1000 FPS HDR Video with a Spike-RGB Hybrid Camera

Yakun Chang[1,2] Chu Zhou[3] Yuchen Hong[1,2] Liwen Hu[2] Chao Xu[3] Tiejun Huang[1,2] Boxin Shi[1,2*]

[1] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University

[2] National Engineering Research Center of Visual Technology, School of Computer Science, Peking University

[3] National Key Laboratory of General AI, School of Intelligence Science and Technology, Peking University

{yakunchang, zhou_chu, huliwen, tjhuang, shiboxin}@pku.edu.cn

yuchenhong.cn@gmail.com, xuchao@cis.pku.edu

Abstract

Capturing high frame rate and high dynamic range (HFR&HDR) color videos in high-speed scenes with conventional frame-based cameras is very challenging. The increasing frame rate is usually guaranteed by using shorter exposure time so that the captured video is severely interfered by noise. Alternating exposures can alleviate the noise issue but sacrifice frame rate due to involving long-exposure frames. The neuromorphic spiking camera records high-speed scenes of high dynamic range without colors using a completely different sensing mechanism and visual representation. We introduce a hybrid camera system composed of a spiking and an alternating-exposure RGB camera to capture HFR&HDR scenes with high fidelity. Our insight is to bring each camera's superiority into full play. The spike frames, with accurate fast motion information encoded, are firstly reconstructed for motion representation, from which the spike-based optical flows guide the recovery of missing temporal information for long-exposure RGB images while retaining their reliable color appearances. With the strong temporal constraint estimated from spike trains, both missing and distorted colors cross RGB frames are recovered to generate time-consistent and HFR color frames. We collect a new Spike-RGB dataset that contains 300 sequences of synthetic data and 20 groups of real-world data to demonstrate 1000 FPS HDR videos outperforming HDR video reconstruction methods and commercial high-speed cameras.

1. Introduction

The spiking camera [17] and event camera [10] are neuromorphic sensors working differently from conventional frame-based digital cameras, which have many attractive characteristics, *e.g.*, high-speed (perceiving scene radiance changes at the microsecond level), high dynamic range (HDR, $\geq$ 100 dB). However, since they only record neuromorphic signals, *i.e.*, spike trains [64] and event streams [25], which are less friendly to the human visual system and cannot be directly processed by CNN-based models for video frames [40, 41], preprocessing modules that convert neuromorphic signals into compatible formats are usually required when applying them to frame-based vision algorithms [61, 65]. In comparison with event streams, spike trains contain concrete textured information of scene radiances, which are more suitable for reconstructing high frame rate (HFR) videos [61–64]. However, since the spiking camera only encodes the absolute intensities of environments, colors are absent in the reconstructed video frames.

When capturing with a frame-based RGB camera, quality of recorded colors for each frame is determined by trading off the exposure time, ambient light, and target objects' moving speed [57]. For high-speed dynamic scenes, it often

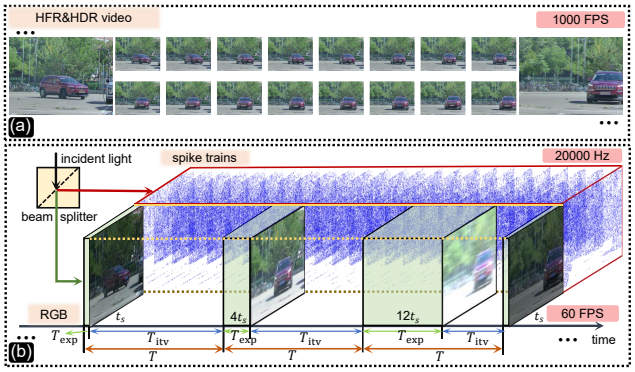

Figure 1. (a) We build a spike-RGB hybrid camera system to achieve 1000 FPS HDR video reconstruction[1]. (b) The RGB camera uses alternating-exposure mode with a frame rate of 60 FPS, where t_s, $4t_s$, and $12t_s$ are the short, middle, and long exposure in our setup, respectively. The sampling frequency of the spiking camera is 20000 Hz.

*Corresponding author.

Project page: https://changyakun.github.io/1000FPS-HDR

[1] The video result is available on our project page.

requires to set shorter exposure time to guarantee a higher frame rate and avoid motion blur. In such a situation, since the exposure time is extremely short, the quality of video frames would be severely degenerated due to noise. Merging a burst of short-exposure images is a simple yet effective approach to reduce the noise level [8, 11], however, the color shift caused by noise is difficult to be corrected. Fusing alternating-exposure (using short, middle, and long exposures) RGB frames is commonly used for synthesizing well-exposed images [3, 19, 21]. However, they are not suitable for high-speed scenes. As illustrated in Fig. 1(b), given a sequence of alternating-exposure RGB images, the total time from the starting of the current exposure to the starting of the next frame, denoted by T, is consistent for all frames, and it is composed of the *exposure time* T_{exp} and *interval time* T_{itv} (containing the readout and waiting time). It can be seen that the information during interval time is lost, and the frame rate they could achieve is thus limited to dozens of FPS. Another possible solution is to build a hybrid camera system to capture low frame rate (LFR) color sequence and high-speed neuromorphic signals simultaneously, then use the neuromorphic signals to interpolate [51, 52] and deblur [14, 18, 59] the RGB frames. However, the saturated regions are usually ignored, leaving the colors of the interpolated frames still unsatisfactory. HDR intensity map (does not contain any chromatic information) built from the neuromorphic signals can also be used to compensate the missing textures in the saturated regions [15]. But such an approach is not robust for scenes with large areas of saturated regions, due to the heavy reliance on the chrominance compensation network to hallucinate the color.

In this paper, we propose an all-in-one framework to reconstruct HRF (Fig. 1(a), at the level of 1000 FPS) color videos with high fidelity from the spike trains and a series of alternating-exposure frames captured by a Spike-RGB hybrid camera system simultaneously (Fig. 1(b)). To make full use of the color information in RGB images, we propose a three-stage strategy to deal with different situations using specific modules: (*i*) For the blurry middle- and long-exposure images, we design a spike guided deblurring module to recover the corresponding sharp images with faithful colors; (*ii*) for missing colors during the interval time, we design a spike guided interpolation module that exploits the abundant motion information (SC-Flow [16]) obtained from spike trains; (*iii*) for suppressing noise in short-exposure images and maintaining temporal consistency, we design a merging module, which exploits the variant of recurrent U-Net [42] as its backbone, to complete the HFR&HDR color video reconstruction process. To summarize, this paper makes contributions by proposing:

- an all-in-one framework to reconstruct high-speed HDR color video by jointly fusing spike trains and a sequence of alternating-exposure frames;
- a three-stage strategy fusing alternating exposures of RGB frames for the generation of well-exposure colors, via a recurrent convolution neural network for continuous frames interpolation guided by spike trains;
- a Spike-RGB hybrid camera system to demonstrate the applicability of the proposed method for capturing high-speed and high dynamic range scenes.

Experimental results show that the proposed method outperforms the state-of-the-art HDR video reconstruction method [3] and commercial cameras with the slow-motion photography capability in reconstructing 1000 FPS HDR color videos on synthetic data and real-world data.

2. Related Work

HDR image and video reconstruction. The most common way to reconstruct HDR images is to fuse a set of LDR images with bracketed exposures [7, 34]. Since the results for dynamic scenes often contain ghosting artifacts, image alignment [28, 45] and deep learning [20, 55] are employed to reconstruct sharp HDR images. To better reduce ghosting artifacts, Lee *et al*. [24] and Shaw *et al*. [46] apply the estimated motion information from a high frame rate sequence to facilitate the HDR image synthesis. Messikommer *et al*. [35] also achieve HDR reconstruction by combining bracketed-exposure RGB images and events. There are methods being designed for HDR reconstruction from a single image. These methods cannot recover the missing textures in clipped regions [9, 44]. Abhiram and Chan [1] reconstruct HDR images with a quanta image sensor (QIS). Han *et al*. [15] find that the reconstructed intensity maps from event streams and spike trains contain abundant textures saturated in LDR images. Therefore, they exploit intensity maps to guide HDR image restoration. For the capturing of HDR videos, many existing methods use specialized hardware, such as scanline exposure [13], per-pixel exposure [37], or multiple sensors [33, 50]. Due to the particularity of hardware, these methods are limited to narrow applications. Merging alternating-exposure image sequences is the most common yet effective way to reconstruct HDR videos [12, 19, 21, 22, 30, 31]. Recently, Chen *et al*. [3] propose a coarse-to-fine network that performs alignment and fusion sequentially both in the image and feature space. However, these methods can only deal with LFR videos with about 20-60 FPS.

HFR video reconstruction. There is plenty of data redundancy in capturing HFR videos directly by commercial high-speed cameras, *e.g*., the Phatom camera[2]. Building a hybrid system with a high-resolution LFR camera and a low-resolution HFR camera, and utilizing HFR signals to reconstruct a sequence of sharp images from blurred images [2, 49] is a more data-efficient way for HFR video

[2] https://www.phantomhighspeed.com/

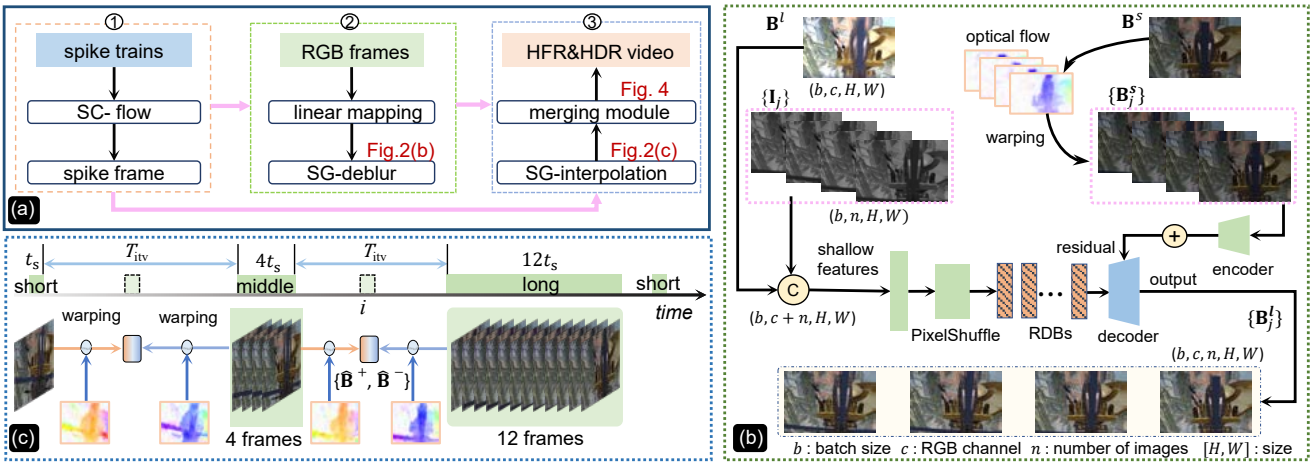

Figure 2. (a) The pipeline of the proposed solution. It contains three steps: Step ① spike preprocessing (Sec. 3.2), Step ② RGB frame processing (Sec. 3.3), and Step ③ merging into HFR video (Sec. 3.4). Given the spike trains, we firstly estimate the optical flow from them as well as reconstruct spike frames. Secondly, we rectify the uneven brightness with a linear mapping function and use spike-guided deblurring (SG-deblur) to reconstruct sharp color frames. Finally, we use spike-guided frame interpolation (SG-interpolation) to recover the missing colors during T_{itv}, and reconstruct time-consistent color frames. (b) and (c) show the detailed pipeline of SG-deblur and SG-interpolation.

reconstruction. Li *et al*. [26] use a stereo pair of low-resolution HFR and high-resolution LFR cameras to calculate the fast motion and the depth map. Avinash *et al*. [38] compute optical flows between two existing frames by utilizing the content of auxiliary HFR videos. Jiang *et al*. [18] recover a sharp video sequence from a motion-blurred image by integrating the visual and temporal knowledge that is contained in the events. Xu *et al*. [54] achieve real-world event-based deblurring with a self-supervised learning method. Tulyakov *et al*. [52] propose the Time Lens that utilizes high-speed events to achieve video frame interpolation (VFI). Following that, Time Lens++ [51] further improves the performance. For the reason that real data are absent, Yu *et al*. [56] propose a weakly supervised method with the help of subpixel attention learning. Although the event-based interpolation realizes HFR video reconstruction [51, 52], the recovered quality of colors is usually unsatisfactory due to that single exposure cannot balance artifacts from noise and blur, we therefore propose to jointly fuse the high-speed spike signals and alternating-exposure RGB frames to achieve high-quality reconstruction.

3. Approach

3.1. Overview

Our goal is to reconstruct HFR&HDR videos from the binary spike trains $\mathbb{S}(x, y) = \{s(x, y, t)\}$ ($s(x, y, t) = 1$ if the accumulated photons reach a certain threshold, then the accumulator is reset and $s(x, y, t) = 0$ before the next spike is fired [17]) and LFR alternating-exposure RGB frames $\mathbb{B} = \{\mathbf{B}_k\}$[3], where (x, y) denote the coordinates of spikes, t denotes the timestamp, and k denotes the index of an RGB image in the sequence. As shown in Fig. 2(a), to achieve this goal, we design a pipeline that consists of three steps:

Step ①: Spike preprocessing (Sec. 3.2). We estimate the optical flow $\mathbf{F}_i$ and spike frames $\mathbf{I}_i$ from the spike trains:

$$\mathbf{F}_i(x, y) = \mathcal{SC}(s(x, y, t_i \rightarrow t_{i+1})), \tag{1}$$

$$\mathbf{I}_i(x, y) = \int_{t_i - t_f/2}^{t_i + t_f/2} s(x, y, t)dt, \tag{2}$$

where $\mathcal{SC}(\cdot)$ denotes optical flow estimation with Hu *et al*.'s [16] method, i and t_i denote the index and timestamp of spike frames, and t_f is the time window. In Sec. 3.2, we further super-resolve $\mathbf{I}_i$ at the feature space.

Step ②: RGB frame preprocessing (Sec. 3.3). For the 60 FPS RGB images captured with alternating exposures, *i.e*., t_s, $4t_s$, and $12t_s$, we firstly unify the uneven brightness with a linear mapping function. Then we conduct motion deblurring for $4t_s$ and $12t_s$ images. For the t_s images, when t_s is sufficiently short, *i.e*., 1 ms, we assume the short-exposure image is free from motion blur, and take t_s as the reference time for the motion deblurring. Consequently, we can recover 4 and 12 sharp images from $4t_s$ and $12t_s$ images, respectively. As shown in Fig. 2(b), we use $\mathbf{B}^l$ to denote a blurry image, and the motion deblurring operation can be formulated as: $\{\mathbf{B}^l_j\} = \mathcal{R}(\mathbf{B}^l, \{\mathbf{I}_j | j \in \mathcal{N}_l\}, \mathbf{B}^s)$, where j is the index of a recovered sharp image, $\mathcal{R}(\cdot)$ is sharp image reconstruction, $\{\mathbf{I}_j | j \in \mathcal{N}_l\}$ is the corresponding spike frames, and $\mathbf{B}^s$ is the nearest short-exposure RGB frame.

Step ③: Merging into HFR video (Sec. 3.4). Following Step ②, for the interval time (T_{itv}) that colors are not recorded, we bidirectionally query two nearest sharp RGB

[3] In this paper, we use $\{\cdot\}$ to denote collections.

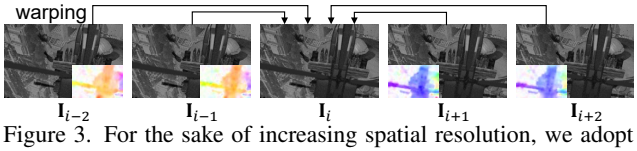

Figure 3. For the sake of increasing spatial resolution, we adopt flow-based warping to merge adjacent 5 spike frames.

images $\{\mathbf{B}_i^+, \mathbf{B}_i\}$ for each spike frame $\mathbf{I}_i$, and get the warped images $\{\hat{\mathbf{B}}_i^+, \hat{\mathbf{B}}_i\}$ with optical flow, where $+$ and $-$ denote the forward and backward warping, respectively. In Fig. 2(c), we provide an illustration of the interpolation procedure. Finally, as shown in Fig. 4, we reconstruct time-consistent color frames, and each frame $\mathbf{C}_i$ is generated by merging the spike frame $\mathbf{I}_i$ with $\{\mathbf{C}_{i-1}, \hat{\mathbf{B}}_i^+, \hat{\mathbf{B}}_i\}$ with the strong constraint of optical flow.

3.2. Spike preprocessing

The optical flow estimation and spike frame reconstruction using in Eqn. (1) and Eqn. (2) are theoretically, yet the reconstructed frames practically have two issues: Since the integration time t_f is very short, noise is relatively strong; the spatial resolution of the first generation spiking camera (VidarOne [17]) is much lower than the RGB camera. To reduce the noise and increase the spatial resolution, inspired by the burst-based super-resolution [4] and denoising [27] for conventional RGB images, it is feasible to merge a group of adjacent spike frames with the help of spatial alignment. Moreover, thanks to the continuous motion recording capability of spiking cameras, the optical flow [16] estimated from spike trains makes the alignment even more stable than RGB images. As illustrated in Fig. 3, we design a computationally efficient module for spike frames, which is formulated as: $\hat{\mathbf{I}}_i = \{\mathcal{W}_{\mathbf{F}_{j\to i}}(\mathbf{I}_j) | j \in \mathcal{N}_i\}$, where $\mathcal{W}_{\mathcal{F}_{j\to i}}(\cdot)$ denotes the flow-based warping operation, $\mathcal{N}_i$ denotes a collection of adjacent frames. Then, we feed $\hat{\mathbf{I}}_i$ to a set of convolutional layers, and we use PixelShuffle [47] to increase the spatial resolution while decreasing the channel of features. It should be noted that the method for spike frame reconstruction is not unique, which means users can choose other learning-based methods [61, 62, 64]. However, those deep learning models are relatively heavy, and less efficient as a submodule fitting to our pipeline.

3.3. RGB image preprocessing

RGB linear mapping. Following previous methods for HDR video reconstruction [3, 19, 21], we first unify the brightness of alternating-exposure RGB frames. Since we use an industrial camera (details in Sec. 3.5) that can acquire data without a nonlinear radiometric response function, the linearity of the captured frames is maintained. We find that the brightness of the frames can maintain a linear relationship with the duration of exposure time. Hence we use the global linear mapping to unify the frame brightness: $\alpha \cdot \mathbf{B}_k(x, y) \to \mathbf{B}_k(x, y)$, where α denotes a linear scalar.

Spike-guided deblurring. The physical model of the blurring process can be simply formulated as the average of a group of sharp images, *i.e.*, $\mathbf{B}^l(x, y) = \frac{1}{N}\sum_{j=1}^{N}\mathbf{B}_j^l(x, y)$, where N denotes the number of sharp images. However, due to the limited dynamic range of the RGB camera, that simplified equation does not hold in the clipped regions of real-world long-exposure frames. In general we should have: $\mathbf{B}^l(x, y) \le \frac{1}{N}\sum_{j=1}^{N}\mathbf{B}_j^l(x, y)$. Therefore, for reconstructing a sequence of sharp HDR images from $\mathbf{B}^l$, we divide it into two sub-tasks: (*i*) For the well-exposure regions, we use the sharp spike frames to guide motion deblurring; (*ii*) for the clipped regions where colors are lost, we compensate them with well-retained colors extracted from the adjacent short-exposure image $\mathbf{B}^s$.

Figure 2(b) shows the spike-guided deblurring (SG-deblur) from $\mathbf{B}_l$ ($\mathbf{B}_l$ may be a middle- or long-exposure image). Similar to Xu *et al.* [54] that exploit event frames to motion deblurring, we first concatenate $\mathbf{B}_l$ with $\{\mathbf{I}_l^j\}$, then extract shallow features and increase feature channels with PixelShuffle [47], which is followed by a set of residual dense blocks (RDBs) [60] and a decoder. To make the colors in over-exposure regions be compensated by the adjacent short-exposure RGB image $\mathbf{B}_j^s$, we warp the short-exposure image with the optical flow estimated from spike trains: $\mathbf{B}_j^s = \mathcal{W}_{\mathbf{F}_{s\to j}}(\mathbf{B}^s)$, where $\mathcal{W}_{\mathbf{F}_{s\to j}}(\cdot)$ denotes the warping operation from timestamp t_s to the timestamp of t_j. Subsequently, we extract features from $\{\mathbf{B}_l^{s\to j}\}$ and add residual links between them and the decoder. Finally, we obtain a sequence of sharp color images. Note that the SG-deblur for the middle- and long-exposure RGB images share the same architecture while the parameters are not shareable. SG-deblur outputs four images for both $4t_s$ and $12t_s$ frames. For the case of $12t_s$ frame, we interpolate the 4 frames to 12 frames with flow-based warping.

Next, we briefly explain the reason why this event-based model [54] can be applied to a spike-based task. Both event streams and spike trains with the high-speed property have been used for motion deblurring and latent frame reconstruction [14, 18, 54]. It is necessary to convert them to event frames and spike frames, both of which belong to the category of 2D images. But event frames and spike frames have different physical meanings: Pixel values in an event frame reveal the residual (relatively sparse information) between two adjacent frames, while pixel values in a spike frame represent exactly the texture (relatively dense information) of the corresponding frame. Since both event frames and spike frames are 2D images and the spike frames have denser texture information, we can replace event frames in such a model with spike frames, so as to make the solution to the problem more well-posed.

3.4. Merging into HFR video

RGB interpolation. Given each middle- and long-exposure

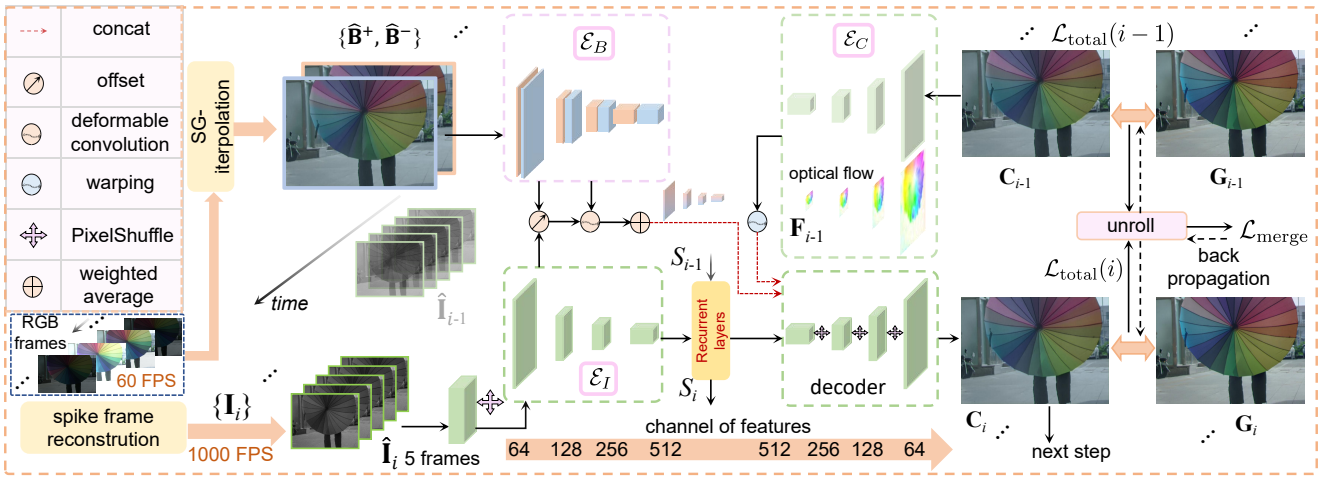

Figure 4. Network architecture of the CNN-RNN-based merging module for reconstructing HFR&HDR videos from alternating-exposure RGB frames and HFR spike frames. This module outputs HDR color frames in a step-wise manner. We unroll the module for M steps during training.

frame, SG-deblur recovers 4 and 12 images. Therefore, the recovered RGB frames have a frame rate of 340[4] FPS. But temporal distribution of them is quite uneven, *e.g.*, there is no recovered color frame interval time T_{itv}. Fortunately, the spike train contains continuous and dense texture information in the temporal domain. In Step ③, we use the SG-interpolation module to interpolate RGB frames into a sequence of uniformly distributed images. For each spike frame $\mathbf{I}_i$, we bidirectionally query its two nearest recovered RGB frames $\{\mathbf{B}_i^+, \mathbf{B}_i\}$ and interpolate two color frames $\{\hat{\mathbf{B}}_i^+, \hat{\mathbf{B}}_i\}$ with the optical flow estimated from spike trains. When $\{\hat{\mathbf{B}}_i^+, \hat{\mathbf{B}}_i\}$ are fed into our merging module, they are weighted by a linear coefficient ($\oplus$ in Fig. 4) related to the distance between t_i and $\{t_+, t\}$, where $\{t_+, t\}$ denote the timestamp of $\{\hat{\mathbf{B}}_i^+, \hat{\mathbf{B}}_i\}$.

Merging module. The aforementioned modules reconstruct coarse HFR video frames, which need to be refined for smoothing over time. We build a CNN-RNN-based HFR&HDR video reconstruction network to merge the spike frames and RGB frames, which is shown in Fig. 4. The merging module consists of three encoders, *i.e.*, $\mathcal{E}_I$, $\mathcal{E}_B$, and $\mathcal{E}_C$, which are respectively designed for feature extraction from the current spike frame $\hat{\mathbf{I}}_i$, the interpolated RGB images $\{\hat{\mathbf{B}}_i^+, \hat{\mathbf{B}}_i\}$, and the previously reconstructed image $\mathbf{C}_{i-1}$. In $\mathcal{E}_I$, we use PixelShuffle [47] to make the spatial resolution of spike features consistent with RGB features. The extracted features are denoted as $\mathbf{E}_I$, $\{\mathbf{E}_B, \mathbf{E}_{B+}\}$, and $\mathbf{E}_{C_{i-1}}$, respectively.

Considering the spike frames and RGB frames may not be perfectly aligned at pixel level for real-world data, we add deformable convolution layers [6] to improve the robustness to this issue. In order to output flicker-free color frames, we adopt two constraints in the merging module: (*i*) We add three ConvLSTM layers [48] to feed previous states forward in temporal domain; (*ii*) we feed $\mathbf{E}_{C_{i-1}}$ into the current step and align it with the current features with flow-based warping. We then use a decoder to reversely map deep features to the current output HDR frame $\mathbf{C}_i$. We achieve the multi-module signal fusion by adding concatenation links between $\{\mathbf{E}_{C_{i-1}}, \mathbf{E}_B, \mathbf{E}_{B+}\}$ and the decoder.

[4]From 60=20×3 to 340=20×(1+4+12).

Table 1. Details of the composition of the dataset (res. is the abbreviation of resolution).

data	RGB res.	spike res.	train/test	time
full-synthetic	500×800	250×400	80/20	0.1s
real-synthetic	600×800	250×400	160/40	0.101s
real-world	484×784	242×392	-/20	0.101s

3.5. Implementation Details

Due to the setting of our method being different from existing HDR and video frame interpolation methods, there are no suitable datasets for training and testing our method. Therefore, we collect a new one with three components, whose details are summarized in Table 1 and sample images are provided in Fig. 5.

Part 1: Full-synthetic data. This part of data is obtained by using the spike simulator proposed by Hu *et al.* [16]. We render 2000 RGB images with their computer graphics based solution as ground truth and generate 2000 spike planes (0.1 s). Since the photons arriving at the sensor follow Poisson probability distribution [43], we synthesize alternating-exposure 60 FPS RGB frames with a Poisson noise model. For the full synthetic data, we randomly select starting time of each group of training data. We randomly shift the RGB frames within 3 pixels to make the trained model more robust to the misalignment in real-world data.

Part 2: Real-synthetic data. To reduce the domain gap between full-synthetic data and real-world data, we design a method to collect real-synthetic (the scenes are real while

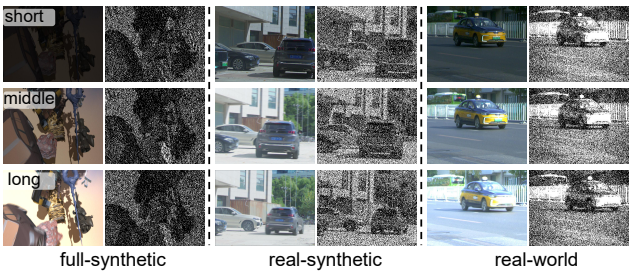

Figure 5. Example frames from the proposed dataset. Each group shows three alternating-exposure RGB frames (left, from top to bottom rows) and the corresponding spike signals (right).

the spike trains are synthetic) data, and we use this part of data to fine-tune our model. The RGB frames are captured with an alternating-exposure mode in slow-motion scenes. Then we synthesize blurry middle-exposure RGB frames by averaging 4 adjacent middle-exposure RGB images, and blurry long-exposure RGB frames are synthesized in a similar way. We synthesize spike trains from ground truth RGB frames with the integrate-and-fire methodology [61].

Part 3: Real-world data. We build a Spike-RGB hybrid camera (Fig. 6) to capture real-world data. The system is composed of an industrial camera (Basler acA800-510uc[5]) with alternating exposure capability and a spiking camera [17]. There is a beam splitter in front of the two sensors. We conduct geometric calibration and time synchronization to align bimodal signals collected by them.

Loss and training. The SG-deblur module and the merging module reconstruct images in the linear luminance domain, which covers a high dynamic range of pixel values. Following existing methods for HDR reconstruction, for the output images $\mathbf{C}$, we compress the range of pixel values by applying the following function proposed by Kalantari *et al*. [20]: $\mathcal{T}(\mathbf{C}) = \log(1+\mu\mathbf{C})/\log(1+\mu)$, where $\mathcal{T}(\cdot)$ denotes the tone mapping operation and μ denotes the amount of compression. For these two modules, we employ widely used l_1 loss, Structure similarity (SSIM) loss [53], and Learned Perceptual Image Patch Similarity (LPIPS) loss [58]. The total loss at step i for both the motion deblurring and merging modules is

$$\mathcal{L}_{\text{total}}(i) = \mathcal{L}_{l_1}(i) + \beta_1 \mathcal{L}_{\text{SSIM}}(i) + \beta_2 \mathcal{L}_{\text{LPIPS}}(i), \quad (3)$$

where $\beta_1 = 1$ and $\beta_2 = 1$. For spike-based optical flow estimation using [16], we fine-tune the parameters with full-synthetic data. During training, we resize the RGB images and spike frames to 512×800 and 256×400. We implement our model with PyTorch, set the batch size to 4, and use ADAM optimizer during the training process. We first train the model on full-synthetic data. The SG-deblur module is trained with 50 epochs, before training the merging module. We unroll the merging module for M steps, and we find $M = 4$ achieves a suitable balance between training time and recovery quality. The total loss for the unrolled M steps is $\mathcal{L}_{\text{merge}} = \sum_{i=1}^{M} \mathcal{L}^{\text{M}}_{\text{total}}(i)$, where $\mathcal{L}^{\text{M}}_{\text{total}}(i)$ denotes the total loss for the merging module at step i. The initial learning rate for both two modules is 0.001, we decay it to 10^{-6} with a linear strategy. For the real-synthetic data, we fine-tune another group of parameters to reduce the gap between synthetic data and real-world data. We use one NVIDIA Tesla A100 for training, and the training procedure consumes about 30 hours.

[5] https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca800-510uc/

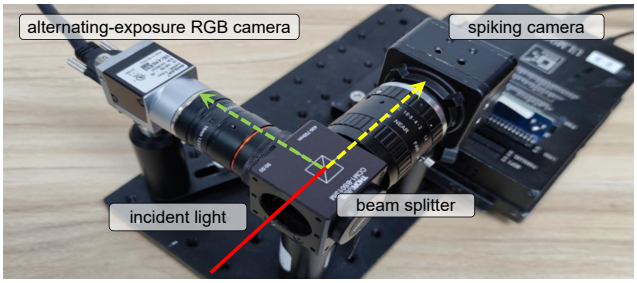

Figure 6. The prototype of our Spike-RGB imaging system composed of a spiking camera and an RGB camera.

4. Experiments

4.1. Quantitative Evaluation using Synthetic Data

Validation on full-synthetic data. Figure 8 shows a group of results on full-synthetic data. We can see that both the flying objects in the short-exposure image and the oversaturated clouds (see the regions marked by boxes) in the long-exposure image are recovered successfully. The results with rich textures and consistent colors show the feasibility of our proposed method.

Evaluation on real-synthetic data. To the best of our knowledge, the proposed method is the first framework to reconstruct HFR&HDR videos with the combination of spike trains and alternating-exposure RGB frames. Therefore, it is unfair to compare our method with existing ones, *i.e*., Kalantari13 [21], Kalantari19 [19], and Chen21 [3][6], which are designed for low frame rate HDR videos.

We choose a state-of-the-art HDR video reconstruction method Chen21 [3], which also uses alternating-exposure RGB frames (the closest setup to ours) as a reference. Figure 7 shows the reconstruction results on real-synthetic data of the proposed method and Chen21 [3]. Thanks to the complementary motion information provided by spike trains, the abundant color extracted from alternating-exposure RGB frames, and the accurate textures contained in spike frames, the proposed method is capable of reconstructing rich texture details with less motion blur. For ex-

[6] In this section, we use "Last name of the first author+year" as synonyms of methods for comparison.

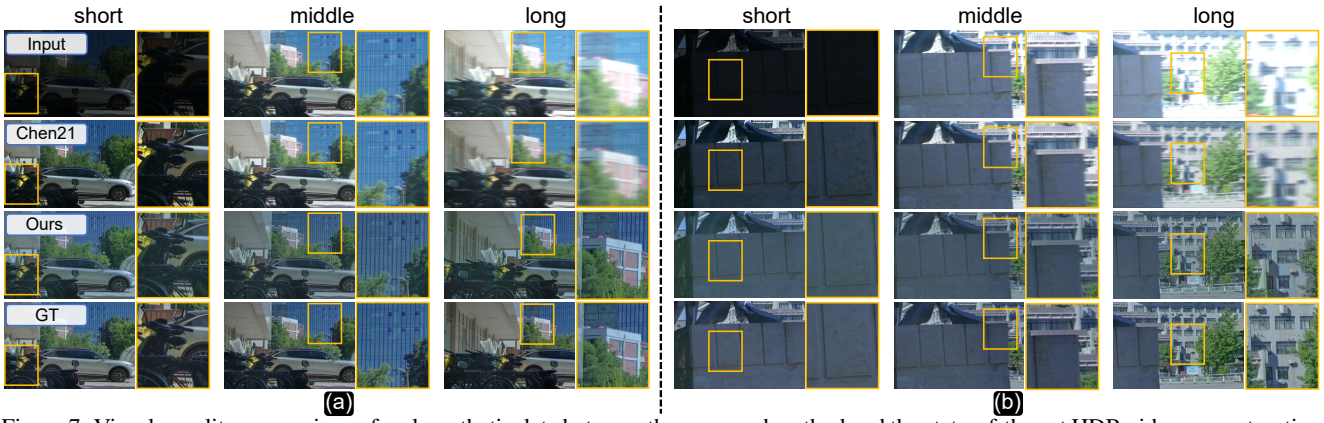

Figure 7. Visual equality comparison of real-synthetic data between the proposed method and the state-of-the-art HDR video reconstruction method: Chen 21 [3]. We present two sets of results in (a) and (b). Please zoom-in electronic versions for better details, and watch the HFR videos on the project page.

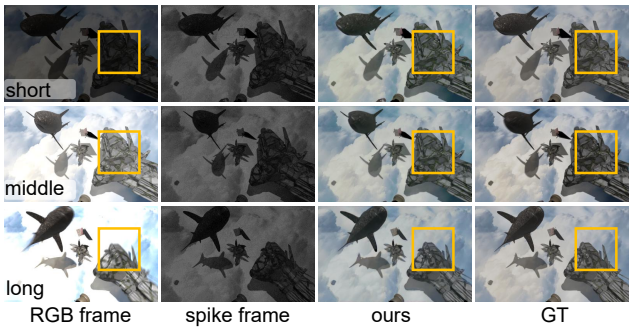

Figure 8. Validation on the synthetic data.

ample, in the long-exposure frame in the first row of (a), the building marked by a yellow box suffers from severe motion blur and overexposure. Chen21 [3] partially recovers the colors of this building, but it fails to remove the blurry artifacts. In the results generated by our method, the edges are sharp and the colors are vivid. In Fig. 7(b), the motions across RGB frames have a very large span, Chen21 [3] can only recover the corresponding LFR videos, while our method can reconstruct an HFR video with smooth motion.

We evaluate the reconstructed HDR in terms of PSNR, SSIM, HDR-VDP-2 [32], and HDR-VQM [36]. Table 2 clearly shows that our framework outperforms the state-of-the-art method [3] in all the metrics on the real-synthetic data in the condition of 60 FPS. And we achieve excellent performance in the condition of 1000 FPS. We designed ablation experiments and used them to demonstrate the effectiveness of the modules in our framework. For "w/o I", we simply stack the spike trains with a time window, and upsample them using bilinear interpolation; for "w/o PS", we replace PixelShuffle with a convolutional layer. The two groups of experiments verify the effectiveness of spike frame preprocessing in Step ①. For "w/o F1" and "w/o F2", we remove the flow-based interpolation in the deblurring module and the merging module. The two groups of experiments verify the effectiveness of SC-Flow [16] based interpolation in Steps ② and ③. To further verify the effectiveness of deblurring module, we completely remove it in "w/o DM". For "w/o DeConv", we replace the deformable convolutional layers with traditional convolution layers. For "w/ t-loss", we remove the warping operation on $\mathbf{C}_{i\ \ 1}$ and add the temporal consistent loss that is estimated by a pre-trained optical flow model [23], which is widely used in video processing [5, 39]. Since the $\mathbf{C}_{i\ \ 1}$ is warped by accurate optical flow $\mathbf{F}_{i\ \ 1}$ and merged into the current step i, our method fundamentally has a strong temporal consistent constraint for video processing. Thus, our merging module does not need this loss during training.

Table 2. Quantitative results and ablation study on our real-synthetic data. We sample 60 FPS videos from our results for the comparison with Chen21 [3]. ↑ (↓) indicates larger (smaller) values are better.

Comparison with the state-of-th-art method					
Method	PSNR↑	SSIM↑	HDR-VDP2↑	HDR-VQM↓	FPS
Chen21 [3]	18.46	0.697	27.34	0.536	60
Ours	<u>30.14</u>	<u>0.921</u>	<u>60.14</u>	<u>0.093</u>	
Chen21 [3]	/	/	/	/	1000
Ours	**24.38**	**0.903**	**47.79**	**0.120**	
Ablation study					
w/o I	23.15	0.886	46.03	0.143	1000
w/o PS	23.98	0.881	46.47	0.141	
w/o F1	19.76	0.723	38.95	0.314	
w/o F2	18.04	0.716	35.89	0.356	
w/ t-loss	22.41	0.864	43.64	0.142	
w/o DeConv	24.31	0.897	47.66	0.127	
w/o DM	19.01	0.714	37.97	0.338	

4.2. Qualitative Evaluation using Real Data

In order to demonstrate the effectiveness of the proposed framework on real-world scenes, we collect 20 sets of real-world data, which are captured by our hybrid camera system shown in Fig. 6. We have compared our slow-motion capability with that of the commercial cameras. As shown in Fig. 9(a), the electric fan is moving at about 40 rounds

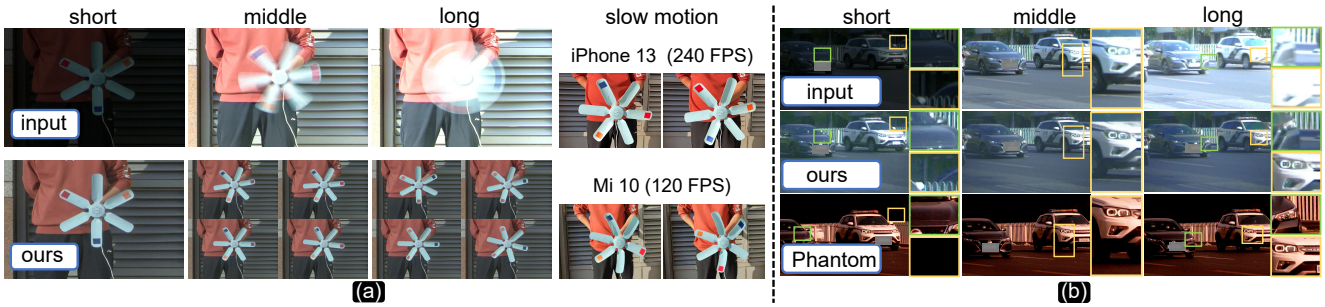

Figure 9. Visual quality comparison of real-world data between the proposed method and commercial cameras with the slow-motion capability. In (a), we show two adjacent frames for the video captured by smartphones that have slow-motion capability. The commercial cameras are not calibrated so their results are not strictly aligned with ours. (b) is the comparison with Phantom camera set to 1000 FPS.

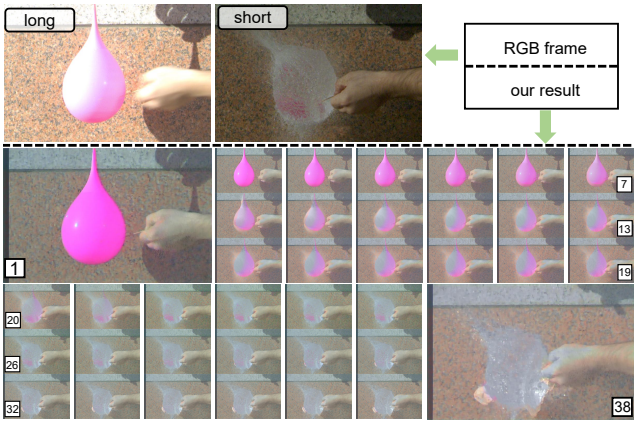

Figure 10. Qualitative visualization of our method in a super fast scene: a balloon bursting. We select 38 frames from our results for showing.

per second. The short-exposure image is severely underexposed with less blurry artifacts, and the middle- and long-exposure images have severe blurring and oversaturated artifacts. With the accurate motion and texture information captured by the spiking camera, we have recovered temporally smooth video sequences. Four recovered images are shown for the middle- and long-exposure images. For the videos captured by iPhone 13 and Mi 10, the motions between frames are not continuous. And the electric fan captured by Mi 10 is deformed due to the rolling shutter. In Fig. 9(b), we compare our method with the Phantom[7] camera set to 1000 FPS. Since the exposure time of the Phantom camera is extremely short, it fails to capture regions where scene radiance is weak.

5. Conclusion

We propose an HFR&HDR video reconstruction method with a hybrid camera that is composed of an alternating-exposure RGB sensor and a spiking sensor. Extensive experiments on synthetic and real-world data demonstrate the superior performance of the proposed method.

[7]Refer to footnote 2. Camera model: VEO 640, F/1.8, 85mm lens.

Discussion. (*i*) For super fast scenes, *e.g.*, a balloon bursting, it is difficult to capture clear motions with a conventional RGB camera at 60 FPS. Therefore, the well-exposed color of the bursting balloon is not captured with the short exposure, which brings challenges to our reconstruction of accurate color. In our results, although the colors are somewhat distorted, we can still recover a smooth video sequence. Once the frame rate of the RGB camera is increased, *e.g.*, 120 FPS, temporally smoother video with more accurate color is expected to be more reliably recovered. (*ii*) Since QIS [1, 29] share the same imaging model with the spiking camera, our method is ready to be applied to it. We show the simulation in supplementary material.

Limitation and future work. Beam splitter is arguable for making a practical system on mobile devices. But when compact design is not a hard constraint, beam splitter has unique advantages in spatial alignment, that is why it is broadly adopted in building a hybrid prototype for HDR [15, 24, 33, 50]. Side-by-side arrangement with parallax unavoidably introduces occlusions and alignment issues, which is a promising direction to explore for our future work. Due to the low spatial resolution (250×400) of the current model we use is, we have to super-resolve the spike frames in feature space. If higher-resolution spike signals can be directly obtained, our method can achieve better visual quality. Besides, there is a domain gap between synthetic spike trains and real-captured spike trains since the noise of the spiking camera is more complex than the simulator. For time complexity, our approach is better suited as a post-processing module. The number of parameters is 45.7M and the time cost per frame is 0.371s with a single NVIDIA GeForce RTX 3090 graphics card. We hope to tackle these issues in the future work and achieve higher frame rate reconstruction.

Acknowledgement

This work was supported by National Key R&D Program of China (2021ZD0109803), National Natural Science Foundation of China under Grant No. 62088102, 62136001. Yakun Chang was also supported by China Postdoctoral Science Foundation (8206300710).

References

[1] Gnanasambandam Abhiram and Chan Stanley H. HDR imaging with quanta image sensors: Theoretical limits and optimal reconstruction. *IEEE Transactions on Computational Imaging*, 6:1571–1585, 2020. 2, 8

[2] Moshe Ben-Ezra and Shree K Nayar. Motion deblurring using hybrid imaging. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003. 2

[3] Guanying Chen, Chaofeng Chen, Shi Guo, Zhetong Liang, Kwan-Yee K Wong, and Lei Zhang. HDR video reconstruction: A coarse-to-fine network and a real-world benchmark dataset. In *Proc. of International Conference on Computer Vision*, pages 2502–2511, 2021. 2, 4, 6, 7

[4] Wooyeong Cho, Sanghyeok Son, and Dae-Shik Kim. Weighted multi-kernel prediction network for burst image super-resolution. In *Proc. of Computer Vision and Pattern Recognition*, pages 404–413, 2021. 4

[5] Jonghyun Choi, Kuk-Jin Yoon, et al. Learning to super resolve intensity images from events. In *Proc. of Computer Vision and Pattern Recognition*, pages 2768–2776, 2020. 7

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proc. of International Conference on Computer Vision*, pages 764–773, 2017. 5

[7] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. of ACM SIGGRAPH*, pages 1–10. 2008. 2

[8] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Burst image restoration and enhancement. In *Proc. of Computer Vision and Pattern Recognition*, pages 5759–5768, 2022. 2

[9] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. HDR image reconstruction from a single exposure using deep cnns. *ACM Transactions on Graphics*, 36(6):1–15, 2017. 2

[10] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2020. 1

[11] Clément Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *Proc. of European Conference on Computer Vision*, pages 538–554, 2018. 2

[12] Yulia Gryaditskaya, Tania Pouli, Erik Reinhard, Karol Myszkowski, and Hans-Peter Seidel. Motion aware exposure bracketing for HDR video. In *Computer Graphics Forum*, volume 34, pages 119–130. Wiley Online Library, 2015. 2

[13] Saghi Hajisharif, Joel Kronander, and Jonas Unger. Adaptive dualiso HDR reconstruction. *EURASIP Journal on Image and Video Processing*, 2015(1):1–13, 2015. 2

[14] Jin Han, Yixin Yang, Chu Zhou, Chao Xu, and Boxin Shi. Evintsr-net: Event guided multiple latent frames reconstruction and super-resolution. In *Proc. of International Conference on Computer Vision*, pages 4882–4891, 2021. 2, 4

[15] Jin Han, Chu Zhou, Peiqi Duan, Yehui Tang, Chang Xu, Chao Xu, Tiejun Huang, and Boxin Shi. Neuromorphic camera guided high dynamic range imaging. In *Proc. of Computer Vision and Pattern Recognition*, pages 1730–1739, 2020. 2, 8

[16] Liwen Hu, Rui Zhao, Ziluo Ding, Lei Ma, Boxin Shi, Ruiqin Xiong, and Tiejun Huang. Optical flow estimation for spiking camera. In *Proc. of Computer Vision and Pattern Recognition*, pages 17844–17853, 2022. 2, 3, 4, 5, 6, 7

[17] Tiejun Huang, Yajing Zheng, Zhaofei Yu, Rui Chen, Yuan Li, Ruiqin Xiong, Lei Ma, Junwei Zhao, Siwei Dong, Lin Zhu, et al. 1000× faster camera and machine vision with ordinary devices. *Engineering*, 2022. 1, 3, 4, 6

[18] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proc. of Computer Vision and Pattern Recognition*, pages 3320–3329, 2020. 2, 3, 4

[19] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep HDR video from sequences with alternating exposures. In *Computer graphics forum*, volume 38, pages 193–205. Wiley Online Library, 2019. 2, 4, 6

[20] Nima Khademi Kalantari, Ravi Ramamoorthi, et al. Deep high dynamic range imaging of dynamic scenes. *ACM Transactions on Graphics*, 36(4):144–1, 2017. 2, 6

[21] Nima Khademi Kalantari, Eli Shechtman, Connelly Barnes, Soheil Darabi, Dan B Goldman, and Pradeep Sen. Patch-based high dynamic range video. *ACM Transactions on Graphics*, 32(6):202–1, 2013. 2, 4, 6

[22] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. *ACM Transactions on Graphics*, 22(3):319–325, 2003. 2

[23] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proc. of European Conference on Computer Vision*, pages 170–185, 2018. 7

[24] Byungju Lee and Byung Cheol Song. Multi-image high dynamic range algorithm using a hybrid camera. *Signal Processing: Image Communication*, 30:37–56, 2015. 2, 8

[25] Juan Antonio Leñero-Bardallo, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. A 3.6 μs latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011. 1

[26] Feng Li, Jingyi Yu, and Jinxiang Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *Proc. of Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 3

[27] Ziwei Liu, Lu Yuan, Xiaoou Tang, Matt Uyttendaele, and Jian Sun. Fast burst images denoising. *ACM Transactions on Graphics*, 33(6):1–9, 2014. 4

[28] Kede Ma, Hui Li, Hongwei Yong, Zhou Wang, Deyu Meng, and Lei Zhang. Robust multi-exposure image fusion: A structural patch decomposition approach. *IEEE Transactions on Image Processing*, 26(5):2519–2532, 2017. 2

[29] Ulku Arin C Bruschini Claudio Charbon Edoardo Ma Sizhuo, Gupta Shantanu and Gupta Mohit. Quanta burst photography. *ACM Transactions on Graphics*, 39(4):79–1, 2020. 8

[30] Stephen Mangiat and Jerry Gibson. High dynamic range video with ghost removal. In *Applications of Digital Image Processing XXXIII*, volume 7798, pages 307–314. SPIE, 2010. 2

[31] Stephen Mangiat and Jerry Gibson. Spatially adaptive filtering for registration artifact removal in HDR video. In *Proc. of International Conference on Image Processing*, pages 1317–1320. IEEE, 2011. 2

[32] Rafał Mantiuk, Kil Joong Kim, Allan G Rempel, and Wolfgang Heidrich. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics*, 30(4):1–14, 2011. 7

[33] Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, Billy Chen, John F Hughes, and Shree K Nayar. Optical splitting trees for high-precision monocular imaging. *IEEE Computer Graphics and Applications*, 27(2):32–42, 2007. 2, 8

[34] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *Pacific Conference on Computer Graphics and Applications*, pages 382–390, 2007. 2

[35] Nico Messikommer, Stamatios Georgoulis, Daniel Gehrig, Stepan Tulyakov, Julius Erbach, Alfredo Bochicchio, Yuanyou Li, and Davide Scaramuzza. Multi-Bracket high dynamic range imaging with event cameras. In *Proc. of Computer Vision and Pattern Recognition*, pages 547–557, 2022. 2

[36] Manish Narwaria, Matthieu Perreira Da Silva, and Patrick Le Callet. HDR-VQM: An objective quality measure for high dynamic range video. *Signal Processing: Image Communication*, 35:46–60, 2015. 7

[37] Shree K Nayar and Tomoo Mitsunaga. High dynamic range imaging: Spatially varying pixel exposures. In *Proc. of Computer Vision and Pattern Recognition*, volume 1, pages 472–479. IEEE, 2000. 2

[38] Avinash Paliwal and Nima Khademi Kalantari. Deep slow motion video reconstruction with hybrid imaging system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1557–1569, 2020. 3

[39] 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, 2019. 7

[40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. of Computer Vision and Pattern Recognition*, pages 779–788, 2016. 1

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Proc. of Advances in Neural Information Processing Systems*, 28, 2015. 1

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[43] Yash Sanghvi, Abhiram Gnanasambandam, and Stanley H Chan. Photon limited non-blind deblurring using algorithm unrolling. *IEEE Transactions on Computational Imaging*, 2022. 5

[44] Marcel Santana Santos, Tsang Ing Ren, and Nima Khademi Kalantari. Single image HDR reconstruction using a cnn with masked features and perceptual loss. *arXiv preprint arXiv:2005.07335*, 2020. 2

[45] Pradeep Sen, Nima Khademi Kalantari, Maziar Yaesoubi, Soheil Darabi, Dan B Goldman, and Eli Shechtman. Robust patch-based HDR reconstruction of dynamic scenes. *ACM Transactions on Graphics*, 31(6):203–1, 2012. 2

[46] Richard Shaw, Sibi Catley-Chandar, Ales Leonardis, and Eduardo Perez-Pellitero. HDR reconstruction from bracketed exposures and events. *arXiv preprint arXiv:2203.14825*, 2022. 2

[47] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proc. of Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. 4, 5

[48] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Proc. of Advances in Neural Information Processing Systems*, 28, 2015. 5

[49] Yu-Wing Tai, Hao Du, Michael S Brown, and Stephen Lin. Image/video deblurring using a hybrid camera. In *Proc. of Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[50] Michael D Tocci, Chris Kiser, Nora Tocci, and Pradeep Sen. A versatile HDR video production system. *ACM Transactions on Graphics*, 30(4):1–10, 2011. 2, 8

[51] 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 *Proc. of Computer Vision and Pattern Recognition*, pages 17755–17764, 2022. 2, 3

[52] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time Lens: Event-based video frame interpolation. In *Proc. of Computer Vision and Pattern Recognition*, pages 16155–16164, 2021. 2, 3

[53] 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. 6

[54] Fang Xu, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu. Motion deblurring with real events. In *Proc. of International Conference on Computer Vision*, pages 2583–2592, 2021. 3, 4

[55] Qingsen Yan, Lei Zhang, Yu Liu, Yu Zhu, Jinqiu Sun, Qinfeng Shi, and Yanning Zhang. Deep HDR imaging via a non-local network. *IEEE Transactions on Image Processing*, 29:4308–4322, 2020. 2

[56] 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 *Proc. of*

International Conference on Computer Vision, pages 14589–14598, 2021. 3

[57] Cheng Zhang, Shaolin Su, Yu Zhu, Qingsen Yan, Jinqiu Sun, and Yanning Zhang. Exploring and evaluating image restoration potential in dynamic scenes. In *Proc. of Computer Vision and Pattern Recognition*, pages 2067–2076, 2022. 1

[58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

[59] Xiang Zhang and Lei Yu. Unifying motion deblurring and frame interpolation with events. In *Proc. of Computer Vision and Pattern Recognition*, pages 17765–17774, 2022. 2

[60] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proc. of Computer Vision and Pattern Recognition*, pages 2472–2481, 2018. 4

[61] Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. Spk2Imgnet: Learning to reconstruct dynamic scene from continuous spike stream. In *Proc. of Computer Vision and Pattern Recognition*, pages 11996–12005, 2021. 1, 4, 6

[62] Yajing Zheng, Lingxiao Zheng, Zhaofei Yu, Boxin Shi, Yonghong Tian, and Tiejun Huang. High-speed image reconstruction through short-term plasticity for spiking cameras. In *Proc. of Computer Vision and Pattern Recognition*, pages 6358–6367, 2021. 1, 4

[63] Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. A retina-inspired sampling method for visual texture reconstruction. In *Proc. of International Conference on Multimedia and Expo*. 1

[64] Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Retina-like visual image reconstruction via spiking neural model. In *Proc. of Computer Vision and Pattern Recognition*, pages 1438–1446, 2020. 1, 4

[65] Yunhao Zou, Yinqiang Zheng, Tsuyoshi Takatani, and Ying Fu. Learning to reconstruct high speed and high dynamic range videos from events. In *Proc. of Computer Vision and Pattern Recognition*, pages 2024–2033, 2021. 1

Supplementary Material
1000 FPS HDR Video with a Spike-RGB Hybrid Camera

Yakun Chang[1,2] Chu Zhou[3] Yuchen Hong[1,2] Liwen Hu[2] Chao Xu[3] Tiejun Huang[1,2] Boxin Shi[1,2*]
[1] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2] National Engineering Research Center of Visual Technology, School of Computer Science, Peking University
[3] National Key Laboratory of General AI, School of Intelligence Science and Technology, Peking University
{yakunchang, zhou_chu, huliwen, tjhuang, shiboxin}@pku.edu.cn
yuchenhong.cn@gmail.com, xuchao@cis.pku.edu

In the supplementary material, we provide details of optical flow estimation (Sec. 3.2),our real-synthetic data (Sec. 3.5), and show additional comparisons with the state-of-the-art method [2] (Sec. 4). We further provide a supplementary video to show the motivation of our methodology and results for both synthetic and real-world scenes.

6. Details of Optical Flow Estimation

The source code of SC-Flow [3] outputs optical flow vectors every 1 ms. However, $\mathbf{F}_{j\to i}$ in Eqn. 4 requires flow vectors estimated from longer time interval. To obtain more accurate optical flow vectors, we first initialize the flow vectors by accumulating the optical flows from j to i, then we conduct refinement by secondly feeding the initial flows to SC-Flow [3].

7. Details of Real-Synthetic Data

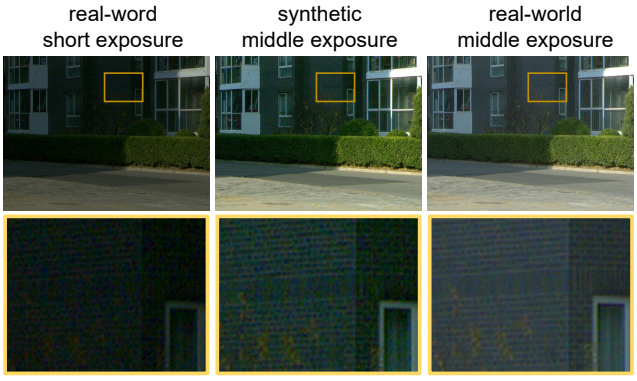

Figure 11. Visualization for noise levels of a short-exposure image (1 ms), a synthetic middle-exposure image by merging 4 short-exposure images, and a real-world middle-exposure image.

*Corresponding author.

Since the dataset provided by Chen *et al*. [2] only contains low frame rate (LFR) alternating-exposure RGB sequence, we cannot synthesize high-speed spike trains from such an LFR dataset. To collect alternating-exposure images and spike trains in high-speed conditions, it may be feasible to synthesize them from HFR videos captured with a short exposure. Theoretically, a middle-exposure image can be synthesized by merging several short-exposure images if there is no camera noise. However, as shown in Fig. 11, we find that the short-exposure image (left) captured with 1 ms exposure contains strong noise, which cannot be effectively suppressed by merging a burst of short-exposure images. In the middle column of Fig. 11, the synthetic middle-exposure image contains strong noise as well, whereas the real-world middle-exposure image (right) captured with 4 ms contains less noise. The reason is that images captured with short exposures are more severely contaminated by camera noise, and the camera noise (the mean is not zero) is also accumulated when we merge short-exposure images. Since it is infeasible to synthesize longer-exposed images by merging a sequence of short-exposure images, we design a method to synthesize blurry longer-exposed images.

In this work, we collect the real-synthetic dataset from alternating-exposure RGB sequences captured in slow-motion conditions. Our pipeline for the synthesis of middle-exposure images is shown in Fig. 12. Firstly, we set the alternating exposures to 1 ms, 4 ms, and 12 ms, which are consistent with our real-world data. Then we capture RGB sequences in slow-motion conditions with a frame rate of 80 FPS (the largest frame rate in this exposure setting). To synthesize the ground truths, we treat each 3 adjacent alternating-exposure frames as a group and synthesize a well-exposed image (ground truth) using exposure fusion [5]. Since the total time T of each frame in the original RGB sequence is 12.5 ms, and we fuse 3 adjacent

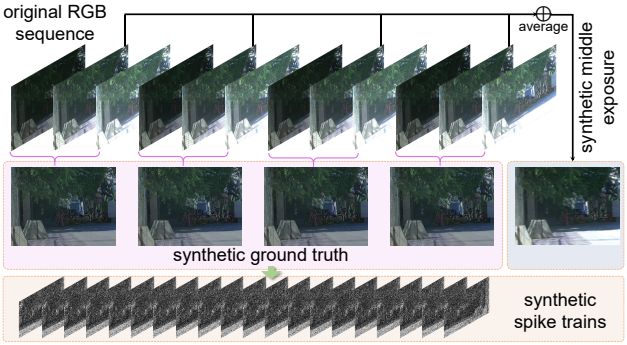

Figure 12. We capture alternating-exposure RGB sequence in slow-motion conditions, and compress the time with a ratio of 37.5 $ms \rightarrow 1\ ms$. We synthesize a blurry middle-exposure image by averaging 4 neighbored middle-exposure images.

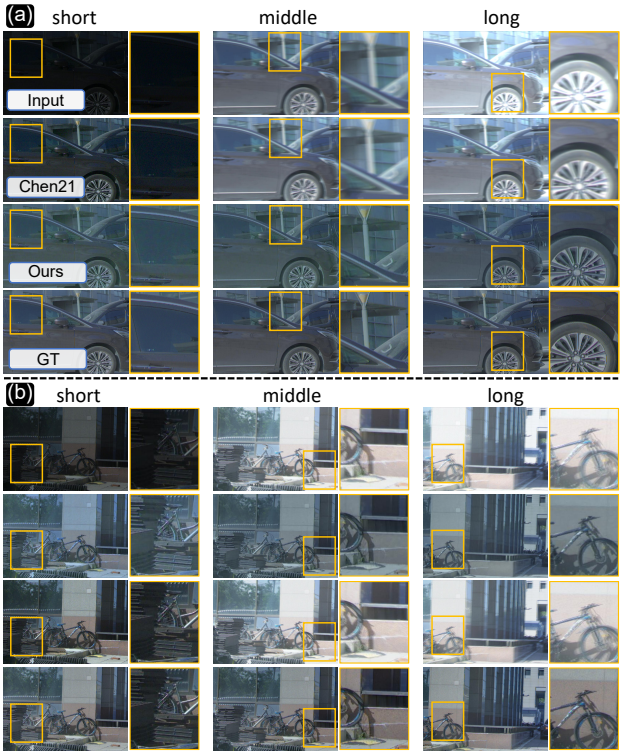

Figure 13. This is the following of Fig. 7 in the main paper. Visual quality comparison of real-synthetic data between the proposed method and the state-of-the-art HDR video reconstruction method: Chen 21 [2].

frames as a ground truth image (1 ms), the compression ratio of time is 37.5 $ms \rightarrow 1\ ms$. We select the synthetic data according to the temporal relationship of real-world data. For example, since the start time of the first three RGB frames in real-world data are at the first, the 17th, and the 34th ms, the first three synthetic alternating-exposure RGB frames are generated from group 1, group 17 to 20, and group 34 to 45, respectively. A short-exposure image is

Figure 14. This is the following of Fig. 10 in the main paper. Please view this figure together with Fig. 10. In this figure, we compare our results on the balloon bursting with the slow-motion capability of iPhone 13 and Mi 10. We show 4 adjacent frames captured by the smartphones.

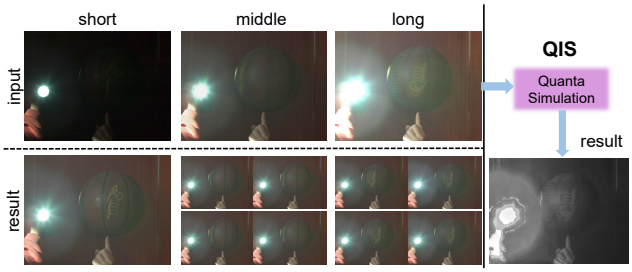

Figure 15. Testing in a scene with strong light. And the simulation of an HDR method that designed for quanta image sensors [1].

obtained by directly selecting the short-exposure image in group 1. To synthesize a blurry middle-exposure image, we average four middle-exposure images selected from group 17 to 20. Similarly, a blurry long-exposure image is synthesized by averaging the 12 long-exposure images of group 34 to 45. Finally, we synthesize spike trains by converting ground truth RGB frames to intensity maps and generate spike trains with the integrate-and-fire methodology [6]. Thanks to this method, the domain gap between the synthetic alternating-exposure images and real-world ones captured in high-speed condition is small.

8. Additional qualitative results

In this section, we present additional two sets of visual comparisons on real-synthetic data. As shown in Fig. 13, our method recovers well-exposed color frames with less motion blur. Figure 14 is the following of Fig. 10 in the main paper, which presents the visual comparison of the balloon bursting with the slow-motion capability of two commercial cameras. We can see that the two cameras also fail to capture continuous motions of the balloon bursting. In Fig. 15, we present a set of results (3 RGB images and 9 output images) to validate HFR&HDR performance in conditions with strong light source. We can see a basketball is

spinning rapidly at the fingertip beside a hand-held strong light. Our method successfully captures the texture details of the basketball without motion blur. Since quanta image sensors (QIS) [4], *e.g.*, the SPAD camera and Gigajot QIS series share similar imaging model with the spiking camera, we conduct comparison with QIS. And for the reason that we do not have a QIS camera on hand, the comparison is performed through a simulation with the source code provided by Abhiram and Chan [1].

References

[1] Gnanasambandam Abhiram and Chan Stanley H. HDR imaging with quanta image sensors: Theoretical limits and optimal reconstruction. *IEEE Transactions on Computational Imaging*, 6:1571–1585, 2020. 2, 3

[2] Guanying Chen, Chaofeng Chen, Shi Guo, Zhetong Liang, Kwan-Yee K Wong, and Lei Zhang. HDR video reconstruction: A coarse-to-fine network and a real-world benchmark dataset. In *Proc. of International Conference on Computer Vision*, pages 2502–2511, 2021. 1, 2

[3] Liwen Hu, Rui Zhao, Ziluo Ding, Lei Ma, Boxin Shi, Ruiqin Xiong, and Tiejun Huang. Optical flow estimation for spiking camera. In *Proc. of Computer Vision and Pattern Recognition*, pages 17844–17853, 2022. 1

[4] Ulku Arin C Bruschini Claudio Charbon Edoardo Ma Sizhuo, Gupta Shantanu and Gupta Mohit. Quanta burst photography. *ACM Transactions on Graphics*, 39(4):79–1, 2020. 3

[5] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *Pacific Conference on Computer Graphics and Applications*, pages 382–390, 2007. 1

[6] Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. Spk2ImgNet: Learning to reconstruct dynamic scene from continuous spike stream. In *Proc. of Computer Vision and Pattern Recognition*, pages 11996–12005, 2021. 2