# Real-data-driven 2000 FPS Color Video from Mosaicked Chromatic Spikes (Supplemental Material)

Siqi Yang[1,#], Zhaojun Huang[2,3,#], Yakun Chang[5,6], Bin Fan[4], Zhaofei Yu[1], and Boxin Shi[2,3,1,*]

[1] Institute for Artificial Intelligence, Peking University
[2] State Key Lab of Multimedia Info. Processing, School of Computer Science, Peking University
[3] Nat'l Eng. Research Ctr. of Visual Technology, School of Computer Science, Peking University
[4] Nat'l Key Lab of General AI, School of Intelligence Science and Technology, Peking University
[5] Institute of Information Science, Beijing Jiaotong University
[6] Visual Intelligence +X International Cooperation Joint Laboratory of the Ministry of Education
{yousiki, huangzhaojun, binfan, yuzf12, shiboxin}@pku.edu.cn, ykchang@bjtu.edu.cn

In the supplementary material, we provide chromatic spike camera details, method implementation details, analysis of the hyperparameters ($K$, $L$, and $W$ in Sec. 4), and additional results. We further provide a supplementary video to show the high-speed color videos reconstructed from mosaicked chromatic spike streams.

## 7 Mosaicked chromatic spike camera

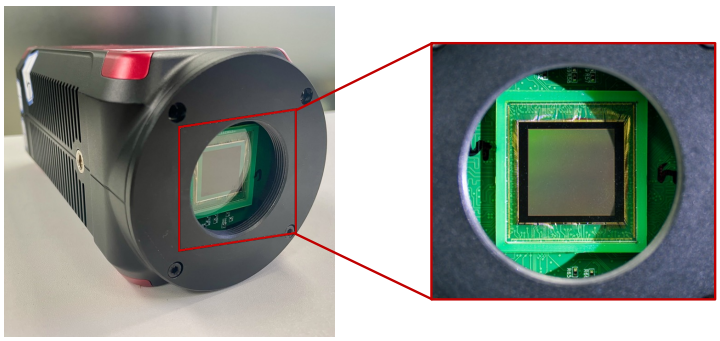

**Fig. 8:** Mosaicked chromatic spike camera.

We describe more specifics of the mosaicked chromatic spike camera. As we mentioned in Sec. 3, a color filter array (CFA) is applied to the sensor to capture mosaicked chromatic spike streams, adhering to the widely used Bayer pattern (RGGB). The chromatic spike frames are transmitted to the main computer via optical fiber, and are then stored to solid-state drives.

# Equal contributions. * Corresponding author.

## 8 Implementation details

In this section, we provide implementation details about our method.

**Chromatic spikes denoising.** During the training stage of our chromatic spike denoiser, we employ a sampling strategy where a subset of the masked volume is randomly selected for each iteration. The loss function remains nearly the same, except for the replacement of mask-aware averaging with summation. In terms of experimental setup, we configured the parameters as follows. We set $h_s = w_s = 4$ for global-aware masking, $\eta = 1$ and $\lambda_i = 0.05$ for loss weighting, and $\lambda$ gradually increasing from 2 to 20 aligned with the training progress. Furthermore, to accommodate a wide range of signal-to-noise ratios, the accumulation temporal window size $W$ is randomly drawn from the range 5 to 200, enhancing the tolerance of the denoising module to varied noise levels. To address the scarcity of real-world chromatic spike streams, we augment our training data with common means, *e.g.*, randomly flipping and cropping accumulated frames. As previously described in Sec. 4.1, our approach is based on the zero-mean assumption of the noise distribution of chromatic spikes. For the chromatic spike denosing module, we modify the U-Net [4] to restore clean frames from the noise contaminated spike frames. The architecture of our network comprises 5 blocks to extract multi-scale features, and there are 5 blocks in the decoder, which reversely map the multi-scale features to an output video frame. To preserve the texture information in low-level features, we add skip connections between the encoder and decoder. Each block in the encoder and the decoder consist of 2 convolutional layers, and the output of each convolutional layer is activated by LeakyReLU [5]. Thus, the denoising module consists of a total of 25 convolutional layers, including head and tail processing layers.

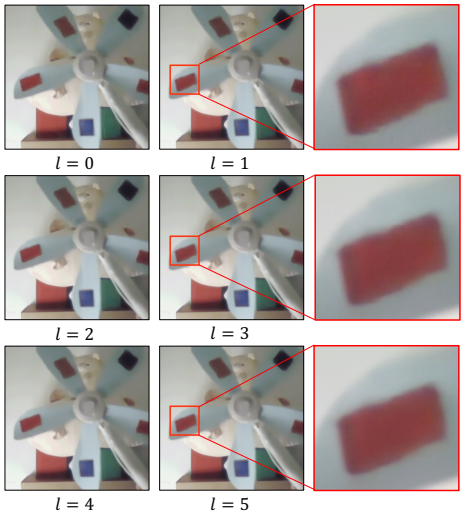


**Fig. 9:** Additional results of progressive warping. We visualize the intermediate frames corresponding to increasing $l$ (from left to right, from up to down) and enlarge some regions in red bounding boxes for detailed observation.

**Progressive warping.** In the progressive warping module, we capitalize on the existing method's capacity to align multiple adjacent frames [2]. This approach offers enhanced robustness against potential noise in frames accumulated over short durations from chromatic spike streams, in comparison to other optical flow estimation techniques. The initially accumulated frames, recovered from a

small window size (*e.g.*, $W = 10$), may exhibit some degree of noise even after denoising, which is detrimental to the alignment of small patches. Consequently, we employ larger patch sizes for these initial frames, progressively decreasing the patch size as the frames' reliability improves. In our experiments, setting $L = 3$, $W = 10$, and $K = 3$ is proved sufficient for most of the scenes. To achieve robust progressive warping, we estimate the optical flow from multi-scale maps. At the first step, we obtain multi-scale maps by downsampling the video frames with the scales of $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$. Thus, including the original resolution frame, each group of multi-scale maps contains 4 frames. The process entails progressively searching for the most suitable match from the pyramid maps to references, starting from the lowest resolution and ascending to the highest. The culmination of this process is the identification of the optimal optical flow required to align $K$ frames with the reference frame, specifically the $(K + 1)/2$-th frame. This procedure is elaborated in Sec. 4.2 and represents a singular warping process. The entire progressive warping pipeline is composed of $L$ such warping steps. It is noteworthy that the initial step in this sequence operates on spike planes ($S_c$), as opposed to intermediate frames ($I_{c,l}$).

**Spike simulator.** Our chromatic spike simulator mainly follows the noise modeling design of existing works (*e.g.*, [6,7]), including dark-current estimation and perturbed stimulation threshold, and also introduces the simulation of Bayer-pattern CFA.

**Inference time.** Our chromatic spike denoising module requires approximately 10 hours for training. The proposed method currently functions as an *off-line solver* for 2000 FPS video reconstruction, with an inference speed of 16.7 FPS, in the condition of $L = 3, W = 10$. It is worth noting that the primary time consumption is optical flow estimation ($\sim 65\%$), which is independent from our main pipeline and can be independently optimized. We compare the inference speed of our method with other methods in Tab. 2. All metrics are benchmarked with an RTX3090 GPU, except SJDD [1], which utilizes an A6000 GPU due to its higher memory requirements.

**Table 2:** Comparison of inference speed.

| **Method** | Ours | TFP | TFI | TFSTP | MS23 | SJDD |
|---|---|---|---|---|---|---|
| **FPS** | 16.7 | 1k | 22 | 13.9 | 2.5 | 0.32 |

## 9 Analysis of hyperparameters

As discussed in Sec. 8, we empirically found that the set of hyperparameters $L = 3, W = 10, K = 3$ is sufficient for our testing scenes. The three hyperparameters jointly determine the pseudo-long exposure, that is, the exposure time is

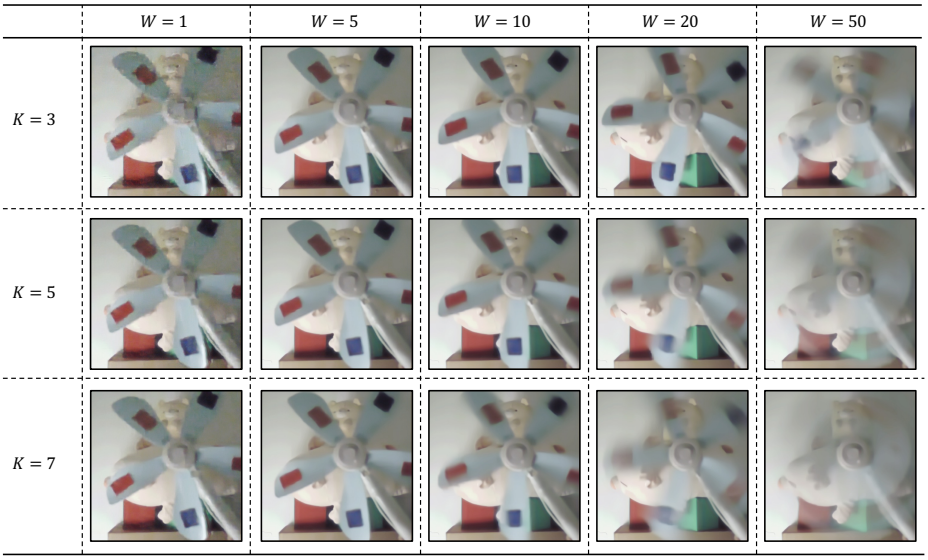


**Fig. 10:** Analysis of hyperparameters ($K$ and $W$). We illustrate the reconstruction results from different combinations of $K$ and $W$.

Fig. 7 (c) Fig. 7 (d) Fig. 12 (e) Fig. 12 (f) Fig. 12 (g) Fig. 12 (h)

**Fig. 11:** The ground truth of synthetic scenes.

equivalent to $(K-1) \times L \times W + 1$. We further conduct an ablation study to analyze the impact of these hyperparameters on the reconstruction quality, as shown in Fig. 9 and Fig. 10. We adjust the hyperparameters and evaluate the reconstruction results qualitatively. With $W$ and $K$ increasing, our proposed method obtains better performance at static regions, while leading to more potential blur at motion regions. Specifically, small $W$ (*e.g.*, $W \leq 5$) makes optical flow estimation almost unpractical, given that the detected photons are extremely limited. Large $W$ (*e.g.*, $W > 20$) introduces motion blur before flow estimation and warping, leading to irretrievable blurry artifacts. We can empirically conclude that $W$ between 10 and 20 fits most of the cases, both static scenes and dynamic objects. While the increase of $W$ doesn't change the computation time very much (because the dimensions of $I$ remain unchanged), $K$ linearly affects the computation costs. While greater $K$ accumulates more spike planes and suppresses noise better, we observe that $K$ between 3 and 5 is sufficient for most of the cases. As shown in Fig. 5 and Fig. 9, with the increase of $L$, our

proposed method can refine the reconstruction results with adjacent frames. In our experiment, we find that when $L = 3$, the quality of the reconstructed color image can converge to a stable value. Note that the hyperparameters are not totally fixed and can be customized based on user preference.

## 10 Additional results

Furthermore, additional results on both real and synthetic data are illustrated in Fig. 12, ground truth images for all synthetic scenes are shown in Fig. 11, and a comparison video is also uploaded with the supplementary material. Please refer to the video for a more comprehensive comparison.

## 11 Compared to supervised learning on synthetic data

One significant challenge in current spike camera research lies in the substantial disparity between synthetic data and real data, which makes models trained on synthetic data perform poorly in real-world applications. Consequently, we utilize real spike streams for self-supervised training to avoid the domain gap issue in this paper. To substantiate this, we employ an identical neural network to conduct supervised training on the GoPro dataset [3], leveraging the video-to-spike simulators to generate synthetic spike streams. Subsequently, we compare the performance of the supervised-learning (SL) model trained on synthetic data with that of our self-supervised learning (SSL) model trained on real data, as shown in Fig. 13. Our proposed method performs significantly better in evaluation.

## 12 Compared to concurrent work

As SJDD [1] was published after our submission, it should be treated as concurrent work, and we did not include a comparison in our main figures. We compare our proposed method with SJDD as shown in Fig. 14, which demonstrates the superiority of our method in terms of noise supervision.

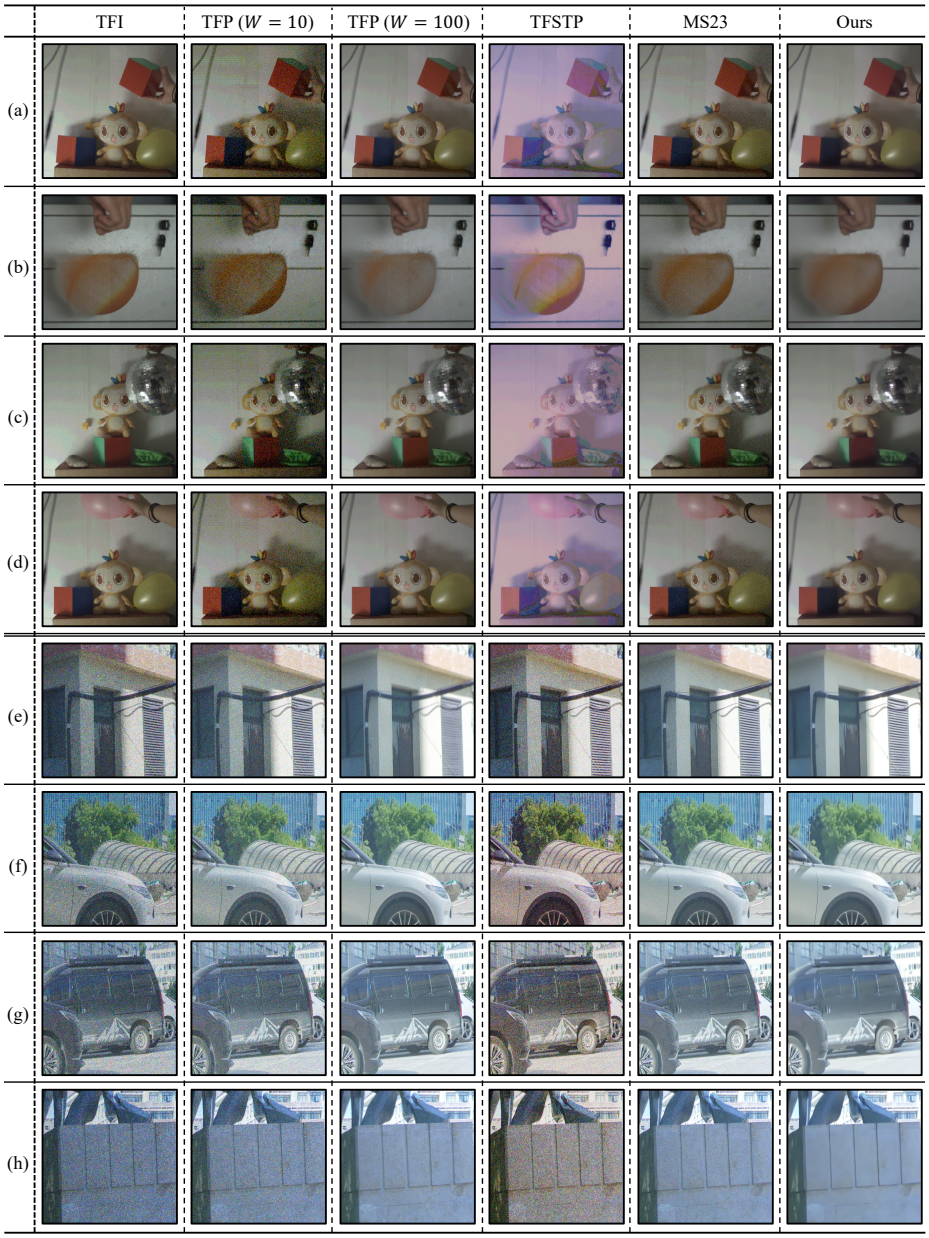


**Fig. 12:** Additional reconstruction results for visual equality comparison of real (a-d) and synthetic (e-h) data between the proposed method and compared methods.

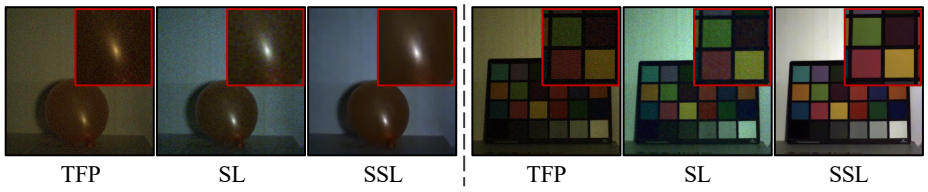


**Fig. 13:** Comparing the performance of supervised-learning denoiser (SL) and our self-supervised learning denoiser (SSL). The noise in short-term temporal window accumulation (TFP) is better removed by our real-data-driven self-supervised learning denoising module.


**Fig. 14:** Comparing our method with SJDD [1].

## References

1. Dong, Y., Xiong, R., Zhao, J., Zhang, J., Fan, X., Zhu, S., Huang, T.: Joint demosaicing and denoising for spike camera. AAAI (2024)
2. Hasinoff, S.W., Sharlet, D., Geiss, R., Adams, A., Barron, J.T., Kainz, F., Chen, J., Levoy, M.: Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM TOG **35**(6), 1–12 (2016)
3. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR. pp. 3883–3891 (2017)
4. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. pp. 234–241 (2015)
5. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
6. Zheng, Y., Zhang, J., Zhao, R., Ding, J., Chen, S., Xiong, R., Yu, Z., Huang, T.: SpikeCV: Open a continuous computer vision era. arXiv preprint arXiv:2303.11684 (2023)
7. Zheng, Y., Zheng, L., Yu, Z., Huang, T., Wang, S.: Capture the moment: High-speed imaging with spiking cameras through short-term plasticity. IEEE TPAMI **45**(7), 8127–8142 (2023)