Learning to Super Resolve Intensity Images from Events

https://github.com/gistvision/e2sri

S. Mohammad Mostafavi I.
GIST, South Korea
mostafavi@gist.ac.kr

Jonghyun Choi
GIST, South Korea
jhc@gist.ac.kr

Kuk-Jin Yoon
KAIST, South Korea
kjyoon@kaist.ac.kr

Figure 1. Reconstructing high-definition photo-realistic intensity images from pure events in end-to-end learning. Our events to super-resolved intensity image reconstruction recovers more details with less artifacts in comparison to recent methods of EG [24] and EV [17].

Abstract

An event camera detects per-pixel intensity difference and produces asynchronous event stream with low latency, high dynamic range, and low power consumption. As a trade-off, the event camera has low spatial resolution. We propose an end-to-end network to reconstruct high resolution, high dynamic range (HDR) images directly from the event stream. We evaluate our algorithm on both simulated and real-world sequences and verify that it captures fine details of a scene and outperforms the combination of the state-of-the-art event to image algorithms with the state-of-the-art super resolution schemes in many quantitative measures by large margins. We further extend our method by using the active sensor pixel (APS) frames or reconstructing images iteratively.

1. Introduction

Event cameras, also known as neuromorphic cameras, have successfully opened their path to the computer vision and robotics society for its low cost and high dynamic sensing range with low latency and low power consumption. It represents the changes of intensity for a pixel location (x, y) as a plus or minus sign (σ) asynchronously by checking the amount of intensity changes with a predefined threshold. This stream-like representation, depending on the scene and camera movement, can achieve μs order of latency through accurate timestamps (t) and is expressed per fired event in the form of (x, y, t, σ). This device has garnered a lot of attention due to the high applicability in systems requiring high dynamic range outputs with low latency, and low power and low memory consumption constraints [15, 24, 17, 27, 5]. New applications for the event cameras have emerged such as intensity image reconstruction or recovering geometric features such as optical flow or depth from the event stream [1, 18, 10, 3, 22, 23].

Unfortunately, most commercially available event cameras produce relatively low resolution event streams for their efficiency. While there are number of proposals on many applicationsestimating super-resolved intensity images from the events has been barely explored in the literature. To generate the high resolution images from the event, one can combine a method to transfer events to inten-

sity images with a super resolution algorithm for intensity images [4, 21, 7, 14]. But these pipelined approaches are sub-optimal in generating the high resolution images from the events and may fail to reconstruct details of scenes. For producing high fidelity high resolution images, we aim to directly learn to estimate pixel-wise super-resolved intensity from events in an end-to-end manner and demonstrate that our method is able to super resolve images with rich details and less artifacts, better than pipelined state of the arts in both qualitative and quantitative analyses.

To the best of our knowledge, we are the first to model super-resolving event data to higher-resolution intensity images in an end-to-end learning framework. We further extend our method to reconstruct more details by considering APS frames as inputs or learning the network iteratively to add details to an initial image.

2. Related Work

Event to intensity images. Early attempts in the applications of event cameras, consider relatively short periods of the event stream data and direct accumulation of the plus or minus events in two colors as a gradient interpreted output [2]. Synthesising intensity images instead of the gradient representation is originated from the task of simultaneously estimating the camera movement and mosaicing them as a panoramic gradient image [10]. In their approach the scene is static and the camera only has rotational movements. By the Poisson integration they transfer a gradient image to an intensity image. In [3], a bio-inspired network structure of recurrently interconnected maps is proposed to predict different visual aspects of a scene such as intensity image, optical flow, and angular velocity from small rotation movements. In [1], a joint estimation of optical flow and intensity simultaneously in a variational energy minimization scheme in a challenging dynamic movement setting is proposed. However, their method propagates errors as shadow-like artifacts in the generated intensity images.

A variational framework based on a denoising scheme that filters incoming events iteratively is introduced in [18]. They utilized manifold regularization on the relative timestamp of events to reconstruct the image with more grayscale variations in untextured areas. In [22], an asynchronous high-pass filter is proposed to reconstruct videos in a computationally efficient manner. This framework is originally designed for complementing intensity frames with the event information but is also capable of reconstructing images from events without the help of APS frames.

Recent approaches use deep convolutional networks to create photo-realistic images directly from the event stream [24, 17]. Both approaches employ a $U\text{-}net$ [19] as their base architecture with modifications such as using conditional generative adversarial neural networks [24] or using a deep recurrent structure (up to 40 steps) together with stacked ConvLSTM gates [17]. They further investigated the possibility of reaching very high frame rates and using the output intensity images for downstream applications.

Image super resolution (SR). Intensity image SR algorithms can be largely categorized into single image SR (SISR) [4, 14] or multiple image SR (MISR) also known as video SR [21, 7]. SISR methods add details inferred from the context of the given single low resolution (LR) image while MISR further uses a sequence of images over time. Since MISR uses more LR images to reconstruct the high resolution image, it is generally more successful in recovering missing details and higher frequency information. Since we have a sequence of stacks, MISR is more similar to our approach, although we aim to reconstruct one single image each time. The learning based SR methods outperform previous methods by using deeper and wider networks while utilizing the power of residual connections to prevent vanishing gradients [14, 7]. Many MISR methods use optical flow representations among the input images as a supplementary source of input to reach higher quality SR outputs [21, 7]. Inspired by these methods, we design our SR subnetwork as described in Sec. 3.2.4.

3. Approach

We propose a fully convolutional network that takes a sequence of events stacks near the timestamp of interest as input, relates them in pairs with their optical flow obtained by $FNet$ and rectify the combination of the paired stacks and the flow by EFR, then feeds them to the recurrent neural network based super-resolution network ($SRNet$) that outputs hidden states and intermediate intensity outputs per each stack. Finally, we mix the intermediate outputs from multiple time stamps by Mix to construct a super resolved intensity image. We briefly illustrate the structure in Fig. 2 and with the detailed data flow in Fig. 3 in Sec. 3.2.1. Beginning with event stacking strategy, we describe the details of our network architecture.

3.1. Event Stacking Method

The stream-like representation of events is sparse in spatial domain and needs preparation to capture scene details to be reconstructed by a convolutional neural network. Despite recent advances of the stacking methods [5, 23], our network performs well with a simple stacking method such as *stacking based on the number of events* (SBN) [24]. Employing the advanced stacking methods is straightforward by minor modifications to the input blocks of our network.

With the SBN, starting from any timestamp in the event stream, we count the number of events until we reach a predefined number (N_e) and accumulate the events to form one *channel* in the stack. We repeat this process c times for one *stack*. Thus, each stack contains $M = c \times N_e$ events in

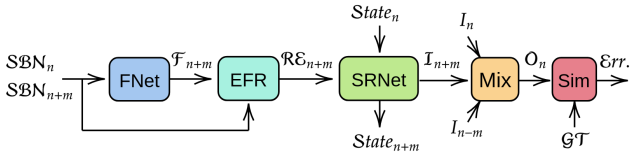

Figure 2. Overview of our end-to-end event to super-resolved intensity image framework. The input stacks SBN_{n+m} and the central stack SBN_n are given to the FNet to create the optical flow (F_{n+m}). The flow and stacks are concatenated and given to the EFR to rectify the event features. Its output RE_{n+m} is given to SRNet together with the previous state ($State_n$) to create intermediate intensity outputs I_{n+m} and the next state ($State_{n+m}$). All intermediate intensity outputs are concatenated and given to the mixer (Mix) network which creates the final output (O_n). Finally, the output is compared to the training groundtruth (GT) using the similarity loss (Sim) including Learned Perceptual Image Patch Similarity (LPIPS) term and ℓ_1 term to compute error (Err).

total and has the dimension of $h \times w \times c$, where h and w are the width and height of the APS images, respectively. This c-channel stack is fed into the network as an input. The corresponding APS frame is sampled at the timestamp of the last event in the stack for the ground truth (GT). At each channel, all pixel values are initially set to 128. If an event is triggered at location (x, y), we replace the pixel value at (x, y) in the same channel with 256 (positive event) or 0 (positive event). Since newly coming events can override older events, the M needs to be carefully chosen to better preserve spatio-temporal visual information. The frame rate can be determined by both the N_e and the number of overlapping events between each stack over time.

We empirically choose to use $3,000$ events per stack in which each stack has 3 channels. This number can be modified for the experiments with larger resolution event inputs to ensure that the average number of events in stacks show visually plausible outputs with fine details. However, since the network is trained on diverse scenes which contain different numbers of local events, the network is not very sensitive to the chosen number of events per stack at inference.

3.2. Network Architecture

We design the network architecture by three principles. First, we take into account the characteristics of the input and target (Sec. 3.2.1). Second, we have a sufficiently large hypothesis space for the super-resolution network ($SRNet$) to address various level of complexity of movements in a scene (Sec. 3.2.4). Finally, we propose a novel objective function that can add structural details while being away from blur and artifacts (Sec. 3.2.6). We describe the details of each component of our proposed network.

3.2.1 Overview

We consider a stream of events stacked for the input to our network. In particular, for the input sequence of three stacks ($3S$), the stacks are the one containing the n^{th} APS timestamp (SBN_n), the stack before it SBN_{n-m} and the stack after it (SBN_{n+m}). We illustrate the network with these inputs in Fig. 3 with detailed data flow through the sub-networks. Note that the network can be used with the input of any number of stacks in a sequence (*e.g.*, 3 or 7).

Each stack has M (*e.g.*, $3,000$) events and its end location m will vary on the timeline of events based on the amount of time it is required to fire M events. SBN_n is the *central stack* among the three sequences. It is fed to the network after SBN_{n-m} and the predicted intensity output is corresponding to this stack. The SBN_{n+m} and SBN_{n-m} stacks are M events away from the beginning or end of the central stack respectively if there is no overlap ($L = 0$) among the stacks ('Non-overlapped' input in Fig. 3). We can also have overlapping stacks for creating higher framerates; the end of the next stack will be M events after the center minus the amount of overlap (M–L) ('overlapped' input in Fig. 3). More details on the overlapped stacking is provided in the supplement.

SBN_{n+m} and SBN_{n-m} are fed separately with the central stack to the optical flow estimation network ($FNet$) to predict the optical flow (F_{n+m} or F_{n-m}) between the stacks. These stacks of events are concatenated with the optical flow obtained by the $FNet$ and then rectified by an event feature rectification network (EFR). The rectified event stack (RE_{n+m}) is then given to the super-resolution network ($SRNet$). The $SRNet$ takes the previous state ($State_n$) with the rectified events stack (RE_{n+m}) and creates the next state ($State_{n+m}$) of the sequential model and a super-resolved intensity like output (I_{n+m}).

Since the stacks quantize continuous event stream into separate inputs, each stack may not contain all necessary details for reconstructing images. Thus, the intermediate intensity outputs from all the stacks are then mixed by a Mixer network (Mix) to reconstruct intensity image O_n with rich details. For the initial stack, only the first stack is fed to the EFR sub-network to create an initial $State_n$. The output of Mix is given to the similarity network (Sim) to optimize the parameters based on the error (Err).

3.2.2 Flow Network (FNet)

An unwanted downside of stacking the event stream is losing temporal relation between the stacks. The lost temporal relation between stacks can be partially recovered by using a sequence of the stacks and the optical flow between each pair of stacks as the optical flow reports how the triggered events in the scene have moved and in which location the changes have happened. The SBN stacking includes sufficient edge information and can be used as an image-like input to well-known learning-based optical flow estimation algorithms. Thus, we do not finetune it but use a pretrained

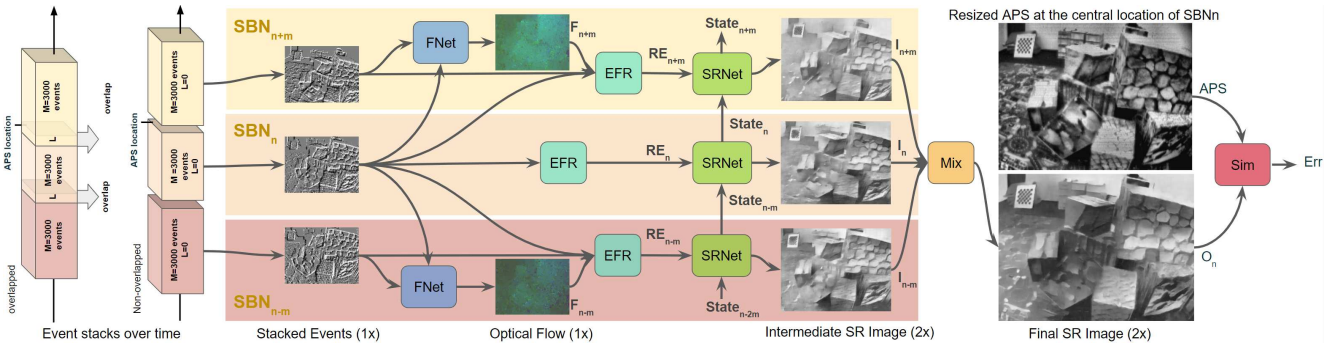

Figure 3. Detailed data flow in the proposed method. This example is based on third stack (SBN_{n+m}), therefore the previous inputs, optical flow, and intermediate intensity outputs are faded. The APS frame is resized to the size of output (O_n) for comparison

$FNet$ for computational efficiency[1]. We use [9] as our flow estimation network and call it as $FNet$.

3.2.3 Event Feature Rectification Network (EFR)

Another downside of stacking events is overwriting previous event information in fast triggering locations. The overwritten events result in a blurry stack of events and eventually lower quality reconstructions. To prevent overwriting events, we concatenate two stack of events with the optical flow and provide it to two convolutional layers called the event feature rectification (EFR) network. By the EFR, we progressively fuse the stacks over the event stream to preserve details from each event.

The EFR helps to reconstruction images when two stacks have events in a location visible to only one stack which the optical flow cannot relate, the events will more likely be maintained for the intensity reconstruction since we use all three inputs by the EFR. Note that the central stack is provided to this network without estimated flow since there is no flow for it.

3.2.4 Super Resolution Network (SRNet)

The rectified events are now super resolved by our main network called $SRNet$. We use a recurrent neural network for the $SRNet$ because each part of the event stream which we stack captures details of the output image and they are originally continuous but quantized by the stacking method. To alleviate the discontinuity, we utilize the internal memory state of recurrent neural network to reconstruct different regions with rich details in a continuous manner as the state is updated internally by each incoming stack. Specifically, a single event stack might partially miss important details from previously fired events which are not in its stacking range but have been captured by the previous stacks.

It has been shown that stacked events are capable of synthesizing intensity images by deep neural networks [24, 17] such as $U\text{-}net$ [19]. Architecturally, we further extend the idea by using $ResNet$ [8] with 15 blocks in depth with more filters and larger kernel size. In particular, following the well-designed networks in MISR [14, 21, 6, 4], we utilize the power of residual learning for super-resolving intensity. We use large field of views inspired from the SISR network [6] to transfer the rectified event features to SR intensity generator ($RNet\text{-}C$). Its main task is to create an initial SR intensity image state by the combination of transposed convolutional operations.

The $SRNet$ is designed to upscale the input RE while adding intensity information. The overall structure of the $SRNet$ is illustrated in Fig. 4. We use the combination of three residual networks ($RNet{-}\{A, B, D\}$) that are composed of five ResNet blocks containing two convolutional layers. These networks are shallower than $RNet\text{-}C$ because they encode feature-like representations from previous states and not directly from the rectified events. The output of $RNet\text{-}A$ which performs as an upsampling encoder is subtracted from the output of $RNet\text{-}C$ to create an internal error (e_n), which measures how much the current rectified event stack RE_{n+m} contributes in comparison to the previous state $State_n$ as

$$e_n = RNet\text{-}C(RE_{n+m}) - RNet\text{-}A(State_n). \quad (1)$$

This error is given as an input to $RNet\text{-}B$ which performs as a general encoder. We define the the next state ($State_{n+m}$) by the output of $RNet\text{-}B$ summed with $RNet\text{-}C$ thus the current input (RE_{n+m}) is emphasized as

$$State_{n+m} = RNet\text{-}B(e_n) + RNet\text{-}C(RE_{n+m}). \quad (2)$$

The $State_{n+m}$ is given to a final decoder ($RNet\text{-}D$) to make the intermediate intensity output (I_{n+m}) as

$$I_{n+m} = RNet\text{-}D(State_{n+m}). \quad (3)$$

In general, the $RNet\text{-}C$ adds new information from the current stack to the previous state by adding details of the scene missed by the previous stack. Even when there is no events in some regions captured by the current stack but there are

[1]Finetuning $FNet$ may further improve the output quality as the stacked image has different visual signature from natural images.

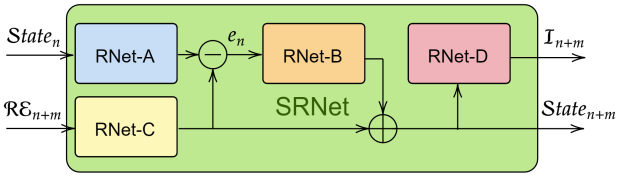

Figure 4. Detailed architecture of the proposed super resolving network ($SRNet$) (Green-block in Fig. 2). Four main residual networks are designed to perform as a large encoder-decoder scheme. $RNet\text{-}A$ is used to update the hidden state while $RNet\text{-}B$ and $RNet\text{-}D$ act as an encoder and decoder respectively to map the hidden state as a super resolved intensity output (I_{n+m}).

scene details in the regions captured by the previous stack, the previous state ($State_n$) holds that information through $RNet\text{-}A$ as its hidden state to reconstruct the scene details in the regions rather missing. We detail other design parameters such as layer type, number of filters in the supplement.

3.2.5 Mixer Network (Mix)

The Mixer network is designed to augment the outputs (I_i) of the SRNet at different time locations (i=$\{n-m, n, n+m\}$) to reconstruct detail-rich intensity image (O_n) at the central stack's timestamp (n). This network employs convolutional layers to reconstruct the intensity image with fine details.

3.2.6 Similarity Loss (Sim)

Given a reconstructed image (O) and its GT (G), we define a loss function with two terms. First, we use an unstructured loss such as the ℓ_1 norm to reconstruct overall sharper images as $\mathcal{L}_{\ell_1}(O,G) = \|O-G\|_1$ rather than ℓ_2 which results in smoothed edges with low frequency texture in output images. As the ℓ_1 may lose the structural information of a scene, we further leverage a criterion capable of compensating the lack of structure by the Learned Perceptual Image Patch Similarity (LPIPS) or perceptual similarity [26] as the second term of our objective function. Specifically, given a pair of images (O, G) encoded by a pretrained network (*e.g.*, AlexNet [11]), the near end features ($\hat{G}^l_{hw}$) of the l^{th} layer are extracted while its activations are normalized by the channel dimension (H_l, W_l). Then, each channel is scaled by a vector w_l [26], and the ℓ_2 distance is computed. Finally, a spatial mean is computed over the image axes (h, w) through all layers (l) for the LPIPS loss as

$$\mathcal{L}_{LPIPS}(O,G) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (\hat{O}^l_{hw} - \hat{G}^l_{hw})\|_2^2. \quad (4)$$

The final objective function, $\mathcal{L}_{sim}$, is the combination of the both terms with a balancing parameter λ as

$$\mathcal{L}_{sim}(O,G) = \mathcal{L}_{\ell_1}(O,G) + \lambda \mathcal{L}_{LPIPS}(O,G), \quad (5)$$

which we minimize to learn the parameters.

4. Experiments and Analyses

For the empirical validation, we use generated sequences using the event camera simulator (ESIM) [16] and four challenging and diverse real-world public datasets.[1, 15, 22, 27]. We describe the details of our dataset in the supplement. For the quantitative analyses, we use PSNR in dB (logarithmic scale), the structural similarity [25] (SSIM) as a fraction between zero (less similar) to one (fully similar), the mean squared error (MSE), and the perceptual similarity (LPIPS) as a metric to evaluate the similarity of the high level features in two images (lower the value, more the similarity). For each experiment, we train our network on a cluster of 8 Titan-Xp GPUs. Batch size is 8 and initial learning rate is 0.01 which is decayed by a factor of 10 at every half of the remaining epochs of the given maximum number of epochs (*e.g.*, 50 in our experiments). We use $\lambda = 0.01$ for all our experiments, otherwise mentioned.

4.1. Comparison with State of the Arts

We are the first to propose the task of direct reconstruction SR intensity image from events thus there are no directly comparable methods. So, we first down-sample our outputs and compare to same-size intensity reconstruction methods to evaluate the quality of our reconstruction. Then we compare our method to the state-of-the-art intensity reconstruction methods combined with the state-of-the-art super-resolution (SR) methods.

Image reconstruction without super-resolution. We compare down-sampled outputs of our method to the state-of-the-art event to intensity image methods on seven challenging real-world sequences from the Event Camera dataset [15]. For notation brevity, we abbreviate the high pass filter method [22] as HF, manifold regularization [18] as MR, event to video generation [17] as EV and event to intensity by conditional GANs as EG [24]. Following the evaluation protocols in many real-world event datasets [15, 22, 27], we consider APS frame as GT. We follow the sequence split of [17] and use the reported performance measures of HF, MR and EV. For EG, we used the authors' reconstructed images to evaluate the performance.

As shown in Table 1, our proposed method outperforms all other methods in LPIPS. It implies that the reconstructed intensity image is perceptually better than the previous methods. Our method also exhibits higher SSIM scores on multiple sequences and comparable MSE errors to EG. Similar to EV, we train the model only with the synthetic sequences and apply to real world sequences. In this challenging zero-shot data transfer setting without fine-tuning, our method outperforms other methods on real-world events. Note that the two runner up methods in LPIPS (EV and EG) also use learning based framework.

Table 1. Comparison to state-of-the-art intensity synthesis methods on real-world sequences [15]. Our method outperforms the previous methods in all sequences in LPIPS, and on average in SSIM. The runner up method is underlined. We used the reported numbers in [17] for HF [22], MR [18] and EV [17] while evaluated the authors' reconstructed images for EG [24].

	SSIM (↑)					MSE (↓)					LPIPS (↓)				
Sequence	HF [22]	MR[18]	EV[17]	EG[24]	Ours	HF [22]	MR[18]	EV[17]	EG[24]	Ours	HF [22]	MR[18]	EV[17]	EG[24]	Ours
dynamic_6dof	0.39	**0.52**	0.46	0.48	0.44	0.10	0.05	0.14	**0.03**	0.05	0.54	0.50	0.46	0.45	**0.42**
boxes_6dof	0.49	0.45	**0.62**	0.45	0.61	0.08	0.10	0.04	0.03	**0.02**	0.50	0.53	0.38	0.48	**0.32**
poster_6dof	0.49	0.54	0.62	0.61	**0.63**	0.07	0.05	0.06	**0.01**	0.02	0.45	0.52	0.35	0.42	**0.29**
shapes_6dof	0.50	0.51	**0.80**	0.56	0.79	0.09	0.19	0.04	0.03	**0.01**	0.61	0.64	0.47	0.51	**0.38**
office_zigzag	0.38	0.45	0.54	0.67	**0.68**	0.09	0.09	0.03	0.01	0.01	0.54	0.50	0.41	0.36	**0.29**
slider_depth	0.50	0.50	0.58	0.54	0.59	0.06	0.07	0.05	0.02	0.02	0.50	0.55	0.44	0.42	**0.34**
calibration	0.48	0.54	0.70	0.67	**0.71**	0.09	0.07	0.02	0.01	0.01	0.48	0.47	0.36	0.42	**0.24**
Average	0.46	0.50	0.62	0.57	**0.64**	0.08	0.09	0.05	0.02	0.02	0.52	0.53	0.41	0.43	**0.33**

Table 2. Quantitative comparison of super-resolved intensity images from events directly (Ours) to events to intensity image synthesis (EV) combined with SISR [4] and MISR[7] methods.

Method	PSNR (↑)	SSIM (↑)	MSE (↓)	LPIPS (↓)
EV + SISR 2×	11.292	0.384	0.348	0.394
EV + MISR 2×	11.309	0.385	0.347	0.392
Ours 2×	**16.420**	**0.600**	**0.108**	**0.172**
EV + SISR 4×	11.168	0.396	0.089	0.543
EV + MISR 4×	11.293	0.384	0.087	0.396
Ours 4×	**16.068**	**0.560**	**0.028**	**0.253**

Super-resolved image reconstruction. We now combine state-of-the-art event to intensity reconstruction algorithms with state-of the-art SR methods and compare our method to them. For the state-of-the-art event to intensity algorithm, we use EV[2] since it is the runner up method that outperforms EG in SSIM and LPIPS in most of the sequences and on average (Table 1). For super resolution algorithms, we use two recent super-resolution algorithms; one for SISR [4] and another for MISR [7]. As shown in Table 2, our method outperforms the state-of-the-art intensity reconstruction algorithms combined with the state-of-the-art SR algorithms in all metrics by large margins. We use 30 sequences from our generated dataset by ESIM.

For qualitative analyses, we demonstrate intensity reconstruction by EV, the combination of EV+MISR and our method on real-world and simulated sequences in the Fig. 5 and Fig. 1. Note that our method reconstructs fine details from events. In Fig. 1, EG does not always reconstruct scene details from the events and sometimes hallucinates jittery artifacts. While EV reconstructs scene details from the events relatively better than EG, it creates a shadow-like artifact and darkens some areas of the scene. Furthermore, in the presence of hot pixels in the data, EV does not filter them; white or black dots appear in the results by EV while our method mostly filters them out without explicit operations to remove. We present more results in the supplementary material.

We further conduct experiments on the sequences from another popular dataset [1] and qualitatively compare our method to EG and EV in Fig. 6. Our method can reveal details that is not visible in constructing the same sized images such as fingertips or texture.

[2]Publicly available at https://github.com/uzh-rpg/rpg_e2vid.

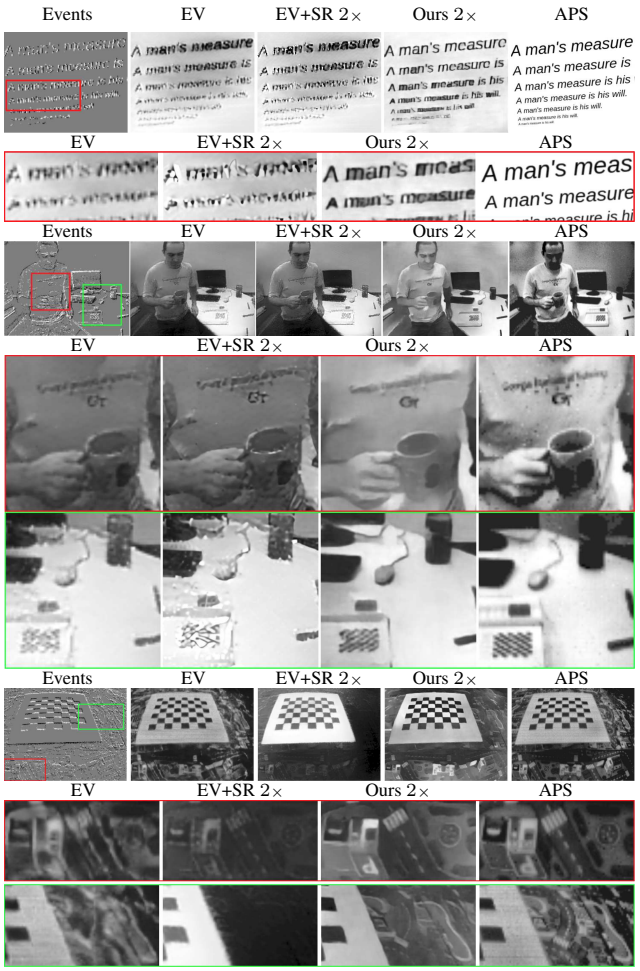

Figure 5. Qualitative comparison among synthesizing SR intensity images directly (ours) and super-resolving as a downstream application to intensity image estimation (EV+MISR). Highlighted boxes are zoomed for better comparison.

4.2. Analysis on Loss Terms ($\mathcal{L}_{sim}$)

We ablate the loss function to investigate the effect of each terms on image reconstruction quantitatively in Ta-

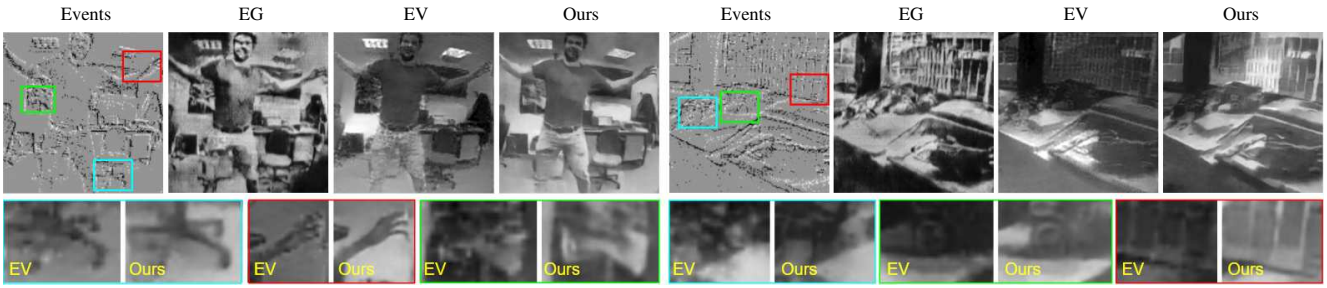

Figure 6. Qualitative comparison of our downscaled outputs to EV ands EG on sequences from [1] (without APS). Our method is able to reconstruct structural details from inputs as small as 128×128 pixels. More results are provided in the supplementary material.

Table 3. Ablation study of the loss function.

Loss	PSNR (↑)	SSIM (↑)	MSE (↓)	LPIPS (↓)
$\mathcal{L}_{\ell_1}$	**15.33**	0.517	0.034	0.485
$\mathcal{L}_{LPIPS}$	10.06	0.388	0.454	**0.232**
$\mathcal{L}_{sim}$ (Full)	15.03	**0.528**	**0.032**	0.258

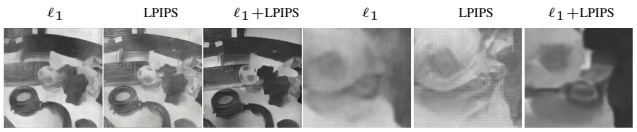

Figure 7. Effect of loss function on reconstruction quality. ℓ_1 norm smooths edges, perceptual similarity (LPIPS) adds structural details but also creates artifacts. The combination of ℓ_1+$LPIPS$ ($\mathcal{L}_{sim}$) shows less artifacts while adding structural details.

Figure 8. Effect of number of stacks and scale factor.

Scale	# Stacks	PSNR (↑)	SSIM (↑)	MSE (↓)	LPIPS (↓)
$2\times$	3S	15.46	0.554	0.323	0.191
	7S	**16.42**	**0.600**	**0.108**	**0.172**
$4\times$	3S	15.03	0.528	0.032	0.258
	7S	**16.06**	**0.560**	**0.028**	**0.253**

ble 3 and qualitatively in Fig. 7. All analyses and ablation studies were performed with the simulated data for reliable quantitative analyses with high quality GT. Using only $\mathcal{L}_{\ell_1}$ term, we observe better performance in PSNR but leads to visually less sharp images thus low performance in all other metrics. Using only $\mathcal{L}_{LPIPS}$ term, we observe that images look visually acceptable but with the downside of lower PSNR with dot-like artifacts on regions with less events and on the edges. The final proposed loss function $\mathcal{L}_{sim}$ performs the best in SSIM and MSE with a slight decrease in PSNR and LPIPS but creates visually the most plausible images.

4.3. Analysis on Super Resolution Parameters

We evaluate the effect of two SR parameters; the upscale factor ($2\times, 4\times$) and size of the sequence of stacks ($3S, 7S$) on the output quality. We summarize the results in Table 8. Comparing $3S$ and $7S$, we observe that $7S$ results in better performance in all metrics. It implies that a longer recursion on the sequences may produce more reliable hidden states and results in better quality output. Also, when using longer sequences, it is more likely to capture events that happen only for a short period of time since unrolling on a larger recursion helps to keep information of short events. It is more challenging to super resolve events to larger images as it is not trivial for an algorithm to handle large spatial locations where no events exist. Although the MSE has decreased, compared to $2\times$, it is because the number in the denominator is larger due to the size of the image and not much related to the output quality.

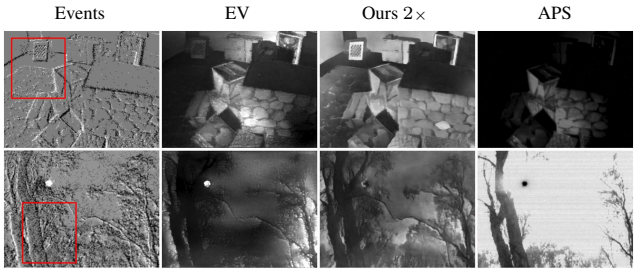

Figure 9. Image reconstruction comparison in extreme HDR scenarios [15, 22]. Our method synthesizes more details while producing less artifacts compared to EV and the APS. Please zoom in and compare the suggested red boxes.

4.4. Qualitative Analysis on HDR Sequences

One challenging scenario using the event camera is to capture events under extreme dynamic range. We qualitatively analyze outputs under such extreme conditions and compare them to EV in Fig. 9. Normal cameras including the APS frame have much lower dynamic range and either create black regions (when the camera misses to sense intensity details under its sensing range as shown in the top row) or white regions (when light floods in the camera and the camera cannot sense higher than its sensing range as shown in the bottom row). We observe that our method can address a higher range and reveal more structural details that EV and the APS frame fail to capture.

4.5. Analysis on the Failure Modes

Failure cases are mostly related to missing background details over long trajectories when the foreground objects

Table 4. Temporal stability error evaluation (Eq. 6). Plus sign indicates blind post-processing [13]. Our method ($3S$, $7S$) does not directly consider temporal consistency however longer sequences of stacks ($7S$) are more consistent. EV[17] uses up to L=40 input stacks and is initially more consistent. However, we get lower errors even on our smallest sequence after post-processing.

$E_{warp}(\downarrow)$	APS	$3S$	$7S$	EV [17]	$3S+$	$7S+$	EV [17]+
dynamic_6dof	0.61	20.35	16.54	**8.78**	**3.42**	3.71	5.56
boxes_6dof	1.81	16.69	17.51	**15.69**	**3.58**	3.95	9.36
poster_6dof	1.10	18.80	22.66	**17.74**	**4.41**	5.91	5.56
shapes_6dof	0.44	24.00	21.23	**16.66**	2.80	**2.63**	8.33
office_zigzag	0.08	3.62	2.19	**0.72**	0.36	**0.34**	0.44
slider_depth	0.02	0.57	0.34	**0.19**	0.06	**0.04**	0.12
calibration	0.36	15.46	9.72	**2.99**	**1.31**	1.24	1.62
Average	0.63	14.21	12.89	**8.97**	**2.28**	2.55	5.20

have rapid movements. In such sequences, our method only recovers parts of the scene that are in a limited temporal distance to our central stack. We showcase and further analyze a number of failure modes in the supplementary material.

5. Extensions

Video reconstruction. We aim to reconstruct a single image not a video. So, the temporal consistency between frames are out of our interest thus not always held. To extend our method to video reconstruction, we utilize a blind post-processing method [12] to encode temporal consistency among the intensity images and demonstrate the qualitative results in a supplementary video. To quantitatively evaluate the temporal consistency, we follow the temporal stability metric from [13], which is based on the flow warping error between two consecutive synthesized frames (F_t, F_{t+1}):

$$E_{warp}(F_t, F_{t+1}) = \frac{1}{\sum_{i=1}^{N} M_t^{(i)}} \sum_{i=1}^{N} M_t^{(i)} ||F_t^{(i)} - \hat{F}_{t+1}^{(i)}||_2^2, \quad (6)$$

where $\hat{F}_{t+1}$ is the warped frame of F_{t+1} and $M_t \in \{0, 1\}$ is the non-occlusion mask based on [20] to ensure that the calculations are applied only in the non-occluded regions. We compute the optical flow used for warping the frame and the non-occlusion map based on the APS frames for evaluating the warping errors of all methods compared and APS as they are the GT. We summarize the results with different size of sequences ($3S$ and $7S$) in comparison to EV in Table 4. While our methods ($3S$ and $7S$) are worse than EV due to lack of temporal consistency, a simple post-processing ($3S+$ and $7S+$) significantly improves the performance, outperforming both the EV [17] and its post-processed version ($EV+$) by large margins.

Complementary and Duo-Pass. To evaluate our method in a challenging set-up, we do not use APS frame to super resolve images. Using APS frame, we can further improve quality of output. We name the extension by using APS frame as *Complementary* [22] or *Comp.* We train the initial state of network with the low resolution (LR) APS frame as a central stack (Sec. 3.2.1) and provide events as its nearby stacks. We observe that the network learns to add higher resolution details from the LR input.

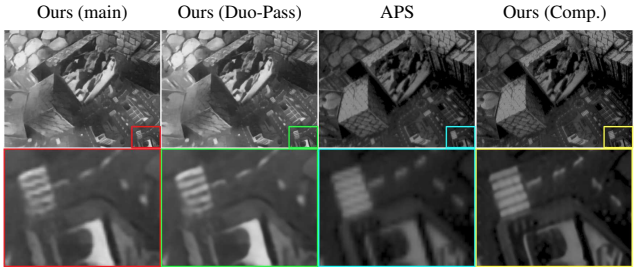

Figure 10. Extensions. Duo-Pass that iterates the SR twice and Complementary (Comp.) that uses events with APS frames.

However, the Complementary method is sensitive to the quality of central stack, specifically if it is blurry or noisy, its artifacts are propagated to the final reconstruction. To avoid such shortcoming, we propose another extension that does not use APS frames but use two iterations or passes from events only, called *Duo-Pass*. In the first pass, we use the main scheme to create intensity images from events only. In the second pass, we use the synthesized intensity image from the first pass as the central stack similar to that we use the APS frame in the Complementary method. By the Duo-Pass, we are able to further recover HR details that the first pass misses without the help of the APS frame. We qualitatively compare the results by our method (main), by the Duo-Pass and by the Comp. in Fig. 10. We provide more results in the supplementary material.

6. Conclusion

We propose to directly reconstruct higher resolution intensity images from events by an end-to-end neural network. We demonstrate that our method reconstructs high quality images with fine details in comparison to the state of the arts in both the same size image reconstruction and super-resolution. We further extend our method to the *Duo-Pass* which performs an extra pass to add missing details and the *Complementary* that utilizes APS frames in addition to events. We also reconstruct videos by our method with a simple post-processing to ensure temporal consistency.

Acknowledgement. This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2019R1C1C1009283) and (NRF-2018R1A2B3008640), the Next-Generation Information Computing Development Program through the NRF funded by MSIT, ICT (NRF-2017M3C4A7069369), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)) and (No.2019-0-01351, Development of Ultra Low-Power Mobile Deep Learning Semiconductor With Compression/Decompression of Activation/Kernel Data).

References

[1] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *IEEE CVPR*, pages 884–892, 2016. 1, 2, 5, 6, 7

[2] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 2

[3] Matthew Cook, Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika Steger. Interacting maps for fast visual interpretation. In *The 2011 IJCNN*, pages 770–776. IEEE, 2011. 1, 2

[4] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *IEEE CVPR*, pages 11065–11074, 2019. 2, 4, 6

[5] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *IEEE ICCV*, 2019. 1, 2

[6] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *IEEE CVPR*, pages 1664–1673, 2018. 4

[7] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *IEEE CVPR*, pages 3897–3906, 2019. 2, 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 4

[9] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE CVPR*, pages 2462–2470, 2017. 4

[10] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43:566–576, 2008. 1, 2

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 5

[12] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018. 8

[13] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *IEEE ECCV*, pages 170–185, 2018. 8

[14] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE CVPR workshops*, pages 136–144, 2017. 2, 4

[15] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1, 5, 6, 7

[16] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982, 2018. 5

[17] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE T-PAMI*, 2019. 1, 2, 4, 5, 6, 8

[18] Christian Reinbacher, Gottfried Graber, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. In *BMVC*, 2016. 1, 2, 5, 6

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 2, 4

[20] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, pages 26–36. Springer, 2016. 8

[21] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *IEEE CVPR*, pages 6626–6634, 2018. 2, 4

[22] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *ACCV*, pages 308–324. Springer, 2018. 1, 2, 5, 6, 7, 8

[23] Stepan Tulyakov, Francois Fleuret, Martin Kiefel, Peter Gehler, and Michael Hirsch. Learning an event sequence embedding for dense event-based deep stereo. In *IEEE ICCV*, pages 1527–1537, 2019. 1, 2

[24] Lin Wang, Mostafavi I. S. Mohammad, Yo-Sung Ho, and Kuk-Jin Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *IEEE CVPR*, pages 10081–10090, 2019. 1, 2, 4, 5, 6

[25] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *ACSSC*, volume 2, pages 1398–1402. Ieee, 2003. 5

[26] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE CVPR*, pages 586–595, 2018. 5

[27] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE RA-L*, 3(3):2032–2039, 2018. 1, 5