SVFI: Spiking-Based Video Frame Interpolation for High-Speed Motion

Lujie Xia[1,2], **Jing Zhao**[1,2,3], **Ruiqin Xiong**[1,2*], **Tiejun Huang**[1,2,4]

[1]National Engineering Research Center of Visual Technology (NERCVT), Peking University
[2]Institute of Digital Media, School of Computer Science, Peking University
[3]National Computer Network Emergency Response Technical Team/Coordination Center of China
[4] Beijing Academy of Artificial Intelligence
{lujie.xia, jzhaopku, rqxiong, tjhuang}@pku.edu.cn

Abstract

Occlusion and motion blur make it challenging to interpolate video frame, since estimating complex motions between two frames is hard and unreliable, especially in highly dynamic scenes. This paper aims to address these issues by exploiting spike stream as auxiliary visual information between frames to synthesize target frames. Instead of estimating motions by optical flow from RGB frames, we present a new dual-modal pipeline adopting both RGB frames and the corresponding spike stream as inputs (SVFI). It extracts the scene structure and objects' outline feature maps of the target frames from spike stream. Those feature maps are fused with the color and texture feature maps extracted from RGB frames to synthesize target frames. Benefited by the spike stream that contains consecutive information between two frames, SVFI can directly extract the information in occlusion and motion blur areas of target frames from spike stream, thus it is more robust than previous optical flow-based methods. Experiments show SVFI outperforms the SOTA methods on wide variety of datasets. For instance, in 7 and 15 frame skip evaluations, it shows up to 5.58 dB and 6.56 dB improvements in terms of PSNR over the corresponding second best methods BMBC and DAIN. SVFI also shows visually impressive performance in real-world scenes.

Introduction

Video frame interpolation (VFI) predicts intermediate frames to transform low frame rate video to high frame rate video. Instead of using specialized and expensive equipment, VFI can generate high frame rate video from the content shot by conventional cameras in low-cost way to capture the moments take place in the blink of an eye. Hence, it has attracted increasing attention and been applied in many applications from super slow motion (Jiang et al. 2018) to video coding/decoding (Pourreza and Cohen 2021).

Most existing VFI methods can be summarized into two categories according to their followed pipelines, *e.g.* i) flow-based methods (Niklaus and Liu 2018; Jiang et al. 2018; Bao et al. 2019; Niklaus and Liu 2020; Park et al. 2020; Gui et al. 2020), which combine flow estimation (Ilg et al. 2017; Sun et al. 2018; Liu et al. 2017) with frames warping (Jaderberg et al. 2015) to predict a new frame, and ii) kernel-based methods (Niklaus, Mai, and Liu 2017a,b) that model VFI as local convolution and estimate the convolution kernel over the input frames. However, kernel-based methods only perform well in small motion scenes because they can only estimate small kernels considering the running speed. And flow-based methods rely on optical flow estimation, they not only are limited by the assumptions of illumination and appearance consistency, but also can hardly handle scenes containing complex motions. If complex motions, *e.g.* nonlinear and occlusion occur, the artifacts will appear in the corresponding areas of target frames.

Considering aforementioned limitations, researchers proposed to combine event camera with conventional camera(Tulyakov et al. 2021). This pipeline further promotes the development of VFI field. Event camera is a kind of neuro-

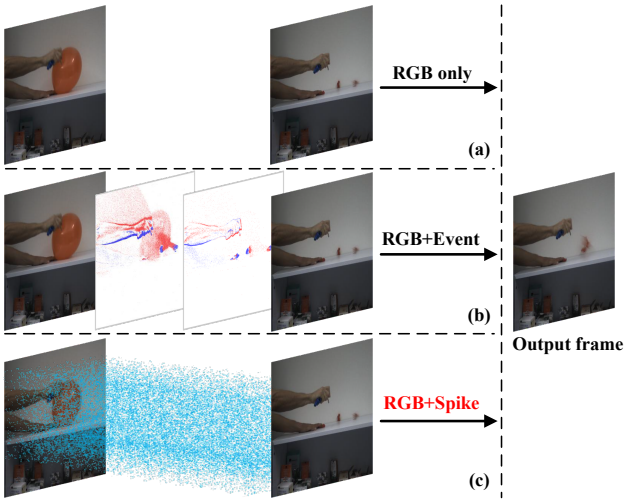

Figure 1: Illustration of VFI methods with three different modal data as inputs. (a) RGB only methods use two consecutive RGB frames as inputs to predict the target frame. (b) RGB+Event methods additionally use the bidirectional event stream between RGB frames. (c) The proposed SVFI first utilizes RGB frames and the spike streams between them. It has potential to enjoy better robustness to synthesize objects with complex motions and occlusion in (a) and objects with few textures and small motions in (b).

*Corresponding Author.

morphic camera, it provides high temporal resolution auxiliary information between two RGB frames. And the motion field extracted from the output of event camera is relatively sparse, for event camera records the edges of objects with high contrast, rich textures and significant motions, no information is generated for other areas based on the working mechanism. Therefore, the event camera has great application value and advantages in tasks such as action recognition, object detection and tracking. However, in low level vision task, *e.g*. VFI and image reconstruction, all of pixels should be paid attentions. For VFI, to synthesize the information of each pixel, it is crucial to obtain the relatively dense in temporal dimension and complete in spatial dimension motion field between input frames and target frames.

To address this issue, we first propose to utilize spiking camera for VFI as shown in Fig. 1. Spiking camera integrates the photons at each pixel location, and spike is reported when accumulated photons exceed the threshold. Spike streams are these spikes in binary data format with high temporal resolution. Spike stream can record the light intensity and continuous changes of all pixels in high-speed dynamic scenes based on this working mechanism. In other words, spike stream tells us what happened of every pixel between two consecutive RGB frames. So, it is more robust to solve the problem of artifacts in complex motion areas by utilizing relatively dense and complete motion field extracted from spike stream.

This work aims to study a new pipeline SVFI which first introduces spiking camera to effectively eliminate artifacts caused by occlusion and motion blur. Given consecutive RGB frames and the corresponding binary spike stream between them, SVFI divides inputs into spiking branch and RGB frames branch for their domain gaps. In spiking branch, SVFI first extracts light intensity feature from spike stream via low resolution light intensity feature extraction (LR-LIFE) module. Then spatial resolution alignment (SRA) module is applied to light intensity feature because the spatial resolution of spiking camera is lower than conventional camera's. The feature with the same spatial resolution as RGB frames is fed to high resolution structure feature extraction (HR-SFE) to obtain feature which contain scene structure and outlines of all objects. In RGB branch, SVFI extracts color and texture features from RGB inputs. The generated features from two branches are fused and decoded into the target frame. In addition, few texture feature from spike stream is upscaled to high spatial resolution to complement the texture feature generated from RGB frames.

We test SVFI on different datasets, *e.g*. Middlebury(Baker et al. 2011), Vimeo90k(Xue et al. 2019) and GoPro(Nah, Kim, and Lee 2017). Experiment results show that, our SVFI achieves superior performance compared with SOTA works. For instance, in 7 and 15 frame skip evaluations on GoPro(Nah, Kim, and Lee 2017), it shows up to 5.58 dB and 6.56 dB improvements in terms of PSNR over the corresponding second best methods BMBC(Park et al. 2020) and DAIN(Bao et al. 2019). Meanwhile, SVFI trained by synthetic data shows robustness to real-world data. To the best of our knowledge, the novel pipeline SVFI, contains some specifically designed modules which aim to bridge the domain gaps, is the first attempt to combine spiking camera into VFI.

Related Work

Flow-Based VFI. Using optical flow estimation(Sun et al. 2018; Ilg et al. 2017) is one of the most conventional strategies for VFI. To interpolate a new frame from two consecutive frames, the information available is very limited, and the quality of result also highly depends on the accuracy of optical flow estimation. Therefore, flow-based methods will produce motion blur. Moreover, the quality of synthesized frames are trapped in occlusion problem. To address these issues, some works attempt to leverage extra concepts. Context-Aware Synthesis(Niklaus and Liu 2018) warps input frames and their pixel-wise contextual information. Super SloMo(Jiang et al. 2018) combines bidirectional optical flow with visibility maps which are introduced to solve the occlusion problem. (Xue et al. 2019) introduces spatial transformation network to warp frames. DAIN(Bao et al. 2019) gives different weights of overlapped flow vectors depending on the object depth of the scene via flow projection. Softmax Splatting(Niklaus and Liu 2020) and BMBC(Park et al. 2020) adds forward warping and dynamic blending filter respectively. Under a cycle consistency constraint, (Reda et al. 2019) optimizes neural network by unsupervised methods. XVFI(Sim, Oh, and Kim 2021) proposes a lightweight framework using pyramid structure to handle larger motion at a small time cost. However, these methods are under the assumptions of linear motion and brightness constancy between frames. Recently, some works assume quadratic(Xu et al. 2019) or cubic(Chi et al. 2020) motion, but they are still limited by their order and can not capture arbitrary motions.

Kernel-Based VFI. Kernel-based methods think that convolution processing can replace optical flow estimation. Compared with flow-based methods, it is more robust to motion blur. AdaConv(Niklaus, Mai, and Liu 2017a) estimates a spatially-adaptive convolution kernel for each pixel with a deep fully convolution neural network, but it consumes large memory. Then the author proposes SepConv(Niklaus, Mai, and Liu 2017b) to estimate two 1D kernels in horizontal and vertical directions respectively instead of 2D kernel to fix it.

Event-Based VFI. Event camera(Lichtsteiner, Posch, and Delbruck 2008; Daniel and Fessler 2000) is biologically inspired visual sensor. It samples the brightness change of each pixel with high temporal resolution. Its unique sampling method allows event cameras to capture contours of high-speed moving objects. Time Lens(Tulyakov et al. 2021) brings event camera to VFI and achieves comparable performance in areas with significant motions.

Spiking Camera

In this section, we present the spike generation mechanism. For spiking camera, the charges proportional to light intensity $I(t)$ on each pixel are accumulated independently. A spike is fired when the accumulative value exceeds the dispatch threshold σ. The integrator is reset and all the charges on it are drained, restarting a new "accumulate-and-fire" cy-

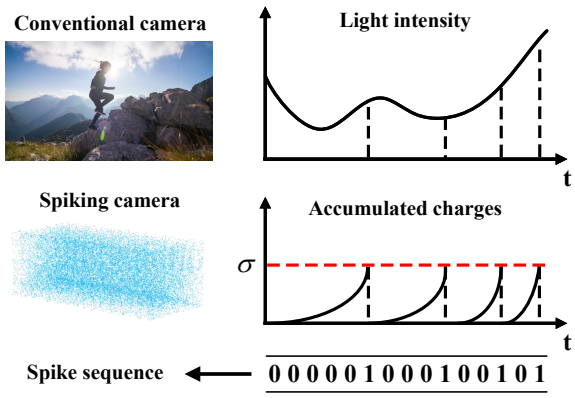

Figure 2: Illustration of working mechanism for spiking camera. The right side shows the light intensity and accumulated charges on a pixel.

cle. The instantaneous electric charge amount on the integrator can be formulated as:

$$A(t) = \int_0^t \zeta \cdot I(x)dx \mod \sigma \tag{1}$$

where ζ is photoelectric conversion rate. Unlike event cameras that use differential model so that they only record relative intensity changes of motion areas, spiking camera records the absolute light intensity of the whole scene.

Fig. 2 illustrates the spike generation for one pixel. Theoretically, a spike can be fired at arbitrary time, when the accumulated charges reach the threshold. But in practice, the time of spike reading is quantified, and a pixel can only read the spike as a discrete-time signal. Given a pixel, when the accumulated charges reach the threshold, its corresponding spike sign is set. Spiking camera uses a high-speed polling to check the spike signs on all the pixels, generating a sequence of spike stream (*e.g.* an $H \times W \times N$ binary spike stream $S(x, y, k)$) to record the instantaneous light intensity information with high temporal resolution and low latency. To be specific, given a pixel for the pixel (x, y), if a spike sign has been set at the k-th polling, it reads out $S(x, y, k) = 1$ and resets the spike sign immediately. Otherwise, it reads out $S(x, y, k) = 0$.

Method

Problem Statement

Assume we are given as input two consecutive RGB frames I_0 and I_1, as well as the spike stream $S_{0\to1}$ between them. Note that the $S_{0\to1}$ includes all spikes from t_0 to t_1. The spatial resolution of the spike frame is generally lower than that of the RGB frames. We aim to interpolate a new frame $\hat{I}_\tau$ at time τ in-between the RGB frames.

Overview

To handle occlusion and high-speed complex motions in dynamic scenes, we incorporate spiking camera in VFI and propose a novel dual-modal end-to-end neural network. The framework consists of six components: (1) *Low resolution light intensity feature extraction (LR-LIFE)* module is utilized to process the input spike stream to generate coarse estimation of light intensity feature at given timestamp τ in low resolution; (2) *Spatial resolution alignment (SRA)* module aims to upsample the output of LR-LIEF from low resolution to high resolution; (3) *High resolution structure feature extraction (HR-SFE)* module is connected to SRA, extracting the structure feature at time τ in high resolution under supervision; (4) *Frame encoder* extracts the color and high-frequency information *e.g.* textures of the RGB frames; (5) *Feature fusion* module fuses the feature maps obtained from dual-modal data; (6) *Synthesizer* module decodes the output of feature fusion module, generating a estimated frame $\hat{I}_\tau$. In this way, SVFI utilizes the spike stream captured by the spiking camera to synthesize a new RGB frame with clear textures and realistic colors in high-speed nonlinear motion scenes. Besides, we empirically demonstrate that problems, *e.g.* light intensity change, motion blur and occlusion, can be well addressed with more intermediate information incorporated. The overall workflow is illustrated in Fig. 3. Hereinafter, the proposed network is divided into spiking branch and RGB frames branch. Note that the synthesizer module only consists of several convolution layers, and we do not highlight it in the following.

Spiking Branch

LR-LIFE. It is unwise to directly fuse two modal data into one network since they have different data distributions. Thus, some preprocesses are required for spike stream to provide more compatible features. Inspired by the impressive performance of ResNet (He et al. 2016), we use Resnet as backbone to design a module, LR-LIFE, that extracts the light intensity features of low resolution from the spike stream. The module selects the sub-sequence s_τ from $S_{0\to1}$ according to the time τ by a temporal mask, and extracts the light intensity features at time τ from s_τ:

$$C_\tau^L = \mathcal{L}(M \circ S_{0\to1}) \tag{2}$$

here M is temporal mask, $\circ$ represents element-wise multiplication.

SRA. The spatial resolution of spiking camera is usually lower than that of conventional cameras. However, when it comes to binary data format, directly interpolating the spike stream to improve the spatial resolution can cause a lot of errors. Hence, the low resolution light intensity features extracted by LR-LIFE need to be aligned with the spatial resolution of RGB frames. Given that DeconvNet(Noh, Hong, and Han 2015) has promising performance in capturing specific shapes in semantic segmentation tasks, the main component of SRA is the deconvolution layer so that it can accurately preserve edge information while improving the spatial resolution. The output of SRA, $C_\tau^H \in \mathbb{R}^{H\times W\times 1}$, is the high resolution estimate at time τ.

HR-SFE. We stacked several residual blocks to extract features in spatial resolution of RGB frames. For the channel of C_τ^H output by SRA is 1, a convolution layer is added before the residual blocks as a channel bridge. Moreover, a long-skip connection between the beginning and end of the

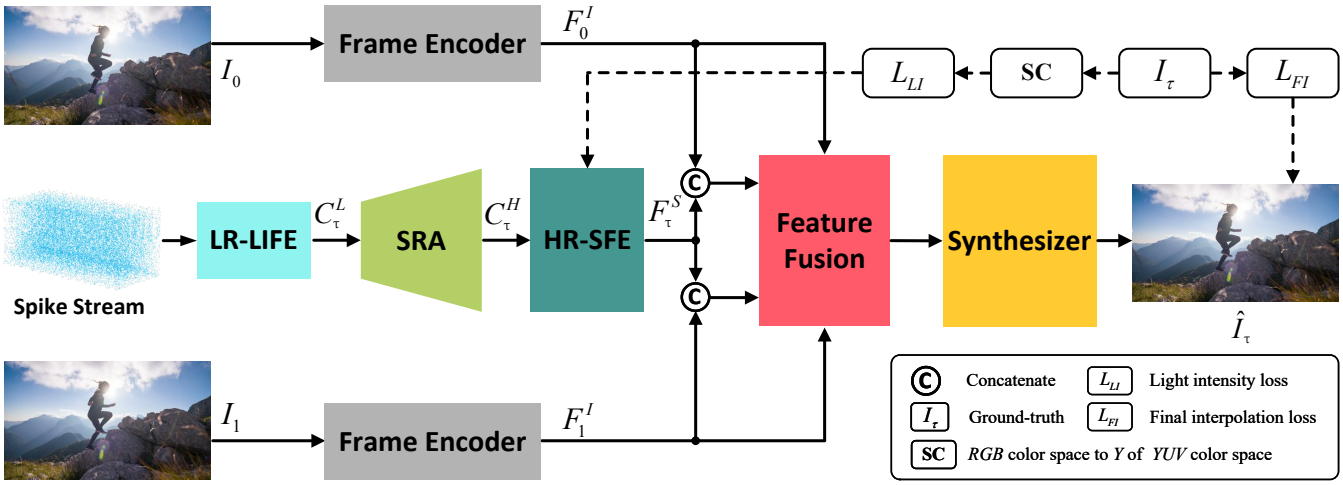

Figure 3: Illustration of the proposed dual-modal end-to-end video interpolation network, which consists of six modules: Low resolution light intensity feature extraction (LR-LIFE), Spatial resolution alignment (SRA), High resolution structure feature extarction (HR-SFE), Frame encoder, Feature fusion and Synthesizer. The inputs are two consecutive RGB frames and a spike stream between them. We show loss functions that we use to train the network.

residual blocks is used for forwarding information and preventing gradient explosion. SRA module might lose some high-frequency features in the process of improving the spatial resolution. Thus, we obtain structure features of scenes, *e.g.* position and edge information, and part of textures by HR-SFE from the output of SRA.

RGB Frames Branch

Frame Encoder. With the help of the spike stream, the spatial position and edge features of objects have been extracted. To synthesize high-frequency and color information, frame encoder is utilized to process the RGB frames:

$$F_i^I = \mathcal{E}(I_i), \quad i = 0, 1 \tag{3}$$

Here $\{I_0, I_1\}$ is the RGB frames. The architecture of frame feature extraction is similar to that of high resolution structure feature extraction. Therefore, the output of them can be matched in subsequent feature fusion module. In particular, to reduce the complexity of the whole network, the parameters of $\mathcal{E}(\cdot)$ are shared across the two RGB frames.

Feature Fusion. In order to combine the edge and spatial position features F_τ^S with the color and high-frequency features $\{F_0^I, F_1^I\}$, we propose a feature fusion module. For $\{I_0, I_1\}$ contain different contents, this module adopts bidirectional alignment to warp $\{F_0^I, F_1^I\}$ to the features of time τ. $\{K_i^F\}, i = 0, 1$ are the hidden features of target frame I_τ fused by F_τ^S and $\{F_i^I\}, i = 0, 1$. It can be calculated by:

$$\{K_i^F\} = \mathcal{J}(F_i^I \odot F_\tau^S, F_i^I), \quad i = 0, 1 \tag{4}$$

where $\odot$ represents concatenation operation.

Deformable convolution(Dai et al. 2017; Zhu et al. 2019) has been widely used for aligning features for its promising transformation modeling capability without explicit motion estimation. Hence, we perform the deformable convolutions to warp features in this module. In addition to $\{F_0^I, F_1^I\}$, the offsets and attention masks are the other inputs of deformable convolutions. The deformable convolution can be formulated as:

$$K_i^F(\mathbf{x}) = \sum_{\mathbf{p} \in N_\mathbf{x}} K(\mathbf{p}) \cdot F_i^I(\mathbf{x} + \mathbf{p} + \Delta\mathbf{p}) \cdot w_i(\mathbf{x}, \mathbf{p}) \tag{5}$$

here, $\mathbf{x} \in \mathbb{R}^{H \times W}$ is the center point of the convolution operation, $K(\cdot)$ is the convolution kernel, and $N_\mathbf{x}$ defines the respective filed size and dilation. $\Delta\mathbf{p}$ is the offsets learned to augment the spatial sampling locations:

$$\Delta\mathbf{p} = m_i(\mathbf{x}, \mathbf{p}) = \mathcal{O}(F_i^I \odot F_\tau^S) \tag{6}$$

In Eq. (5), $w_i(\mathbf{x}, \mathbf{p})$ is calculate by:

$$w_i(\mathbf{x}, \mathbf{p}) = Sigmoid(\mathcal{W}(F_i^I \odot F_\tau^S)) \tag{7}$$

which is the attention masks learned to weight the spatial sampling locations. $\mathcal{W}(\cdot)$ is a convolution block before the sigmoid operation.

Loss Function

In order to synthesize an intermediate frame with clear scene structure, the total loss we use is a weighted sum of two loss functions:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{FI}} + \lambda \cdot \mathcal{L}_{\text{LI}} \tag{8}$$

$$\mathcal{L}_{\text{FI}} = \rho(\hat{I}_\tau - I_\tau) \tag{9}$$

$$\mathcal{L}_{\text{LI}} = \rho(C_\tau^H - I_\tau^Y) \tag{10}$$

here $\rho(x) = \sqrt{x^2 + \epsilon^2}$ is the Charbonnier penalty function(Charbonnier et al. 1994), we set the constant ϵ to $1e-6$, I_τ^Y is the luminace channel in YUV color space of I_τ. The first term is used to describe the discrepancy between I_τ and $\hat{I}_\tau$. The second term is used on C_τ^H to obtain high-quality light intensity features in high resolution.

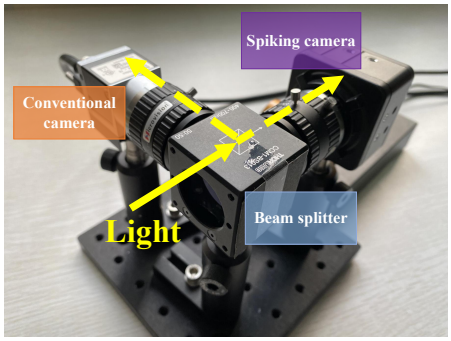

Figure 4: The proposed hybrid camera system is composed of a conventional camera and a spiking camera. The light that enters the beam splitter is divided into two branches and transmitted to the conventional camera and the spiking camera.

Experiments

Datasets

Training Datasets. *Vimeo90k (interpolation)*(Xue et al. 2019) is used for training the proposed network. *Vimeo90k (interpolation)* consists of three consecutive frames per-scene, we train the network by skipping 1 frame. However, *Vimeo90k (interpolation)* only contains RGB frames. To train the network, it still needs a spike stream between two input RGB frames. We use optical flow to generate a nearly continuous dynamic optical scene between RGB frames, and then use the spiking camera simulator proposed in (Zhao et al. 2021) to generate the corresponding spike streams. The example of simulated spike stream is showed in supplementary materials.

Testing Datasets. To evaluate the performance of our proposed method, we conduct experiments on both synthesized data and real-world data. For syntheszied data, we use popular video interpolation benchmark datasets, such as *Vimeo90k (interpolation)*(Xue et al. 2019), *GoPro*(Nah, Kim, and Lee 2017). These datasets do not provide the corresponding event stream. In order to compare with the event-based method, *High Speed Event and RGB camera (HS-ERGB) dataset*(Tulyakov et al. 2021) which contains RGB frames and the corresponding event stream is used as one of the testing datasets. And the spike streams of all testing datasets are generated by simulating. It is worth noting that the above simulation method is applied to every two consecutive RGB frames of the entire RGB sequences.

For real-world data, we capture several RGB frames and spike streams through the proposed hybrid camera system as shown in Fig. 4.

Implementation Details

All experiments of our work are implemented using the Pytorch framework.[1] For training, we use Adam optimizer(Kingma and Ba 2014) with default setting and the batch size is set to 8. SVFI is trained via total of 130 epochs with initial learning rate of 10^{-4}, reduced by a factor of 2 at [90, 110]-*th* epoch. The hyperparameter λ of loss function is set to 0.5. Considering that the spatial resolution of conventional cameras is higher than that of spiking camera, we simulate spike stream whose spatial resolution is half of the RGB frame. In *LR-LIFE* module, 21 consecutive spike frames centered on time τ are selected from input spike stream to extract the light intensity feature. We train our network on two NVIDIA Tesla V100 GPUs, which takes about 4 days to converge.

[1]Code is available at https://github.com/Bosserhead/SVFI

Metric	A_1	A_2	A_3	Ours
Spiking branch	✓		✓	✓
RGB branch		✓	✓	✓
Feature fusion		✓		✓
PSNR	-	24.84	33.60	**34.66**
SSIM	-	0.776	0.948	**0.952**

Table 1: Interpolation quality for the four sets of ablation experiments on GoPro test dataset. The best result is highlighted.

Figure 5: The effect of two branches and feature fusion module. It is clear that spiking branch can estimate precise scenes structure but can not provide color information. The feature fusion module is effective in color alignment.

Ablation Study

To study the contribution of our proposed method to the final interpolation, we investigate the effect of the two branches and feature fusion module. We implement three sets of ablation experiments, noted as A_1, A_2 and A_3. A tick indicates that the corresponding sub-network structure is included in the set of experiment. In A_3, we replace the feature fusion module with simple concatenation and convolution layers. In particular, we do not calculate the PSNR and SSIM in A_1, only show the visualized result. Because the output of A_1 is single channel, different from others three channels outputs. The PSNR and SSIM of other sets are shown in Table 1. All the synthesized results are shown in Fig. 5.

From Fig. 5, we note that it is unable to synthesize color frame from spike stream. Moreover, due to the low resolution of spike stream, the output of A_1 loses a few high-frequency information during the upsampling process. The output of A_2 shows that the network without spiking branch can not estimate precise motions. A_3 verifies that the feature fusion module helps color alignment. From Table 1, we also observe that our network outperforms A_2 by about 9.82 dB, which validates that spike stream plays an important role in our method. Competing A_3 with our network, we notice that ours achieves better performance than A_3, demonstrating that feature fusion module is beneficial for improving

Method	RGB	Events	Spikes	PSNR / SSIM	PSNR / SSIM
Middlebury(Baker et al. 2011)				**1 frame skip**	**3 frames skip**
DAIN(Bao et al. 2019)	✓	✗	✗	30.87 / 0.899	26.67 / 0.838
SuperSloMo(Jiang et al. 2018)	✓	✗	✗	29.75 / 0.880	26.43 / 0.823
CAIN(Choi et al. 2020)	✓	✗	✗	29.58 / **0.932**	24.88 / 0.854
RRIN(Li, Yuan, and Wang 2020)	✓	✗	✗	31.08 / 0.896	27.18 / 0.837
BMBC(Park et al. 2020)	✓	✗	✗	30.83 / 0.897	26.86 / 0.834
SVFI(ours)	✓	✗	✓	**33.87** / 0.925	**31.29 / 0.887**
Vimeo90k(Xue et al. 2019)				**1 frame skip**	**3 frames skip**
DAIN(Bao et al. 2019)	✓	✗	✗	34.20 / 0.962	- / -
SuperSloMo(Jiang et al. 2018)	✓	✗	✗	32.93 / 0.948	- / -
CAIN(Choi et al. 2020)	✓	✗	✗	34.48 / **0.972**	- / -
RRIN(Li, Yuan, and Wang 2020)	✓	✗	✗	34.72 / 0.962	- / -
BMBC(Park et al. 2020)	✓	✗	✗	34.56 / 0.962	- / -
XVFI-Net(Sim, Oh, and Kim 2021)	✓	✗	✗	34.90 / 0.967	- / -
SVFI(ours)	✓	✗	✓	**37.78** / 0.964	- / -
GoPro(Nah, Kim, and Lee 2017)				**7 frames skip**	**15 frames skip**
DAIN(Bao et al. 2019)	✓	✗	✗	28.81 / 0.876	24.39 / 0.736
SuperSloMo(Jiang et al. 2018)	✓	✗	✗	28.98 / 0.875	24.38 / 0.747
CAIN(Choi et al. 2020)	✓	✗	✗	28.54 / 0.898	23.22 / 0.791
RRIN(Li, Yuan, and Wang 2020)	✓	✗	✗	28.96 / 0.876	24.32 / 0.749
BMBC(Park et al. 2020)	✓	✗	✗	29.08 / 0.875	23.68 / 0.736
XVFI-Net(Sim, Oh, and Kim 2021)	✓	✗	✗	27.64 / 0.853	22.25 / 0.677
SVFI(ours)	✓	✗	✓	**34.66 / 0.952**	**30.97 / 0.912**

Table 2: Comparison of quantitative results on standard video interpolation benchmarks. Note, for *Vimeo90k(interpolation)* contain 3 frames in each test subset, we can not implement 3 frames skip tests. For these test datasets only contain RGB frames, we compare the proposed method with RGB-only methods on them. ✓ and ✗ indicate whether to use.

Method	RGB	Events	Spikes	PSNR / SSIM	PSNR / SSIM
HS-ERGB(Tulyakov et al. 2021)				**5 frames skip**	**7 frames skip**
DAIN(Bao et al. 2019)	✓	✗	✗	27.92 / 0.780	27.13 / 0.748
SuperSloMo(Jiang et al. 2018)	✓	✗	✗	25.66 / 0.727	24.16 / 0.692
CAIN(Choi et al. 2020)	✓	✗	✗	27.57 / 0.876	24.99 / 0.833
RRIN(Li, Yuan, and Wang 2020)	✓	✗	✗	25.26 / 0.738	23.73 / 0.703
BMBC(Park et al. 2020)	✓	✗	✗	25.62 / 0.742	24.13 / 0.710
Time Lens(Tulyakov et al. 2021)	✓	✓	✗	33.13 / 0.877	32.31 / 0.869
SVFI(ours)	✓	✗	✓	**33.98 / 0.919**	**32.84 / 0.903**

Table 3: Comparison of quantitative results on test set of *High Speed Event and RGB camera (HS-ERGB)*(Tulyakov et al. 2021) dataset. For it contains RGB frames and event stream, we add a comparative experiment of the event-based method.

the quality of synthesized frame.

Comparison with State-of-the-Art Methods

We compare the proposed method to six state-of-the-art frame-based methods DAIN(Bao et al. 2019), SuperSloMo(Jiang et al. 2018), CAIN(Choi et al. 2020), RRIN(Li, Yuan, and Wang 2020), BMBC(Park et al. 2020), XVFI(Sim, Oh, and Kim 2021) and a event-based method Time Lens(Tulyakov et al. 2021).

Quantitative Evaluation. To compare different VFI methods quantitatively, we use peak signal to noise ratio (PSNR) and structural similarity (SSIM) as image quality assessment (IQA) metrics to evaluate the performance.

The performance on the popular datasets which only provide RGB frames as illustrated in Table. 2. We notice that SVFI almost achieves the best performance on them. In particular, our SVFI achieves a PSNR gain over 4 dB. As illustrated in Table. 3, a comparison with the event-based method is added to the *HS-ERGB* dataset containing RGB frames and event data. The result shows that our SVFI is better than event-based and achieve the state-of-the-art performance. Our SVFI has been validated by extensive datasets

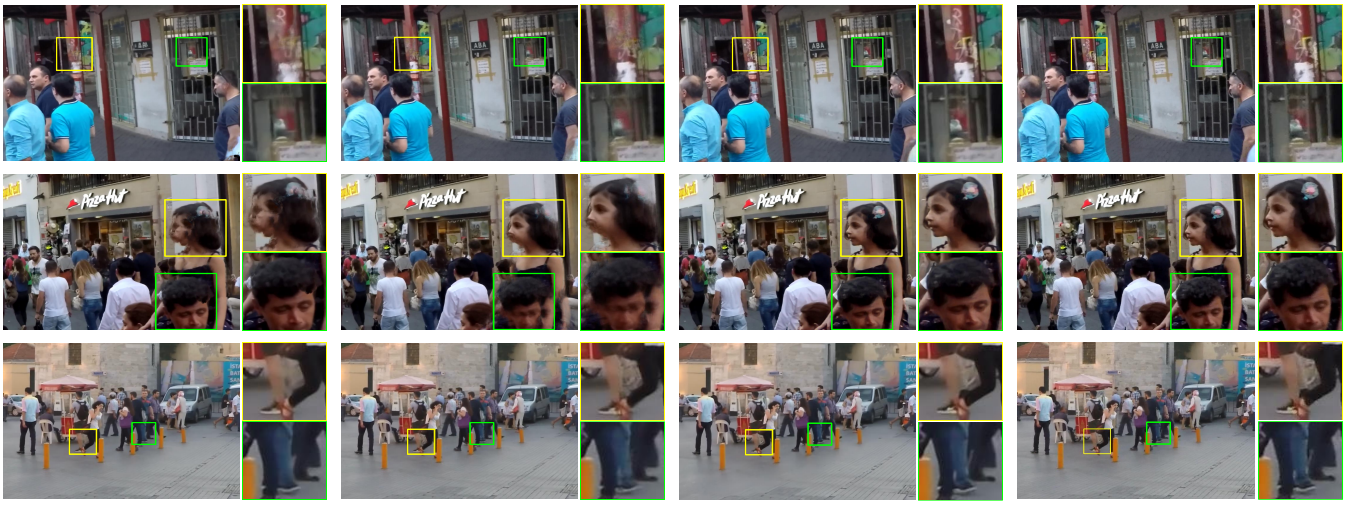

Figure 6: Qualitative results for our proposed method and its closes competitor BMBC(Park et al. 2020) and the latest method XVFI-Net(Sim, Oh, and Kim 2021) on the 7 frames skip GoPro test dataset.

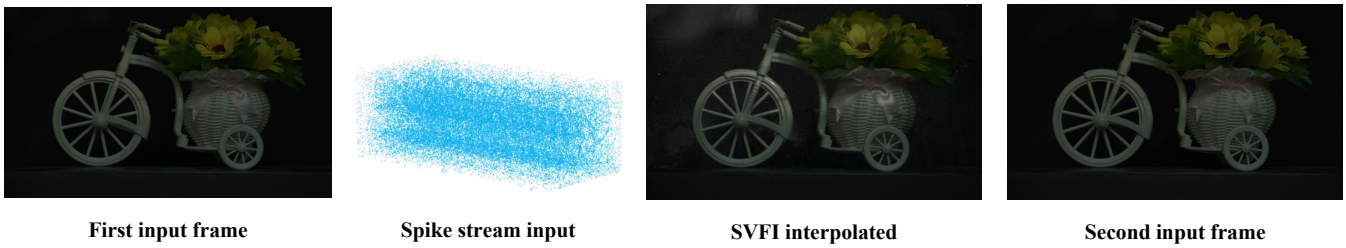

Figure 7: The intermediate frame interpolated by our proposed method on real-world data.

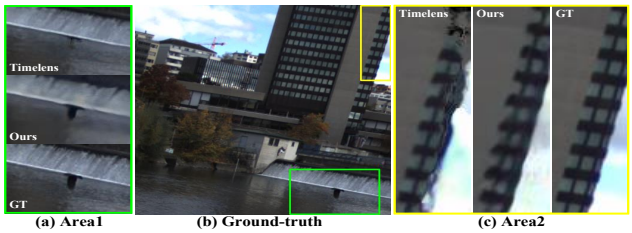

Figure 8: Qualitative results of event-based method and ours.

and demonstrates its effectiveness.

Qualitative Evaluation. Fig. 6 and Fig. 8 visualize the results of RGB-only methods with our SVFI and event-based method with ours, respectively. Especially, we generate high-speed motion by skipping 7 frames between input RGB frames. As shown in Fig. 6, for high-speed moving objects, BMBC and XVFI-Net can not estimate accurate motions, which leads to artifacts in synthesized frames. In addition, there are some blurs due to the occlusion in the synthesized frames of the baselines, *e.g.* in the area of brand logo on shoes in bottom scene. In the bird area (green box area) of Fig. 8, our SVFI exhibits robustness to objects with relatively sparse textures and small motion. Fig. 7 shows the result of SVFI on real-world data.

In summary, SVFI shows robust and outstanding performance, even for the scenes contain objects with occlusion and complex motions.

Conclusion

This paper presents a novel dual-modal method for VFI. Distinct from the previous works, we introduce a neuromorphic spiking camera to VFI. The spiking camera uses continuous spike stream to record the dynamic visual scenes at extremely high temporal resolution, which can provide additional intensity cues for the intermediate frame. In particular, to exploit the dual-modal information effectively, we design several tailored modules to bridge the domain gaps on spatial resolution, data modality. Experiments on a wide variety of datasets show our method achieves the state-of-art performance. Our method also achieves promising visual quality on real-world data captured by our hybrid camera system.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2021YFF0900501 and in part by the National Natural Science Foundation of China under Grants 62072009 and 22127807.

References

Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M. J.; and Szeliski, R. 2011. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1): 1–31.

Bao, W.; Lai, W.-S.; Ma, C.; Zhang, X.; Gao, Z.; and Yang, M.-H. 2019. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3703–3712.

Charbonnier, P.; Blanc-Feraud, L.; Aubert, G.; and Barlaud, M. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing (ICIP)*, volume 2, 168–172. IEEE.

Chi, Z.; Mohammadi Nasiri, R.; Liu, Z.; Lu, J.; Tang, J.; and Plataniotis, K. N. 2020. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 107–123. Springer.

Choi, M.; Kim, H.; Han, B.; Xu, N.; and Lee, K. M. 2020. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 10663–10671.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 764–773.

Daniel, F. Y.; and Fessler, J. A. 2000. Mean and variance of single photon counting with deadtime. *Physics in Medicine & Biology*, 45(7): 2043.

Gui, S.; Wang, C.; Chen, Q.; and Tao, D. 2020. Featureflow: Robust video interpolation via structure-to-texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14004–14013.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2462–2470.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. *Advances in Neural Information Processing Systems*, 28: 2017–2025.

Jiang, H.; Sun, D.; Jampani, V.; Yang, M.-H.; Learned-Miller, E.; and Kautz, J. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9000–9008.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, H.; Yuan, Y.; and Wang, Q. 2020. Video frame interpolation via residue refinement. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2613–2617. IEEE.

Lichtsteiner, P.; Posch, C.; and Delbruck, T. 2008. A 128× 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-state Circuits*, 43(2): 566–576.

Liu, Z.; Yeh, R. A.; Tang, X.; Liu, Y.; and Agarwala, A. 2017. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4463–4471.

Nah, S.; Kim, T. H.; and Lee, K. M. 2017. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3883–3891.

Niklaus, S.; and Liu, F. 2018. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1701–1710.

Niklaus, S.; and Liu, F. 2020. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5437–5446.

Niklaus, S.; Mai, L.; and Liu, F. 2017a. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 670–679.

Niklaus, S.; Mai, L.; and Liu, F. 2017b. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 261–270.

Noh, H.; Hong, S.; and Han, B. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1520–1528.

Park, J.; Ko, K.; Lee, C.; and Kim, C.-S. 2020. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 109–125. Springer.

Pourreza, R.; and Cohen, T. 2021. Extending Neural P-Frame Codecs for B-Frame Coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 6680–6689.

Reda, F. A.; Sun, D.; Dundar, A.; Shoeybi, M.; Liu, G.; Shih, K. J.; Tao, A.; Kautz, J.; and Catanzaro, B. 2019. Unsupervised video interpolation using cycle consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 892–900.

Sim, H.; Oh, J.; and Kim, M. 2021. XVFI: Extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14489–14498.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), 8934–8943.

Tulyakov, S.; Gehrig, D.; Georgoulis, S.; Erbach, J.; Gehrig, M.; Li, Y.; and Scaramuzza, D. 2021. Time Lens: Event-Based Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16155–16164.

Xu, X.; Siyao, L.; Sun, W.; Yin, Q.; and Yang, M.-H. 2019. Quadratic Video Interpolation. *Advances in Neural Information Processing Systems*, 32: 1647–1656.

Xue, T.; Chen, B.; Wu, J.; Wei, D.; and Freeman, W. T. 2019. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8): 1106–1125.

Zhao, J.; Xiong, R.; Xie, J.; Shi, B.; Yu, Z.; Gao, W.; and Huang, T. 2021. Reconstructing Clear Image for High-Speed Motion Scene With a Retina-Inspired Spike Camera. *IEEE Transactions on Computational Imaging*, 8: 12–27.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9308–9316.