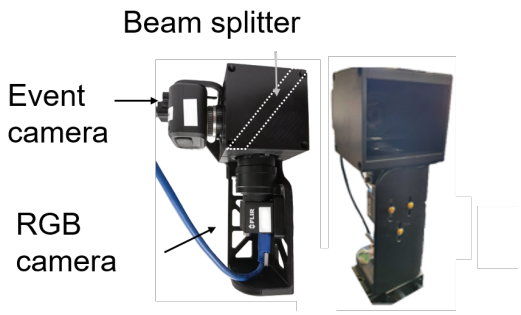

Figure 5. The beamsplitter setup mounts a FLIR RGB global shutter camera and a Prophesee Gen4 event camera [1] mounted on a case with a 50R/50T beam splitter mirror that allows the sensors to share a spatially aligned field of view. view

this misalignment for each sequence using a small global x-y shift, computed by maximizing the cross-correlation of event integrals and temporal image difference. Finally, after image acquisition, we adjust the brightness and contrast of the images. Using the procedure above, we collected 123 diverse and challenging scenes with varying depth, some of which we show in Fig. 4. The dataset is divided into 78 training scenes, 19 validation scenes, and 26 test scenes.

References

[1] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, Hirotsugu Takahashi, Hayato Wakabayashi, Yusuke Oike, and Christoph Posch. A 1280x720 back-illuminated

stacked temporal contrast event-based vision sensor with 4.86μm pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline. In *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2020. 2, 3

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