Event-based High Dynamic Range Image and Very High Frame Rate Video Generation using Conditional Generative Adversarial Networks

Lin Wang[1]*, S. Mohammad Mostafavi I.[2]*, Yo-Sung Ho, and Kuk-Jin Yoon[1]
[1]Visual Intelligence Laboratory, Dept. Mechanical Engineering, KAIST, Korea
[2]Computer Vision Laboratory, Dept. EECS, GIST, Korea
wanglin@kaist.ac.kr, mostafavi@gist.ac.kr, hoyo@gist.ac.kr, kjyoon@kaist.ac.kr

Abstract

Event cameras have a lot of advantages over traditional cameras, such as low latency, high temporal resolution, and high dynamic range. However, since the outputs of event cameras are the sequences of asynchronous events over time rather than actual intensity images, existing algorithms could not be directly applied. Therefore, it is demanding to generate intensity images from events for other tasks. In this paper, we unlock the potential of event camera-based conditional generative adversarial networks to create images/videos from an adjustable portion of the event data stream. The stacks of space-time coordinates of events are used as inputs and the network is trained to reproduce images based on the spatio-temporal intensity changes. The usefulness of event cameras to generate high dynamic range (HDR) images even in extreme illumination conditions and also non blurred images under rapid motion is also shown. In addition, the possibility of generating very high frame rate videos is demonstrated, theoretically up to 1 million frames per second (FPS) since the temporal resolution of event cameras are about 1 μs. Proposed methods are evaluated by comparing the results with the intensity images captured on the same pixel grid-line of events using online available real datasets and synthetic datasets produced by the event camera simulator.

1. Introduction

Event cameras are bio-inspired vision sensors that mimic the human eye in receiving the visual information [14]. While traditional cameras transmit intensity frames at a fixed rate, event cameras transmit the changes of intensity at the time of the changes, in the form of asynchronous events that deliver space-time coordinates of the intensity changes. They have lots of advantages over traditional cameras, *e.g.* low latency in the order of microseconds, high temporal resolution (around 1 μs) and high dynamic range. However, since the outputs of events cameras are the sequences of asynchronous events over time rather than actual intensity images, most existing algorithms cannot be directly applied. Thus, although it has been recently shown that event cameras are sufficient to perform some tasks such as 6-DoF pose estimation[24] and 3D reconstruction [22, 11], it will be a great help if we can generate intensity images from events for other tasks such as object detection, tracking and SLAM.

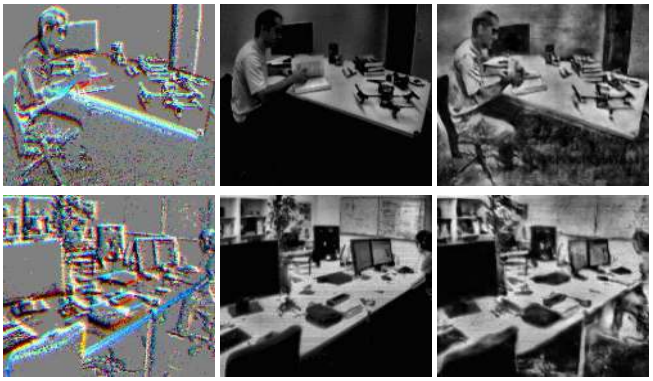


Figure 1. From left to right, input events, active pixel sensor (APS) images from the DAVIS camera, and our results. Our methods construct HDR images with more details that normal cameras could not reproduce as in APS frames.

Actually, it has been stated that event cameras, in principle, transfer all the information needed to reconstruct images or a full video stream [2, 25, 24]. However, this statement has never been thoroughly substantiated. Motivated by recent advances of deep learning in image reconstruction and translation, we tackle the problem of generating intensity images from events, and further unlock the potential of event cameras to produce high quality HDR intensity images and high frame rate videos with no motion blur, which is especially important when the robustness to fast motion and to extreme illumination conditions is critical as in autonomous driving. To the best of our knowledge, our work is the first attempt focusing on pure events to HDR images and high frame

*These two authors contributed equally

rate video translation, and proving that event cameras can produce high-quality non-blurred images and videos even under fast motion and extreme illumination conditions. We first propose the event-based domain translation framework that generates better quality images from events compared with active pixel sensor (APS) frames and other previous methods. For this framework, two novel and initiative event stacking methods are also proposed based on shifting over the event stream, stacking based on time (SBT) and stacking based on the number of events (SBE), such that we can reach high frame rate and HDR representation with no motion blur, which is, in contrast, impossible for the normal cameras. It turns out that it is possible to generate a video with up to 1 million FPS using these stacking methods.

To verify the robustness of the proposed methods, we conduct intensive experiments and evaluation/comparison. In experiments, real datasets from a dynamic and active-pixel vision sensor, DAVIS, which is a joint event and intensity camera [20], are used. The sensor's pixel grid-line of the events and the intensity are on the same location which helps reducing extra steps of rectification and warping for adjusting two images to each other. We make an open dataset that includes more than $17K$ images captured by the DAVIS camera to learn a generic model for event-to-image/video translation. In addition, we make a synthetic dataset containing $17K$ images by using the event camera simulator [23] for experiments[1].

2. Related work

2.1. Intensity-image reconstruction from events

One of the early attempts on visually interpreting or reconstructing the intensity image from events is the work by Cook *et al*. [6], in which recurrently interconnected areas called maps were utilized to interpret intensity and optic flow. Kim *et al*. [10] used pure events on rotation only scenes to track the camera and also built a super-resolution accurate mosaic of the scene based on probabilistic filtering. In [3], intensity images were reconstructed using a patch-based sparse dictionary both on simulated and real event data in the presence of noise. Bardow *et al*. [2] took a few steps further by reconstructing the intensity image and the motion field for generic motion in contrast to previous rotation only schemes. Meanwhile, Reinbacher *et al*. [25] introduced a variational denoising framework that iteratively filters incoming events. They guided the events through a manifold regarding their timestamps to reconstruct the image. The measurements and simulations on the event camera with RGBW color filters were proposed by Moeys *et al*. in [19]. They presented the naive and computational method for reconstructing the intensity image. The aforementioned methods did create intensity images mainly by pure events, however, the reconstruction was not photorealistic. Recently, Shedligeri *et al*. [28] introduced a hybrid method that fuses intensity images and events to create photorealistic images. Their method relies on a set of three autoencoders. This method produces promising results for normally illuminated scenes, but it fails in recovering HDR scenes under extreme illumination conditions since it only utilizes event data for finding the 6-DoF pose.

2.2. Deep learning on events

Although deep learning has not been much applied to event-based vision, some recent studies have demonstrated that deep learning successfully performs with event data. Moeys *et al*. [18] utilized both event data and APS images to train a convolutional neural network (CNN) for controlling the steering of a predator robot. Other methods on steering prediction for self-driving cars by using pure events and/or by incorporating the APS images in an end-to-end fashion have been also studied in [4, 15]. On the other hand, a stacked spatial LSTM network was introduced in [22], which relocalizes the 6-DoF pose from events, and the optical flow estimation based on a self-supervised encoder-decoder network was proposed in [33]. Supervised learning is adopted to create pseudo labels for detecting objects under ego-motion in [5]. The pseudo labels are transferred to the event image by training a CNN on APS images. And, as mentioned in the previous section, the fusion of event data and APS images was introduced in [28], which utilized autoencoders to create photorealistic images. To the best of our knowledge, we are the first to apply generative adversarial networks on event data.

2.3. Condition GANs on image translation

Actually, there is no qualitative research showing the effectiveness of conditional GANs (cGANs) on event data. Prior works have focused on cGANs for image prediction from a normal map[29], future frame prediction[16] and image generation from sparse annotations[9]. The difference between using GANs for image-to-image translation conditionally and unconditionally is that unconditional GANs highly rely on the confining lost function to control the output to be conditioned. cGANs have been successfully applied to style transfer [13, 1, 8, 34, 12] in the frame image domain, and these applications mostly focused on converting an image from one representation to another based on the supervised setting. Besides, it requires input-output pairs for graphics tasks while assuming some relationship between domains. When comes to event vision, cGANs have not yet been examined qualitatively and quantitatively, and therefore, we seek to unlock the potential of cGANs for image reconstruction based on event data. However, since the general approach for frame-based image translation is typically different from event-based one, we first propose a

[1] Available online at http://vi.kaist.ac.kr

deep learning framework to accomplish this task and fully take advantages of an event camera such as low latency, high temporal resolution, high dynamic range with the proposed framework. We then qualitatively and quantitatively evaluate the proposed framework with real and synthetic datasets.

3. Proposed method

To reconstruct HDR and high temporal resolution images and videos from events, we exploit currently available deep learning models, such as cGANs, as potential solutions for event vision. cGANs are generative models that learn a mapping from observed image x and random noise vector z to the output image y, $G : \{x, z\} \rightarrow y$. The generator G is trained to produce output that is not distinguishable from original images by an adversarially trained discriminator, D [7]. The objective is to minimize the distance between ground truth and output from generator, and to maximize the observation from discriminator.

cGANs such as Pix2Pix [8] and CycleGANs [34] have proved their capability in image-to-image translation bringing breakthrough results. The key strength of cGANs is that there is no need to tailor the loss function regrading given specific tasks, and it can generally adapt its own learned loss to the data domain where it is trained. However, event data is quite different from those used for traditional vision approaches based on cGANs, so we propose new methods that can provide off-the-shelf inputs for neural networks in Sec. 3.1 first and build a network in Sec. 3.2.

3.1. Event stacking

In an event camera, each event e is represented as a tuple (u, v, t, p), where u and v are the pixel coordinates and t is the timestamp of the event, and $p = \pm 1$ is the polarity of the event, which is the sign of the brightness change ($p = 0$ for no event). These events are shown as a stream on the left of Fig. 2. Based on the frame rate of intensity camera, we have synchronized APS images and asynchronous events in-between two consecutive APS frames. To feed event data input to the network, new representations of event data are required. One simple way is to form the 3D event volume as $p(u, v, t)$ for some time duration ensuring event data enough for image reconstruction. When denoting the temporal resolution of an event camera by δt and the time duration by t_d, the size of the 3D volume is (w, h, n), where w and h represent the spatial resolution of an event camera and $n = t_d/\delta t$. This is equivalent to have the n-channel image input for the network. This representation preserves all the information about events. However, the problem is that the number of channels is very huge. For example, when t_d is set to $10ms$, then n is about $10K$, which is extraordinarily large, since the temporal resolution of an event camera is about 1 μs. For this reason, we construct the 3D event volume with small n by forming each channel via merging and stacking the events within a small time interval. Event stacking can be done in different ways, but the temporal information of event is necessarily sacrificed in return.

3.1.1 Stacking Based on Time (SBT)

In this approach, the streaming events in-between the time references of two consecutive intensity images (APS) of the event camera, denoted as Δt, are merged. But not all events are merged into a single frame. Instead, the time duration of the event stream is devided into n equal-scale portions, and then n grayscale frames, $S_p^i(u, v)$, $i = 1, 2, .., n$, are formed by merging the events in each time interval $[\frac{(i-1)\Delta t}{n}, \frac{i\Delta t}{n}]$. $S_p^i(u, v)$ is the sum of polarity(p) values at (u, v). These n grayscale frames are stacked together again to form one stack $S_p(u, v, i) = S_p^i(u, v)$, $i = 1, 2, .., n$, which is fed to the network as the input. As mentioned, this stacking method loses the time information of events within time interval $\frac{\Delta t}{n}$. However, the stack itself, as the sequence of frames from one to n, still holds the temporal information to some extent. Therefore, larger n can keep more temporal information.

Figure 2 illustrates how to merge and stack the events. When $n = 3$ (*i.e.* stacking frames F_A, F_B, and F_C into one stack), the stack can be visualized as a pseudo color frame, as shown in the left part of Fig. 2 above the APS image. Based on the time shown at the event manifold in the middle of Fig. 2, starting from time zero on the 3D view, the location of APS image is around the location of the third red rectangle near 0.03 sec (the frame rate of the APS image is 33 FPS).

3.1.2 Stacking Based on the number of Events (SBE)

Unfortunately, SBT brings an intrinsic limitation originated from the event camera, which is the lack of events when there is no movement of the scene or the camera. When the event data within the time interval are not enough for the image reconstruction, it is hard to get good HDR images inevitably. This is the case for the fourth and fifth frame of the event stream at the left of Fig. 2. Furthermore, another flaw comes from the case of having too many events in one time frame as in the third time frame.

SBE more coincides with the nature of an event camera, which is being asynchronous to time, and can overcome the aforementioned limitations of SBT. In this method, a frame is formed by merging the events based on the number of incoming events as illustrated in Fig. 2. The first N_e events are merged into frame 1 and next N_e events into frame 2, and this is continued up to frame n to create one stack of n frames. Then, this n-frame stack containing nN_e events in total is used as an input to the network. This method guarantees rich event data enough to reconstruct images depending on the N_e value. F_E, F_F, F_G, and F_H in Fig. 2 are the frames corresponding to different numbers of events, N_e, $2N_e$, $3N_e$, $4N_e$, respectively. Since we count the number of events with time, we can adaptively adjust the number of events in each frame and also in one stack.

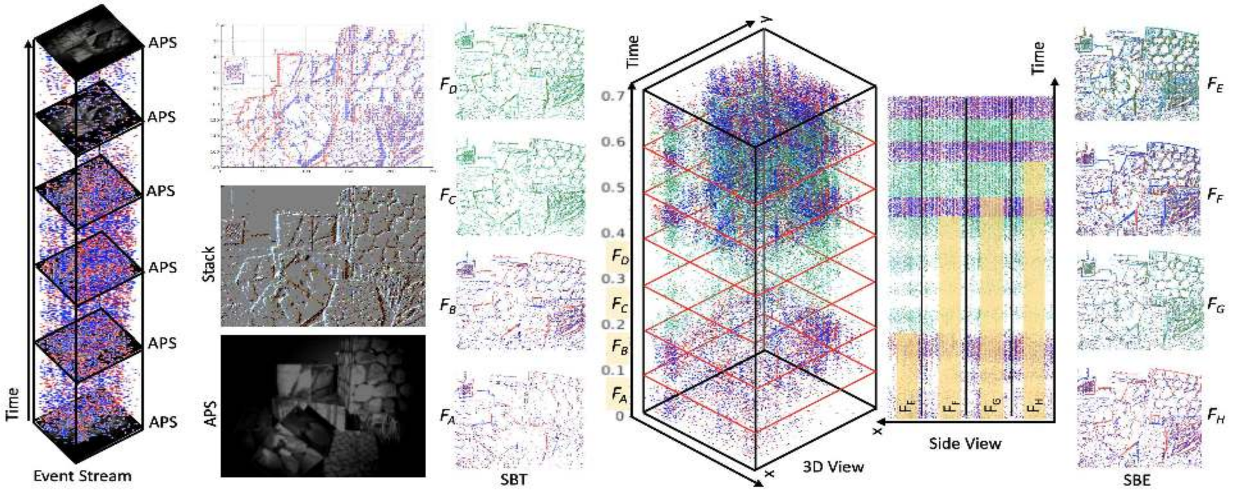


Figure 2. The event stream and construction of stacks by SBT and SBE. Two main color tuples of (Red(+), Blue(-)) and (Green(+), Cyan(-)) express the event polarity (plus, minus) throughout this paper. In the main 3D view two types of stacking (SBT on left and SBE on right) are shown using the yellow highlighted time. The 3D view followed by its side view are color coded with (Red, Blue) and (Green, Cyan) periodically (every 5000 events) for better visualization. All the images and plotted data are from the "hdr_boxes" sequence of [20].

3.1.3 Stacking for video reconstruction

Both SBT and SBE can be applied for video reconstruction from events using the proposed network, and in both methods, the frame rate of the output video can be adjusted by controlling the amount of time shift of two adjacent event stacks used as inputs to the network. When the events in the time interval $[i - \Delta t, i]$ are used for one input stack for the image $I(i)$ in a video, the next input stack for the image $I(i + t_s)$ in a video can be constructed by using the events in the time interval $[i - \Delta t', i + t_s]$ (for SBT $\Delta t' = \Delta t - t_s$), with the time shift t_s. Then, the frame rate of the output video becomes $\frac{1}{t_s}$. It is also worthy of notice that two stacks have large time overlap $[i - \Delta t', i]$ with duration $\Delta t'$. If $\Delta t' >> t_s$, the temporal consistency is naturally enforced for nearby frames. Since the temporal resolution of an event camera is about 1 μs, we can reach up to one million FPS video with temporal consistency. This will be demonstrated in Sec. 4

3.2. Network architectures

In this paper, we describe our generator and discriminator motivated by [13]. Details of the architectures including the size of each layer can be found in Fig. 3 and Fig. 4.

3.2.1 Generator architecture

The core of the event-to-image translation is how to map a sparse event input to a dense HDR output with details, sharing the same structural image features, such as edges, corners, blobs, etc. Encoder-Decoder network is the mostly used network for image to image translation tasks. The input is continuously downsampled through the network, and then upsampled back to get the translated result. Since, in the event-to-image translation problem, there is a huge amount of high-frequency important information from event data passing through the network, it is likely to lose detailed features of events during this process and induce noise to the outputs. For that reason, we consider the similar approaches proposed in [8], where we further add skip connections to the "U-net" network structure in [25]. In Fig. 3, the detailed information including number of layers and inputs/output are depicted.

3.2.2 Discriminator architecture

Our network is originated from the network in [31]. Figure 4 illustrates the details of our network architecture. Our discriminator can be considered as a method to minimize the style transfer loss between events and intensity images. Mathematically, the objective function is defined as

$$L_{cGAN}(G, D) = E_{e,g}[\log D(e, g)] + E_{e,\epsilon}[\log(1 - D(e, G(e, \epsilon)))]. \quad (1)$$

where e indicates the original event, g indicates the generated image, and ϵ indicates the Gaussian noise as input to the generator. Meanwhile, G tries to minimize the difference of images from events, and D is to maximize it. Here, for the regularization, the $L1$ norm is used to shrink blurring as

$$L_{L1}(G) = E_{e,g,\epsilon}[\|g - G(e, \epsilon)\|_1]. \quad (2)$$

This $L1$ norm is aimed to make the discriminator more focus on high-frequency structure of generated images from events. Eventually, the objective is to estimate the total loss from event-to-image translation as

$$G^* = \arg\min_G \max_D [L_{cGAN}(G, D) + \lambda L_{L1}(G)], \quad (3)$$

where λ is a parameter to adjust the learning rate. With the noise ϵ, the network could learn a mapping from event e and

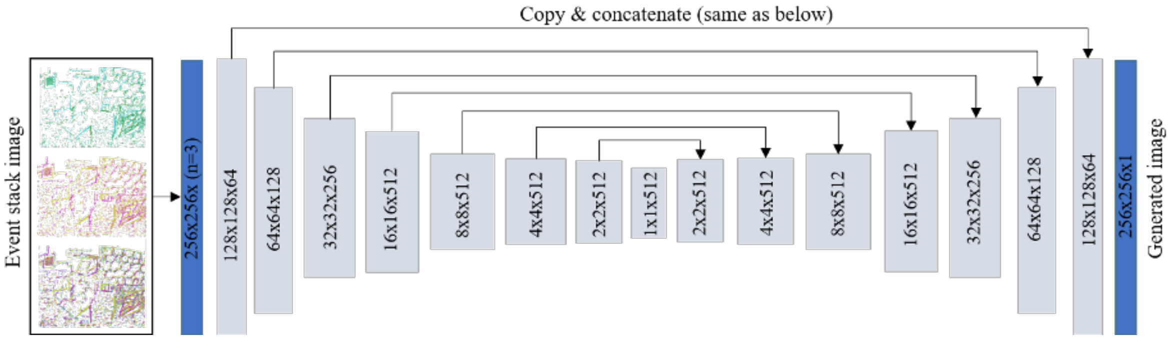


Figure 3. Generator network: A U-network[26, 8] architecture (with skip connections) that takes an input with the dimension of 256×256×n (n = 3 for this example), followed by gray boxes corresponding to multi-channel feature maps. The number of channels is denoted inside each box. The first two numbers(from bottom to top) indicate the filter sizes and the last number indicates the number of filters.

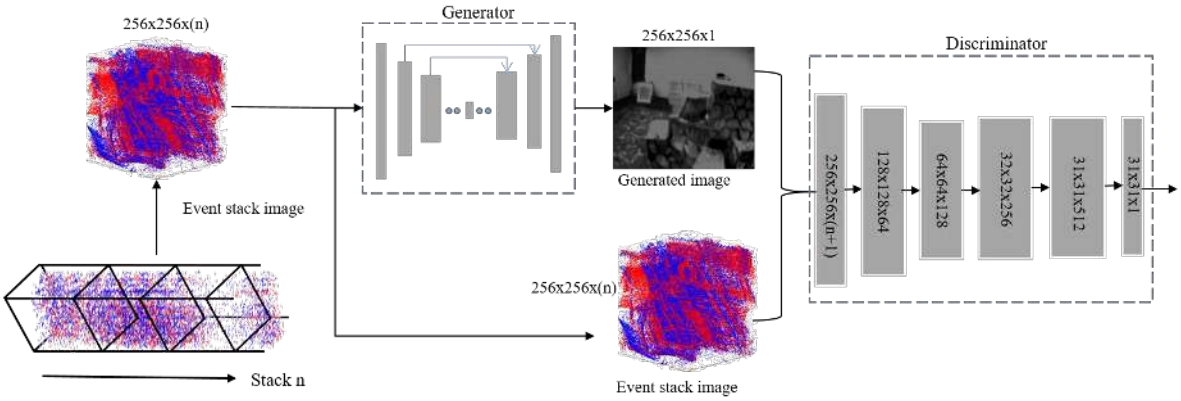


Figure 4. The proposed framework with the generator and discriminator networks. Our discriminator network is similar to PatchGAN [31], which takes two images (original APS image and the image generated by the generator from events). The discriminator first concatenates the condition of feature maps from the last layer of the generator and discriminates whether the generated image respects the condition of domain transfer from event to intensity.

ϵ to g, which could match the distribution based on events and help to produce more deterministic outputs.

3.3. Dataset preparation

Our training and test datasets are prepared based on three folds of methods. We create the first group of datasets by referring to [20], where many real-world scenes are included. We also make the second group of datasets by ourselves for various training and test purposes and also for opening to public afterwards. The datasets are captured using DAVIS camera, and have many series of scenarios. The third type of datasets is generated from ESIM[23], an open-source event camera simulator. The real datasets contain many different indoor and outdoor scenes captured with various rotations and translations of the DAVIS camera. Our training data consist of pairs of stacked events as explained in Sec. 3.1 together with the APS frames from both the real-world scenes and the ground truth (GT) frames generated in ESIM. Here, to use real data for training the network, we carefully prepare the training data to refrain the network from learning improper properties of the APS frames. Actually, APS frames suffer from motion blur under fast motion, and also have limited dynamic range resulting in the loss of details as shown in Fig. 11. Therefore, directly using the real APS frames as ground truth is not a good way for training the network, since our goal is to produce HDR images with less blur by fully exploiting the advantages of event cameras.

For that reason, the events relevant to the black and white regions of the training data are removed from the input to make the network learn to generate HDR images from events. In addition, the APS images are classified as blurred and non-blurred based on BRISQUE scores (that will be explained later) and manual inspection, and we refrain from using the blurred APS images in the training set. The simulated sequences are mainly generated from ESIM, where events are produced while a virtual camera moves in all directions to capture different scenes in given images. Since the events and APS images are generated from a controlled simulation environment, the APS frames are counted directly as the ground truth for image reconstruction. Therefore, the aforementioned training data refinement is not required for simulated datasets.

4. Experiments and evaluation

To explore the capability of our method, we conduct intensive experiments on the datasets depicted in Section 3.3, and also use another open-source dataset with three real sequences (Face, jumping, and ball) [2] for comparison. We create a training dataset about 60K event stacks with corresponding APS image pairs based on their precise timestamps, and test our method on both scenes with normal illumination and also HDR scenes. From both the real and simulated datasets, we randomly chose 1,000 APS or ground truth images with corresponding event stacks, not used in the training step, for testing. Here, it is worthy of notice that, since real datasets do not include ground truth images for training and testing, we use their APS images as ground truth for training purposes. However, the APS image itself suffers from motion blur and low dynamic range. Thus, using APS images might not be the best way for training and also for evaluating the results. For that reason, we prepare the training APS images as described in Sec. 3.3, and assess the results using the structure similarity (SSIM) [30], feature similarity (FSIM) [32] computed by comparing the results with APS images, as well as by using the no-reference quality measure. In order to reach a holistic measure of quality, especially when evaluating the quality of reconstruction of real datasets without ground truth, the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [17], which utilizes normalized luminance coefficients to quantify the *naturalness* in images, is applied.

On the other hand, to assess the similarity between ground truth and generated images for synthetic datasets created using ESIM [23], each ground truth is matched with the corresponding reconstructed image with the closest timestamp, as mentioned in [27]. The SSIM, FSIM, and the peak signal-to-noise ratio (PSNR) are adopted to evaluate non-HDR scenes and scenes that we have reliable ground truth.

4.1. SBT versus SBE

We compare two event stacking methods, SBT and SBE, using our real datasets. 17K event stack-APS image pairs are used for training, where we set Δt for SBT to 0.03s and the number of events in one stack to 60K for SBE. To clearly see the effect of a stacking method, the number of frames (n) in one stack is set to 3 for both methods.

Figure 5 shows reconstructed images on our real-world datasets using SBE and SBT, respectively, for qualitative comparison. It is shown that our methods (both SBT and SBE) are robust enough to reconstruct the images on different sequences, and the generated images are quite close to APS images considered as *ground truth*. Our methods could successfully reconstruct shapes, appearance of human, building, etc. When comparing SBT and SBE, SBE produces better results in general. Table 1 shows quantitative evaluation results of using SBE. Note that large SSIM and FSIM values in Table 1 do not always mean the better output quality because they just present the similarity with APS images suffering from motion blur and low dynamic range.

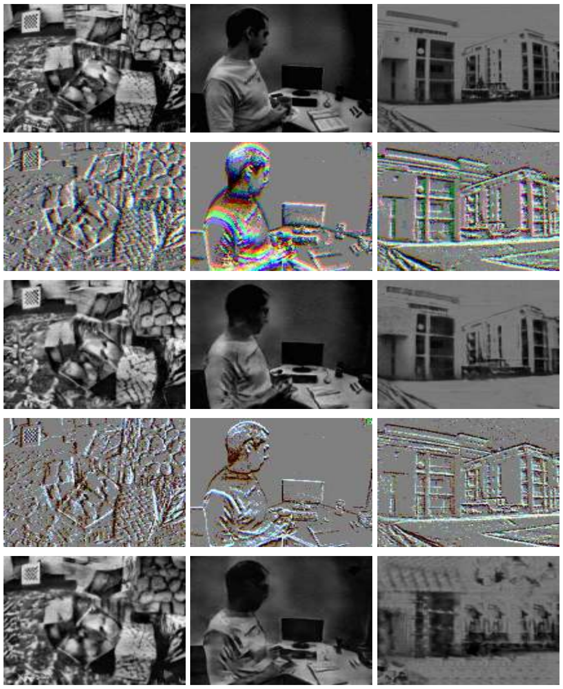


Figure 5. Reconstruction results using input event stacks (visualized as pseudo color images) on different real-world sequences [20]. From top to bottom, APS images as ground truth, event stacks using SBE, reconstructed images with SBE, event stacks using SBT, and reconstructed images with SBT.

Table 1. Quantitative evaluation of SBE on real-world datasets.

	BRISQUE	FSIM	SSIM
Ours(n = 3)	37.79±5.86	0.85±0.05	0.73±0.16

4.2. Quantitative evaluation with simulated datasets

In Section 4.1, we investigate the potential of our method on real-world data which indicate that SBE is more robust than SBT. Therefore, we conduct experiments based on SBE and show the robustness of our methods on datasets from ESIM [23], which can generate large amount of reliable event data. Since the simulator produces noise-free APS images with corresponding events for a given image, APS images can be regarded as ground truth, leading to evaluate the results quantitatively. In addition, although our method is capable of stacking, namely, any number of frames (n) into a stack, we choose the number of channels $n = \{1, 3\}$ to examine the effect of different numbers of channels. The number of events in one stack is set to 60K.

Table 2 shows the quantitative evaluation of our method with $n = 1$ and $n = 3$. It is shown that our method with $n = 3$ produces better results than with $n = 1$, proving that

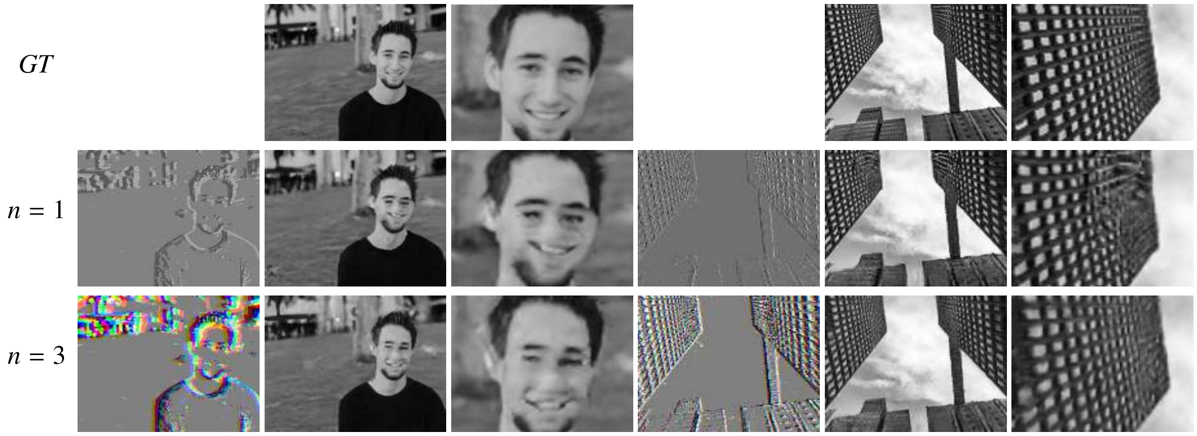


Figure 6. Reconstructed outputs from the inputs generated by ESIM [23]. Using 3 frames per stack ($n = 3$) results in a more robust reconstructions in comparison to one-frame stack in which images are distorted due to over-accumulated events.

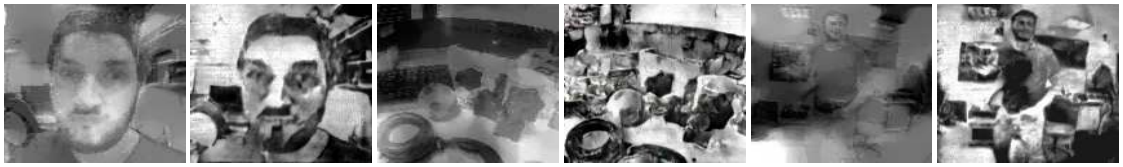


Figure 7. Comparing our method (image 2, 4, and 6 from left) to Reinbacher *et al.* [25] (image 1, 3, and 5) on the dataset of [2]. Our method produces more details (e.g. face, beard, jumping pose, etc) as well as more natural gray variations in less textured areas.

Table 2. Experiments on ESIM (simulator) datasets. Having more frames in one stack yields better results.

	PSNR (dB)	FSIM	SSIM
Ours($n = 1$)	20.51±2.86	0.81±0.09	0.67±0.20
Ours($n = 3$)	**24.87**±3.15	**0.87**±0.06	**0.79**±0.12

having more frames in one stack really improves the performance since it can preserve more temporal information as mentioned in Sec. 3.1. In Fig. 6, we show a few reconstructed images as well as input event stacks and ground truth images. One thing needs to mention is that the face reconstructed with $n = 1$ and the top of the building are a little bit distorted, which may be induced by too many events accumulated in one single channel. Further challenging scenes togtether with the GT are presented in Fig. 11.

4.3. Comparison to relevant works

We also qualitatively compare our methods on the sequences (*face, jumping, and ball*) with the results of manifold regularization (MR) [21] and intensity estimation (IE) [2] in Fig. 7. Since we deal with highly dynamic data, we provide more persuasive and explicit explanation and results in the supplementary video, which shows the whole sequence of several hundred of frames.

To compare the performance quantitatively, we use the BRISQUE score because no ground-truth image is available for these sequences. We compare the outputs of our method (SBE, $n = 3$) on sequences (*face, jumping, and ball*) to the results of MR [21] and IE [2] in Table 3. The results are quite consistent to the visual impression of Fig. 7. Our outputs on all face, jumping, and ball sequences show much more details and result in relatively higher BRISQUE score.

Table 3. Quantitative comparison of our method to the methods in [2] and [21]. The reported numbers are the mean and standard deviation of the BRISQUE measure applied to all reconstructed frames of the sequences. Our method shows better BRISQUE scores for all sequences.

Sequence	Face	Jumping	Ball
Bardow [2]	22.27±8.81	29.39±7.27	29.37±9.61
Munda [21]	27.29±7.27	48.18±6.70	34.98±9.31
Ours($n = 3$)	**48.26**±3.14	**48.34**±2.18	**39.18**±3.49

5. Discussion

Although creating intensity images from an event stream itself is challenging, the resultant images can also be used for other vision tasks such as object recognition, tracking, 3D reconstruction, SLAM etc. In that sense, the proposed method can be applied to many applications that use event cameras. Here, since the proposed method can fully exploit the advantages of events cameras such as high temporal resolution and high dynamic range, it can generate HDR images even better than APS images and very high frame rate videos as mentioned in Sec. 3.1.3, greatly increasing the usefulness of the proposed method.

Events to HDR images: In this paper, it is clearly shown that event stacks have rich information for HDR image reconstruction. In many cases, some parts of the scene are

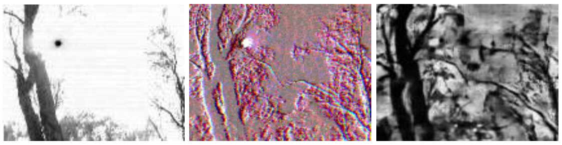


Figure 8. HDR imaging against direct sunlight (extreme illumination). Left to right: APS, event stack, our reconstruction result. (sequence from [27]).

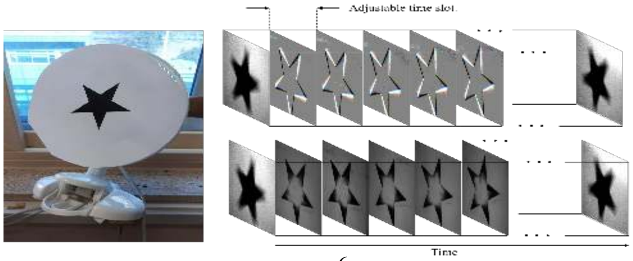


Figure 9. High frame rate (up to 10^6 FPS) video reconstruction.

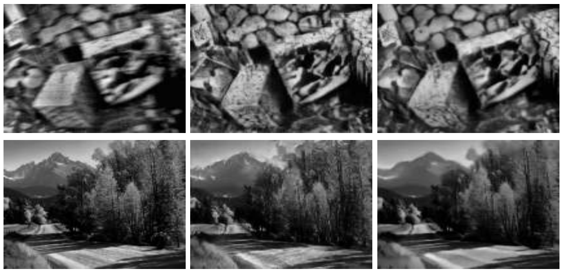
not visible in the APS image because of its low dynamic range. But many events really exist in those regions in the event camera as in the region under the table in Fig. 11 or the checkerboard pattern at the top left part of the stacked image in Fig. 2. Although both examples are from dark illumination but normal cameras also fail in rather bright illumination. Figure 8 shows the ability of the proposed method for HDR image generation in such cases.

Events to high frame rate videos: The motion blur due to fast motion of a camera or the scene is one of the challenging problems, and this makes the vision methods unreliable. However, our method can actually generate very high frame rate (HFR) videos with much less motion blur under the fast motion as mentioned in Sec. 3.1.3. To prove this ability, we conducted the tracking experiments using the reconstructed HFR video: with the event-based high frame rate video reconstruction framework, we can recover clear motion of a star-shape object attached on a fan with rotation speed of 13000 RPM, and the result in Fig. 9 shows that it is capable of generating the video up to 1 million fps.

Effect of the loss function: We also conduct ablation studies on different combination of the loss terms. The results are shown in Table 4 and Fig. 10. In terms of PSNR, the $L1$ norm reaches higher values, while we use cGAN + $L1$ throughout our experiments since it reflected a higher BRISQUE score in the simulated inputs. Higher PSNR does not always mean the better output quality because it just presents the similarity with APS images (used as GT) suffering from noise, motion blur and, low dynamic range. For example, higher PSNR means that the result with $L1$ in Fig. 10 is more similar to the low-quality APS image. Since we want to reconstruct images more realistic and better than APS images (used as GT), we do not use $L1$ but use cGAN + $L1$. Moreover, the $L1$ norm by itself blurs the image and averages out fine details.

Table 4. Effect of GAN, CGAN, and standard $L1$ loss function on real world (R) and simulated (S) inputs.

	cGAN+$L1$	cGAN	$L1$	GAN+$L1$	GAN
PSNR (S)	24.82	22.91	28.59	25.13	8.09
BRISQ. (S)	40.7	39.2	39.7	40.3	39.7
SSIM (S)	0.809	0.729	0.897	0.823	0.120
PSNR (R)	20.36	18.51	21.34	19.78	13.71
BRISQ. (R)	35.06	33.37	39.47	36.20	36.53
SSIM (R)	0.587	0.543	0.670	0.568	0.271



APS ↑, GT↓ cGAN + $L1$ $L1$

Figure 10. Results with different loss functions (real↑, simulated↓)

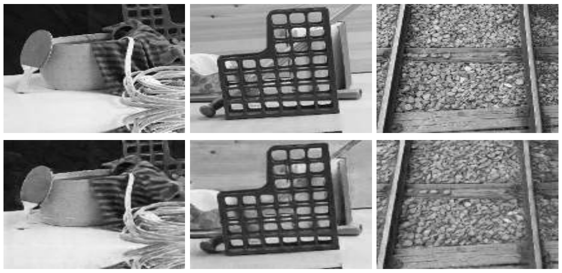


Figure 11. Our results(↓) with simulated events from GT images(↑)

6. Conclusion

We demonstrated how our cGANs-based approach can benefit from the properties of event cameras to accurately reconstruct HDR non-blurred intensity images and high frame rate videos from pure events. We first proposed two initiative event stacking methods (SBT and SBE) for both image and video reconstruction from events using the network. We then showed the advantages of using event cameras to generate high dynamic range images and high frame rate rate videos through experiments based on our datasets made of online available real-world sequences and simulator. In order to show the robustness of our method, we compared our cGANs-based event-to-image framework with other existing reconstruction methods and showed that our method outperforms other methods on public available datasets. We also showed it is possible to generate high dynamic range images even in extreme illumination conditions and also non-blurred images under rapid motion.

Acknowledgement

This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2018R1A2B3008640).

References

[1] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 18, page 1, 2018. 2

[2] P. Bardow, A. J. Davison, and S. Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 884–892, 2016. 1, 2, 6, 7

[3] S. Barua, Y. Miyatani, and A. Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. 2

[4] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck. Ddd17: End-to-end davis driving dataset. *arXiv preprint arXiv:1711.01458*, 2017. 2

[5] N. F. Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under egomotion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 644–653, 2018. 2

[6] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger. Interacting maps for fast visual interpretation. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 770–776. IEEE, 2011. 2

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 2, 3, 4, 5

[9] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016. 2

[10] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43:566–576, 2008. 2

[11] H. Kim, S. Leutenegger, and A. J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016. 1

[12] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017. 2

[13] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016. 2, 4

[14] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008. 1

[15] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garcıa, and D. Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018. 2

[16] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 2

[17] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. 6

[18] D. P. Moeys, F. Corradi, E. Kerr, P. Vance, G. Das, D. Neil, D. Kerr, and T. Delbrück. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on*, pages 1–8. IEEE, 2016. 2

[19] D. P. Moeys, C. Li, J. N. Martel, S. Bamford, L. Longinotti, V. Motsnyi, D. S. S. Bello, and T. Delbruck. Color temporal contrast sensitivity in dynamic vision sensors. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, 2017. 2

[20] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. 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. 2, 4, 5, 6

[21] G. Munda, C. Reinbacher, and T. Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, 126(12):1381–1393, 2018. 7

[22] A. Nguyen, T.-T. Do, D. G. Caldwell, and N. G. Tsagarakis. Real-time 6dof pose relocalization for event cameras with stacked spatial lstm networks. *arXiv preprint*. 1, 2

[23] H. Rebecq, D. Gehrig, and D. Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982, 2018. 2, 5, 6, 7

[24] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2017. 1

[25] C. Reinbacher, G. Graber, and T. Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *arXiv preprint arXiv:1607.06283*, 2016. 1, 2, 4, 7

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[27] C. Scheerlinck, N. Barnes, and R. Mahony. Continuous-time intensity estimation using event cameras. *arXiv preprint arXiv:1811.00386*, 2018. 6, 8

[28] P. A. Shedligeri, K. Shah, D. Kumar, and K. Mitra. Photorealistic image reconstruction from hybrid intensity and event based sensor. *arXiv preprint arXiv:1805.06140*, 2018. 2

[29] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016. 2

[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[31] Z. Yi, H. R. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2868–2876, 2017. 4, 5

[32] L. Zhang, L. Zhang, X. Mou, D. Zhang, et al. Fsim: a feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. 6

[33] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Evflownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018. 2

[34] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 2, 3