Learn to See by Events:
Color Frame Synthesis from Event and RGB Cameras

Stefano Pini[1] [a], Guido Borghi[2] [b] and Roberto Vezzani[1,2] [c]

[1]*DIEF - Dipartimento di Ingegneria "Enzo Ferrari", University of Modena and Reggio Emilia, Italy*
[2]*AIRI - Artificial Intelligence Research and Innovation Center, University of Modena and Reggio Emilia, Italy*

Keywords: Event Cameras, Event Frames, Simulated Event Frames, Color Frame Synthesis, Automotive.

Abstract: Event cameras are biologically-inspired sensors that gather the temporal evolution of the scene. They capture pixel-wise brightness variations and output a corresponding stream of asynchronous events. Despite having multiple advantages with respect to traditional cameras, their use is partially prevented by the limited applicability of traditional data processing and vision algorithms. To this aim, we present a framework which exploits the output stream of event cameras to synthesize RGB frames, relying on an initial or a periodic set of color key-frames and the sequence of intermediate events. Differently from existing work, we propose a deep learning-based frame synthesis method, consisting of an adversarial architecture combined with a recurrent module. Qualitative results and quantitative per-pixel, perceptual, and semantic evaluation on four public datasets confirm the quality of the synthesized images.

1 INTRODUCTION

Event cameras are neuromorphic optical sensors capable of asynchronously capturing pixel-wise brightness variations, *i.e. events*. They are gaining more and more attention from the computer vision community thanks to their extremely high temporal resolution, low power consumption, reduced data rate, and high dynamic range (Gallego et al., 2018b).

Moreover, event cameras filter out redundant information as their output intrinsically embodies only the temporal dynamics of the recorded scene, ignoring static and non-moving areas. On the other hand, standard intensity cameras with an equivalent frame rate are able to acquire the whole complexity of the scene, including textures and colors. However, they usually require a huge amount of memory to store the collected data, along with a high power consumption and a low dynamic range (Maqueda et al., 2018).

Given the availability of many mature computer vision algorithms for standard images, being able to apply them on event data, without the need of designing specific algorithms or collecting new datasets, could contribute to the spread of event sensors.

Differently from existing works, which are built on filter- and optimization-based algorithms (Brandli et al., 2014b; Munda et al., 2018; Scheerlinck et al., 2018), in this paper we investigate the use of deep learning-based approaches to interpolate frames from a low frame rate RGB camera using event data.

In particular, we propose a model that synthesizes color or gray-level frames preserving high-quality textures and details (Fig. 1) thus allowing the use of traditional vision algorithms like object detection and semantic segmentation networks.

We explore the use of a conditional adversarial network (Mirza and Osindero, 2014) in conjunction with a recurrent module to estimate RGB frames, relying on an initial or a periodic set of color key-frames and a sequence of event frames, *i.e.* frames that collect events occurred in a certain amount of time.

Moreover, we propose to use simulated event data, obtained by means of image differences, to train our model: this solution leads to two significant advantages. First, event-based methods can be evaluated on standard datasets with annotations, which are often not available in the event domain. Second, learned models can be trained on simulated event data and used with real event data, unseen during the training procedure. In general, we propose to shift from event-based context to a domain where more expertise is available in terms of mature vision algorithms.

[a] https://orcid.org/0000-0002-9821-2014
[b] https://orcid.org/0000-0003-2441-7524
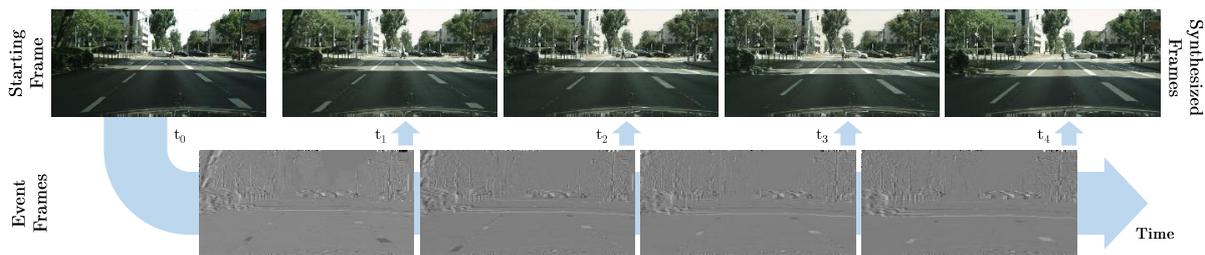[c] https://orcid.org/0000-0002-1046-6870

37

Figure 1: Sample frames synthesized by the proposed framework. Given an initial RGB frame at time t_0 and a set of following event frames at time $t_1, ..., t_n$ as input, the proposed framework accordingly synthesizes an RGB frame for each time step.

As a case study, we embrace the automotive context, in which event and intensity cameras could cover a variety of applications (Borghi et al., 2019; Frigieri et al., 2017). For instance, the growing number of high-quality cameras placed on recent cars implies the use of a large bandwidth in the internal and the external network: sending only key-frames and events might be a way to reduce the bandwidth requirements, still maintaining a high temporal resolution.

We probe the feasibility of the proposed model testing it on four automotive and publicly-released datasets, namely *DDD17* (Binas et al., 2017), *MVSEC* (Zhu et al., 2018), *Kitti* (Geiger et al., 2013), and *Cityscapes* (Cordts et al., 2016).

Summarizing, our contributions are threefold: *i)* we propose a framework based on a conditional adversarial network that performs the synthesis of *color* or gray-level frames. *ii)* we investigate the use of simulated event frames to train systems able to work with real event data; *iii)* we probe the effectiveness of the proposed method employing four public automotive datasets, investigating the ability to generate *realistic* images, preserving colors, objects, and the semantic information of the scene.

2 RELATED WORK

Event-based vision has recently attracted the attention of the computer vision community. In the last years, event-based cameras, also known as neuromorphic sensors or *Dynamic Vision Sensors* (DVSs) (Lichtsteiner et al., 2006), have been mainly explored for monocular (Rebecq et al., 2016) and stereo depth estimation (Andreopoulos et al., 2018; Zhou et al., 2018), optical flow prediction (Gallego et al., 2018b) as well as for real time feature detection and tracking (Ramesh et al., 2018; Mitrokhin et al., 2018) and ego-motion estimation (Maqueda et al., 2018; Gallego et al., 2018a). Moreover, various classification tasks were addressed employing event-based data, as classification of faces (Lagorce et al., 2017) and gestures (Lungu et al., 2017).

Recently, a limited amount of work focused on the reconstruction of intensity images or videos from event cameras. Bardow (Bardow et al., 2016) proposed an approach to simultaneously estimate the brightness and the the optical flow of the recorded scene: the optical flow was shown to be necessary to correctly recover sharp edges, especially in presence of fast camera movements.

In (Reinbacher et al., 2016) and its extended version (Munda et al., 2018), a manifold regularization method was used to reconstruct intensity images. However, predicted images exhibit significant visual artifacts and a relatively high noise. (Munda et al., 2018) and the method proposed by (Kim et al., 2014) show best visual results under limited camera or subject movements.

Brandli (Brandli et al., 2014b) investigated the video decompression task and proposed an online event-based method that relies on an optimization algorithm. The synthesized image is reset with every new frame to limit the growth of the integration error. In (Scheerlinck et al., 2018), a continuous-time intensity estimation using event data is introduced. This method is based on a complementary filter, which is able to exploit both intensity frames and asynchronous events to output the gray-level image. We point out that, as highlighted in (Scheerlinck et al., 2018), optimizazion- and filter-based methods imply the tuning of several parameters (such as the contrast threshold and the event-rate) for each recording scenario. This could limit the usability and the generalization capabilities of those methods. In fact, in (Scheerlinck et al., 2018) these parameters are tuned for each testing sequence in order to improve the intensity estimation.

Recently, in (Pini et al., 2019) an encoder-decoder architecture has been proposed to synthesize only gray-level frames starting from event data. The proposed approach is limited since neither an adversarial approach nor color frame information have been exploited to improve the final result.
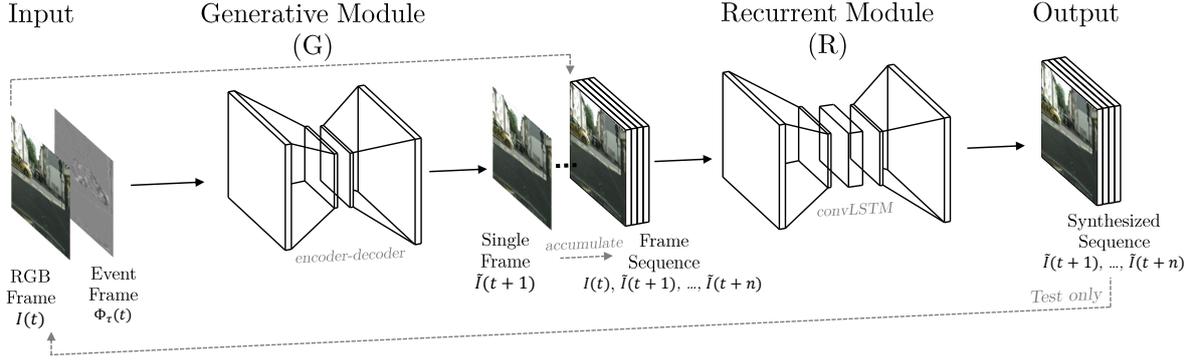
Figure 2: Overview of the proposed framework. The input of the *Generative Module* is the initial intensity frame $I(t)$ and the corresponding event frame $\Phi_\tau(t)$. $I(t)$ and the generated frames $\tilde{I}(t_i)$ are then stacked and used as input for the *Recurrent Module* that learns how to refine contextual details and maintain the temporal coherence between consecutive frames. At testing time, the output of the *Recurrent Module* is used in the next forward step as the new input frame of the framework.

In the method proposed by (Rebecq et al., 2019), a deep learning-based architecture is presented in order to reconstruct gray-level frames directly from event data, represented as a continuous stream of events along the acquisition time. The use of raw event data makes this method difficult to compare to the ours.

3 MATHEMATICAL FORMULATION

In this section, we present definitions and mathematical notations for events and event frames, followed by their relation to intensity images and the formulation of the proposed task, *i.e.* the intensity frame synthesis.

3.1 Event Frames

Following the notation proposed in (Maqueda et al., 2018), the k-th event e_k captured by an event camera can be represented as:

$$e_k = (x_k, y_k, t_k, p_k) \qquad (1)$$

where x_k, y_k, and t_k are the spatio-temporal coordinates of a brightness change and $p_k \in \{-1, +1\}$ specifies the polarity of this change, which can be either positive or negative.

By summing up all events captured in a time interval $\Delta t = [t, t+\tau]$ at a pixel-wise level, an event frame $\Phi_\tau(t)$ is obtained, integrating all the events occurred in that time interval. Formally, an event frame can be defined as:

$$\Phi_\tau(t) = \sum_{e_k \in E_{t,\tau}} p_k \qquad (2)$$

where $E_{t,\tau} = \{e_k \,|\, t_k \in [t, t+\tau]\}$. Therefore, an event frame could be represented as a gray-level image of size $1 \times w \times h$, which summarizes all events occurred in a certain time interval in a single channel. For numerical reasons, $\Phi_\tau(t)$ saturates if the amount of events exceeds the number of gray levels used to represent the event frame image.

3.2 Intensity Frame Synthesis

The core of the proposed approach consists in learning a parametric function

$$\Gamma: \mathbb{R}^{c \times w \times h} \times \mathbb{R}^{1 \times w \times h} \to \mathbb{R}^{c \times w \times h} \qquad (3)$$

that takes as input a c-channel intensity image $I_t \in \mathbb{R}^{c \times w \times h}$ captured at time t and an event frame $\Phi_\tau(t) \in \mathbb{R}^{1 \times w \times h}$, which combines pixel-level brightness changes between times t and $t+\tau$, and outputs the predicted intensity image $\tilde{I}(t+\tau) \in \mathbb{R}^{c \times w \times h}$ at time $t+\tau$. Here, w and h represent the width and the height of both intensity images and event frames.
It follows that

$$\tilde{I}(t+\tau) = \Gamma(I(t), \Phi_\tau(t), \theta) \qquad (4)$$

where θ corresponds to the parameters of the function Γ, that we define as the combination of multiple parametric functions (Sec. 4).

3.3 Difference of Images as Event Frames

Event cameras are naturally triggered by pixel-level logarithmic brightness changes and thus they provide some output data only if there is a relative movement between the sensor and the objects in the scene or a brightness change occurs (Gehrig et al., 2018). For small time intervals, *i.e.* small values of τ, the brightness variation can be approximated with a first-order
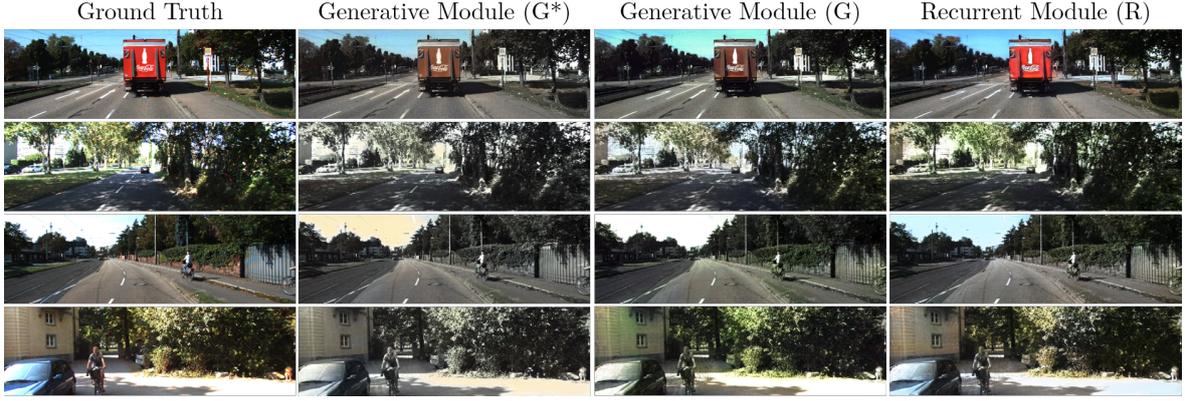
Figure 3: Sample output frames from *Kitti* dataset (Geiger et al., 2013). The ground truth is placed on the first column, then the output of the *Generative Module* without (G*) and with (G) the discriminator (cfr. Section 4.1) and finally the output of the *Recurrent Module* (R), that is able to preserve more realistic colors, enhance contrast and reduce visual artifacts.

Taylor's approximation as:

$$\lim_{\tau \to 0} \frac{\delta L}{\delta t} \tau \approx L(t+\tau) - L(t) \doteq \Delta L \qquad (5)$$

where $L(t) = \log(Br(I(t)))$, $I(t)$ is the image acquired at time t and $Br(\cdot)$ is a function to convert a c-channel image into the corresponding single-channel brightness. In the experiments, for instance, RGB images are converted into brightness images using the standard channel weights defined as $[0.299, 0.587, 0.114]$.

Therefore, an event frame $\Phi_\tau(t)$ can be approximated as follows:

$$\Phi_\tau(t) \approx \Delta L = \log[Br(I(t+\tau))] - \log[Br(I(t))] \quad (6)$$

Thanks to this assumption, given two intensity frames $I(t)$ and $I(t+\tau)$, it is possible to retrieve the corresponding event frame $\Phi_\tau(t)$ for small values of τ. However, since intensity frames have more than one channel (*e.g.* , three channels for RGB images), $I(t+\tau)$ cannot be analytically obtained given $I(t)$ and $\Phi_\tau(t)$.

4 IMPLEMENTATION

An overview of the proposed architecture is depicted in Figure 2. The framework integrates two main components. The first one – the *Generative Module* (G) – receives an intensity image $I(t)$ and an event frame $\Phi_\tau(t)$ as input and synthesizes the frame $I(t+\tau)$ as output.

The second one – the *Recurrent Module* (R) – refines the output of the Generative component, relying on the temporal coherence of a sequence of frames.

4.1 Generative Module

We follow the conditional GAN paradigm (Mirza and Osindero, 2014; Isola et al., 2017; Borghi et al., 2018) for the designing of the *Generative Module*.

The module consists of a generative network G and a discriminative network D (Goodfellow et al., 2014; Mirza and Osindero, 2014). Exploiting the *U-Net* architecture (Ronneberger et al., 2015), G is defined as a fully-convolutional deep neural network with skip connections between layers i and $n-i$, where n is the total number of layers. The discriminative network proposed by (Isola et al., 2017) is employed as D.

In formal terms, G corresponds to an estimation function that predicts the intensity frame $\tilde{I}(t+\tau) = G(I(t) \oplus \Phi_\tau(t))$ from the concatenation of an intensity frame and an event frame at time t (cfr. Equation 4) while D corresponds to a discriminative function able to distinguish between real and generated frames.

The training procedure can be formalized as the optimization of the following min-max problem:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{x \sim p(x), y \sim p(y)}[\log D(x,y)] \\ + \mathbb{E}_{x \sim p(x)}[\log(1 - D(x, G(x)))] \qquad (7)$$

where $D(x,y)$ is the probability of being a *real* frame and $1 - D(x, G(x))$ is the probability to be a synthesized frame, $p(x)$ is the distribution of concatenated frames $I(t) \oplus \Phi_\tau(t)$, and $p(y)$ is the distribution of frames $\tilde{I}(t+\tau)$.

This approach leads to a *Generative Module* which is capable of translating pixel intensities accordingly to an event frame and producing output frames that are visually similar to the real ones.

Table 1: Comparison between the proposed method and state of the art approaches using per-pixel and perceptual metrics.

Dataset	Method	Norm ↓		RMSE ↓			Threshold ↑			Indexes ↑		Perceptual ↓
		L_1	L_2	Lin	Log	Scl	1.25	1.25^2	1.25^3	PSNR	SSIM	LPIPS
DDD17	Munda et al.	0.268	94.277	0.314	5.674	5.142	0.152	0.448	0.536	10.244	0.216	0.637
	Scheerlinck et al.	0.080	29.249	0.098	4.830	4.352	0.671	0.781	0.827	20.542	0.702	0.208
	Pini et al.	0.027	8.916	0.040	4.048	3.571	0.775	0.848	0.875	29.176	0.864	**0.105**
	Ours	**0.022**	**8.583**	**0.039**	**3.766**	**3.408**	**0.787**	**0.855**	**0.880**	**29.428**	**0.884**	0.107
MVSEC	Munda et al.	0.160	86.419	0.288	8.985	8.016	0.088	0.163	0.232	11.034	0.181	0.599
	Scheerlinck et al.	0.067	26.794	0.089	7.313	6.982	0.263	0.357	0.467	21.070	0.551	0.257
	Pini et al.	0.026	12.062	0.054	**6.443**	6.102	**0.525**	**0.642**	**0.708**	25.866	0.740	0.172
	Ours	**0.022**	**11.216**	**0.051**	6.559	**6.003**	0.514	0.637	0.699	**26.366**	**0.845**	**0.137**

4.2 Recurrent Module

The architecture of the *Recurrent Module* is a combination of an *encoder-decoder* architecture and a *Convolutional LSTM* (ConvLSTM) module (Xingjian et al., 2015). The underlying idea is that while the *Generative Module* learns how to successfully combine intensity and event frames, the *Recurrent Module*, capturing the context of the scene and its temporal evolution, learns to visually refine the synthesized frames, removing artifacts, enhancing colors, and improving the temporal coherence.

We adopt the same *U-Net* architecture of the *Generative Module* and we insert a 512-channel two-layer ConvLSTM block in the middle of the hourglass model. During the training phase, the *Recurrent Module* receives as input a sequence of frames produced by the *Generative Module* and outputs a sequence of the same length, sequentially updating the internal state. The activation of each ConvLSTM layer can be defined as follows:

$$I_s = \sigma(W_i * X_s + U_i * H_{s-1} + b_i) \qquad (8)$$

$$F_s = \sigma(W_f * X_s + U_f * H_{s-1} + b_f) \qquad (9)$$

$$O_s = \sigma(W_o * X_s + U_o * H_{s-1} + b_o) \qquad (10)$$

$$G_s = tanh(W_c * X_s + U_c * H_{s-1} + b_c) \qquad (11)$$

$$C_s = F_s \odot C_{s-1} + I_s \odot G_s \qquad (12)$$

$$H_s = O_s \odot tanh(C_s) \qquad (13)$$

where, I_s, F_s, O_s are the gates, C_s, C_{s-1} are the memory cells, G_s is the candidate memory, and H_s, H_{s-1} are the hidden states. Each b is a learned bias, each W and U are a learned convolutional kernel, and X_s corresponds to the input. Finally, $*$ represents the convolutional operator while $\odot$ is the element-wise product.

4.3 Training Procedure

The framework is trained in two consecutive steps.

In the first phase, the *Generative Module G* is trained following the adversarial approach detailed in Section 4.1. We optimize the network using

Adam (Kingma and Ba, 2014) with learning rate 0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$, and a batch size of 8. In order to improve the stability of the training process, the discriminator is updated every 8 training steps of the generator. The objective function of *D* is the common binary categorical cross entropy loss, while the objective function of *G* is a weighted combination of the adversarial loss (*i.e.* the binary crossentropy) and the *Mean Squared Error* (MSE) loss.

In the second phase, the *Recurrent Module R* is trained while keeping the parameters of the *Generative Module* fixed. We apply the Adam optimizer with the same hyper-parameters we used for the *Generative Module*, with the exception of the batch size which is set to 4. The objective function of the module is a weighted combination of the MSE loss and the *Structural Similarity* index (SSIM) loss which is defined as:

$$\text{SSIM}(p,q) = \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)} \qquad (14)$$

Given two windows p, q of equal size, $\mu_{p,q}$, $\sigma_{p,q}$ are the mean and variance of p, q while $c_{1,2}$ are used to stabilize the division. See (Wang et al., 2004) for further details. The losses are combined with a weight of 0.5 each. The network is trained with a fixed sequence length, which corresponds to the length of the sequences used during the evaluation phase.

Only during the testing phase, to obtain a sequence of synthesized frames, the framework receives as input the previously-generated images or an intensity key-frame.

5 FRAMEWORK EVALUATION

In this section, we present the datasets that we used to train and test the proposed framework. Then, we describe the evaluation procedure that has been employed to assess the quality of the synthesized frames, followed by the report of the experimental results and their analysis.

Table 2: Experimental results of pixel-wise metrics computed on synthesized frames from *DDD17*, *MVSEC*, *Kitti*, and *Cityscapes* (CS) datasets. Details on adopted metrics are reported in Section 5.2. Tests are carried out employing the *Generative Module* (G), and both the *Generative* and *Recurrent Module* (G+R).

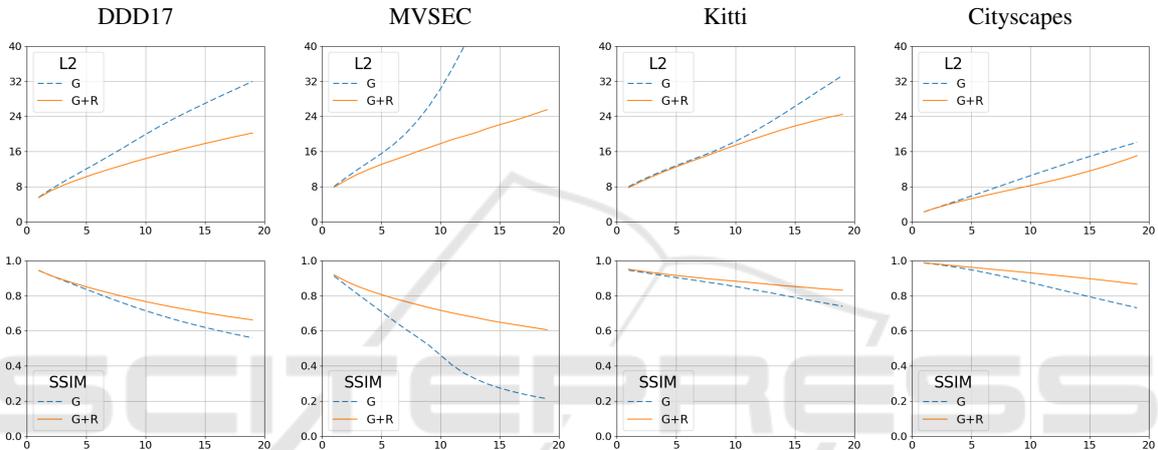| Dataset	Model	Norm ↓		Difference ↓		RMSE ↓			Threshold ↑			Indexes↑	
		L_1	L_2	Abs	Sqr	Lin	Log	Scl	1.25	1.25^2	1.25^3	PSNR	SSIM
DDD17	G	0.029	9.658	0.114	0.007	0.044	2.296	2.268	0.854	0.919	0.941	28.486	0.876
	G+R	0.022	8.583	0.167	0.006	0.039	3.766	3.408	0.787	0.855	0.880	29.428	0.884
MVSEC	G	0.026	12.830	0.311	0.013	0.058	6.302	6.233	0.562	0.675	0.733	25.309	0.784
	G+R	0.022	11.216	0.354	0.010	0.051	6.559	6.003	0.514	0.637	0.699	26.366	0.845
Kitti	G	0.030	10.95	0.125	0.006	0.048	0.472	0.463	0.782	0.940	0.981	27.140	0.919
	G+R	0.029	10.71	0.105	0.005	0.046	0.194	0.191	0.846	0.968	0.991	27.295	0.928
CS	G	0.019	4.534	0.086	0.003	0.025	0.232	0.211	0.877	0.974	0.992	32.769	0.962
	G+R	0.015	4.192	0.059	0.002	0.023	0.172	0.170	0.968	0.997	0.999	33.315	0.971

Figure 4: Variation of L_2 and *SSIM* as a function of the *i*-th synthesized frame since the last key-frame by the *Generative Module*, without (blue) and with (orange) the discriminator, and the *Recurrent Module* (green), computed on *DDD17*, *MVSEC*, *Kitti*, and *Cityscapes*. The horizontal axis refers to the frame on which the metric is calculated, starting from an initial color frame and estimating the following ones.

5.1 Datasets

Due to the recent commercial release of event cameras, only few event-based datasets are currently publicly-released and available in the literature. These datasets still lack the data variety and the annotation quality which is common for RGB datasets. These considerations have motivated us to exploit the mathematical intuitions presented in Section 3.3 in order to take advantage of non-event public automotive datasets (Geiger et al., 2013; Cordts et al., 2016), which are richer in terms of annotations and data quality, along with two recent event-based automotive datasets (Binas et al., 2017; Zhu et al., 2018).

DDD17. Binas *et al.* (Binas et al., 2017) introduced *DDD17: End-to-end DAVIS Driving Dataset*, which is the first open dataset of annotated event driving recordings. The dataset is captured by a *DAVIS* sensor (Brandli et al., 2014a) and includes both gray-level frames (346×260 px) and event data. Se-

quences are captured in urban and highway scenarios, during day and night and under different weather conditions. Similar to (Maqueda et al., 2018), experiments are carried out selecting only sequences labelled as `day`, `day wet`, and `day sunny`, but we create train, validation, and test split using different sequences.

MVSEC. The *Multi Vehicle Stereo Event Camera Dataset* (Zhu et al., 2018) contains data acquired from four different vehicles, in both indoor and outdoor environments, during day and night, using a pair of *DAVIS 346B* event cameras (346×260 px), a stereo camera, and a *Velodyne* lidar. In this paper, we use only the outdoor car scenes recorded during the day. From these, we select the first 70% as train set, and the following as validation (10%) and test (20%) set.

Kitti. The *Kitti Vision Benchmark Suite* was introduced in (Geiger et al., 2012). In this work, we use the *KITTI raw* (Geiger et al., 2013) subset, which includes 6 hours of 1242×375 rectified RGB image
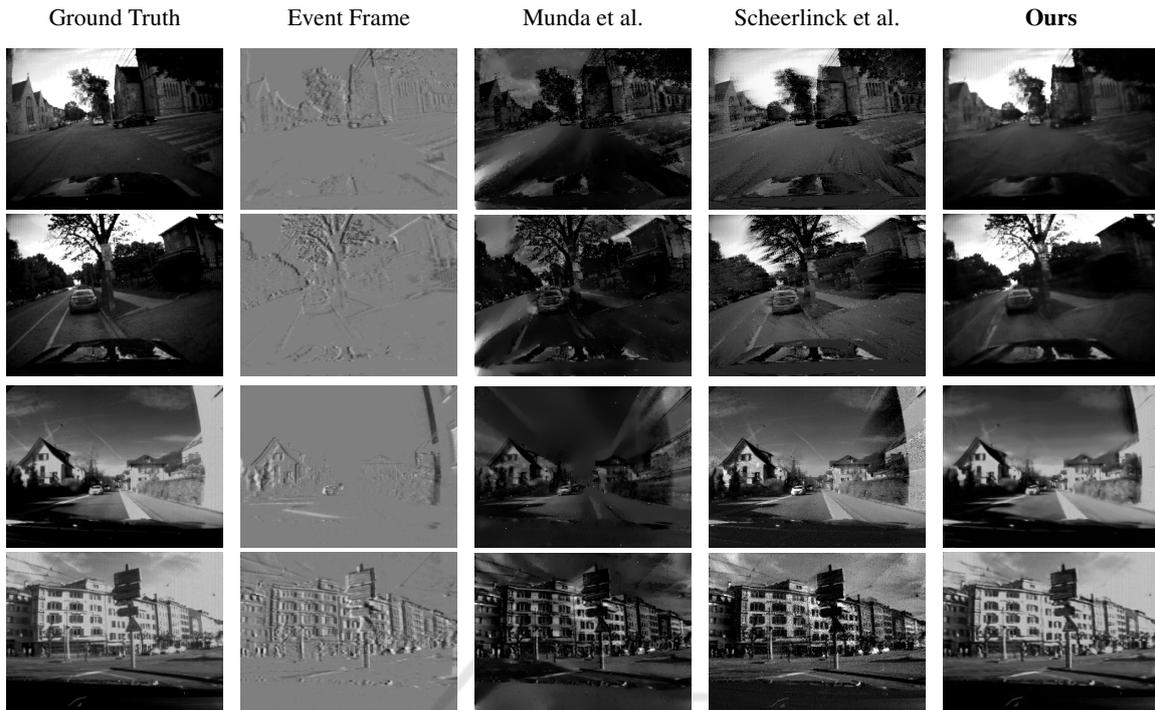
Figure 5: Output of the proposed algorithm compared to existing methods on MVSEC and DDD17 datasets.

sequences captured on different road scenarios with a temporal resolution of 10Hz. The dataset is rich of annotations, as depth maps and semantic segmentation. We adopt the train and the validation split proposed in (Uhrig et al., 2017) to respectively train the method and to validate and test it.

Cityscapes. The *Cityscapes* dataset (Cordts et al., 2016) consists of thousands of RGB frames with a high spatial resolution (2048 × 1024 px) and shows varying and complex scene layouts and backgrounds. Fine and coarse annotations of 30 different object classes are provided as both semantic and instance-wise segmentation. We select a particular subset, namely leftImg8bit_sequence, following official splits, in order to use sequences with a frame rate of 17Hz and to have access to fine semantic segmentation annotations.

5.2 Metrics

Inspired by (Eigen et al., 2014; Isola et al., 2017), we exploited a variety of metrics to check the quality of the generated images, being aware that evaluating synthesized images is, in general, a still open problem (Salimans et al., 2016). We firstly design a set of experiments in order to investigate the contribution of each single module of the proposed framework and to compare it with state-of-art methods by using pixel-wise and perceptual metrics. Then, we exploit

off-the-shelf networks pre-trained on public datasets in order to evaluate semantic segmentation and object detection scores on generated images.

Pixel-wise and Perceptual Metrics. A collection of evaluation metrics is used to assess the quality of the synthesized images. In particular, we report the $L1$ and $L2$ distance, the root mean squared error (RMSE), and the percentage of pixel under a certain error threshold (δ-metrics). Moreover, we include the *Peak Signal-to-Noise Ratio* (PSNR), which estimates the level of noise in logarithmic scale, and the *Structural Similarity* (SSIM) (Wang et al., 2004), which measures the perceived closeness between two images. Finally, the visual quality of the generated images is assessed through the *Learned Perceptual Image Patch Similarity* (LPIPS) (Zhang et al., 2018) which was shown to correlate well with human judgement (Zhang et al., 2018).

Semantic Segmentation Score. We adopt a pre-trained semantic classifier to measure the accuracy of a certain set of pixels to be a particular class. Specifically, we rely on the validation set of the *Kitti* and *Cityscapes* dataset. If synthesized images are close to the real ones, the classifier will achieve a comparable accuracy to the one obtained on the reference dataset. We adopt the recent state-of-art WideResNet+38+DeepLab3 (Rota Bulò et al., 2018) trained on the original train annotations of the *Cityscapes* dataset. Since semantic fine annotations

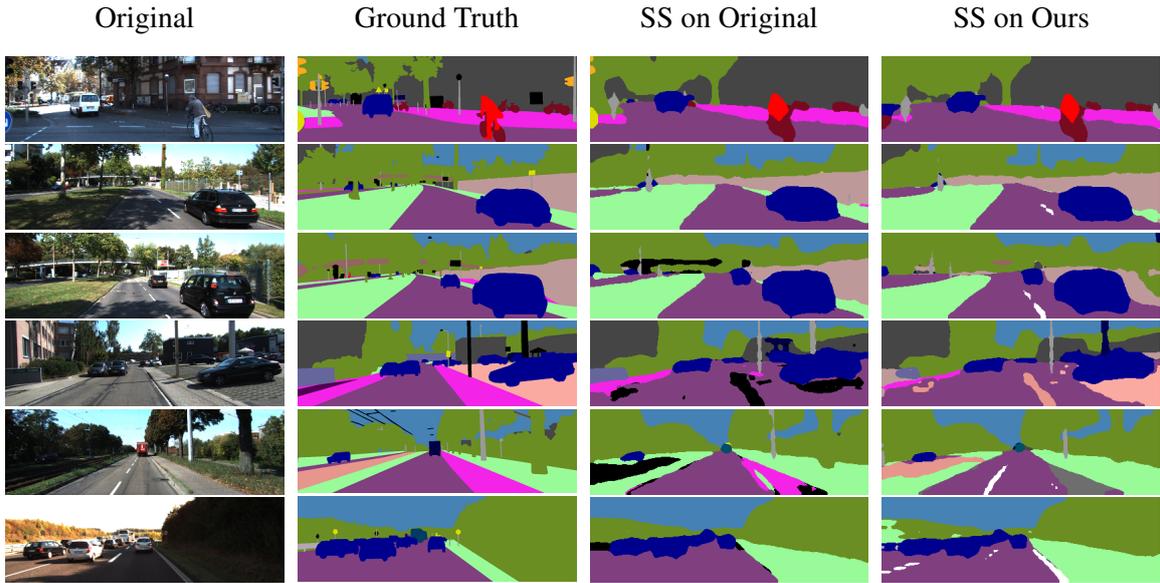| Original	Ground Truth	SS on Original	SS on Ours

Figure 6: Semantic Segmentation (SS) applied on *Kitti* dataset (Geiger et al., 2013). From left, we report the original RGB frame, the ground truth of the semantic segmentation and then the segmentation computed on original and synthesised frames.

are provided only for a limited subset of frames in each sequence, we compare these annotations with the semantic maps produced using as input the last frame of a synthesized sequence (*i.e.* the worst case).

Object Detection Score. A pre-trained object detector is employed in order to investigate if the proposed model is able to preserve details, locations, and realistic aspect of the objects that appear in the scene. We adopt the popular `Yolo-v3` network (Redmon and Farhadi, 2018), a real-time state-of-the-art object detection system, pre-trained on the *COCO* dataset (Lin et al., 2014). In this way, since we use automotive datasets, we investigate the ability of the proposed framework to preserve objects in the generated frames, in particular people, trucks, cars, buses, trains, and stop signals.

5.3 Experimental Results

For a fair comparison, we empirically set the same sequence length of 6 synthesized frames for every experiment and competitor method reported in this section. We split data following the training and testing sets of each dataset, and we use the validation set to stop the training procedure. When considering *DDD17* and *MVSEC*, only real event data are used while we obtain synthetic event frames on *Kitti* and *Cityscapes*. We adapt the image resolution of the original data to comply with the *U-Net* architecture (see Sec. 4.1) requirements while trying to keep the original image aspect ratio. Therefore, we adopt input images with a spatial resolution of 416×128 for *Kitti*,

Table 3: Semantic Segmentation and Object Detection scores computed on synthesized frames from *Kitti* and *Cityscapes*. Results are compared with the Ground Truth (GT), when available.

Data	Model	Semantic Segmentation ↑			Object Det. ↑	
		Per-pixel	Per-class	class IoU	mIoU	%
Kitti	G	0.814	0.261	0.215	0.914	65.8
	G+R	0.813	0.261	0.215	0.912	71.4
	GT	0.827	0.283	0.235	-	-
CS	G	0.771	0.197	0.162	0.924	83.5
	G+R	0.790	0.201	0.166	0.926	86.4
	GT	0.828	0.227	0.192	-	-

256×128 for *Cityscapes*, and 256×192 for *DDD17* and *MVSEC*.

In Table 1, per-pixel evaluation shows that our model overcomes all the competitors on both the event-based *DDD17* and *MVSEC* datasets. We point out that (Munda et al., 2018) is based only on event data while the input of (Scheerlinck et al., 2018; Pini et al., 2019) are both gray-level images and events, similarly to our method. According to Table 1, the visual results reported in Figure 5 confirm the superior quality of the images synthesized by our method and suggest that the proposed learning-based approach can be an alternative of filter-based algorithms. Indeed, visual artifacts (*e.g.* shadows) and high noise (*e.g.* salt and pepper) are visible in the competitor generated frames while our method produces more accurate brightness levels. In addition, competitors are limited to the gray-level domain only.

As an ablation study, we exploit the pixel-wise

Original   OD on original   OD on Ours

Figure 7: Object Detection (OD) on *Cityscapes* dataset (Cordts et al., 2016).

metrics also to understand the contribution of each single module of the proposed system. Table 2 and Figure 3 show that the output of the *Generative Module* has a good level of quality and learns efficiently to alter pixel values accordingly to event frames. *Recurrent Module* visually improves the output frames, enhancing the colors, the level of details, and the temporal coherence.

Generally, we note that the low quality of the gray-level images provided in *DDD17* and *MVSEC* partially influences the performance of the framework.

In Table 3, results are reported in terms of per-pixel, per-class, and *IoU* accuracy for the Semantic Segmentation score and in terms of mean *IoU* and percentage of the correctly detected objects for the Object Detection score.

Segmentation results confirm that our approach can be a valid option to avoid the development of completely new vision algorithms relying on event data. Also in this case, the *Recurrent Module* improves the final score (Fig. 6). The Object Detection scores are reported in Table 3 in terms of mean *IoU* on detection bounding boxes and the percentage of objects detected with respect to the `Yolo-v3` network detections on the ground truth images. Object detection scores

are interesting, since we note that even though the mean *IoU* computed is similar, the *Recurrent Module* allows to find a higher number of detections, suggesting that the synthesized frames are visually similar to the corresponding real ones, as shown in Figure 7. Through these tests, we verify the capability of the proposed framework to preserve objects and semantic information in the synthesized frames, which is mandatory for employing the proposed method in real-world automotive scenarios.

Finally, we conduct a cross-modality test: we train our model on *DDD17* and *MVSEC* datasets, using as event frame the logarithmic difference of two consecutive frames (*i.e.* simulated event frames). Then, we test the network using as input real event frames, without any fine-tuning procedure. On the *DDD17* dataset we obtain PSNR of 23.396 and SSIM of 0.779, and values of 21.935 and 0.736 on the *MVSEC* dataset. These results confirm the ability of the proposed system to deal with both simulated (during train) and real (during test) event data. Furthermore, it is proved that the logarithmic difference of grayscale images can be efficiently used in place of real event frames, introducing the possibility to use common RGB dataset and their annotations to simulate

the input of an event camera, in the form of event frames.

In Figure 4, we plot *L2* and *SSIM* values for each frame within a synthesized sequence, showing the performance drop with respect to the number of generated frames from the last key-frame. As expected, the contribution of the *Recurrent Module* increases along with the length of the sequence, confirming the effectiveness of the proposed model in the long-sequence generation task.

Our system implementation, tested on a *NVidia 1080Ti*, takes an average time of 47.6 ± 3.7ms to synthesize a single image, reaching a frame rate of about 20Hz.

6 CONCLUSION

In this paper, we propose a framework able to synthesize color frames, relying on an initial or a periodic set of key-frames and a sequence of event frames. The Generative Module produces an intermediate output, while the Recurrent Module refines it, preserving colors and enhancing the temporal coherence. The method is tested on four public automotive datasets, obtaining state-of-art results.

Moreover, semantic segmentation and object detection scores show the possibility to run traditional vision algorithms on synthesized frames, reducing the need of developing new algorithms or collecting new annotated datasets.

ACKNOWLEDGEMENTS

REFERENCES

Andreopoulos, A., Kashyap, H. J., Nayak, T. K., Amir, A., and Flickner, M. D. (2018). A low power, high throughput, fully event-based stereo system. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 7532–7542.

Bardow, P., Davison, A. J., and Leutenegger, S. (2016). Simultaneous optical flow and intensity estimation from an event camera. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 884–892.

Binas, J., Neil, D., Liu, S.-C., and Delbruck, T. (2017). Ddd17: End-to-end davis driving dataset. *Workshop on Machine Learning for Autonomous Vehicles (MLAV) in ICML 2017*.

Borghi, G., Fabbri, M., Vezzani, R., Cucchiara, R., et al. (2018). Face-from-depth for head pose estimation on depth images. *IEEE transactions on pattern analysis and machine intelligence*.

Borghi, G., Pini, S., Vezzani, R., and Cucchiara, R. (2019). Driver face verification with depth maps. *Sensors*, 19(15):3361.

Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T. (2014a). A 240×180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341.

Brandli, C., Muller, L., and Delbruck, T. (2014b). Real-time, high-speed video decompression using a frame- and event-based davis sensor. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 686–689. IEEE.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3213–3223.

Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Neural Information Processing Systems*, pages 2366–2374.

Frigieri, E., Borghi, G., Vezzani, R., and Cucchiara, R. (2017). Fast and accurate facial landmark localization in depth images for in-car applications. In *International Conference on Image Analysis and Processing*, pages 539–549. Springer.

Gallego, G., Lund, J. E., Mueggler, E., Rebecq, H., Delbruck, T., and Scaramuzza, D. (2018a). Event-based, 6-dof camera tracking from photometric depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2402–2412.

Gallego, G., Rebecq, H., and Scaramuzza, D. (2018b). A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1.

Gehrig, D., Rebecq, H., Gallego, G., and Scaramuzza, D. (2018). Asynchronous, photometric feature tracking using events and frames. In *European Conference on Computer Vision*.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Ben-

gio, Y. (2014). Generative adversarial nets. In *Neural Information Processing Systems*, pages 2672–2680.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

Kim, H., Handa, A., Benosman, R., Ieng, S., and Davison, A. J. (2014). Simultaneous mosaicing and tracking with an event camera. In *British Machine Vision Conference*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B. (2017). Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1346–1359.

Lichtsteiner, P., Posch, C., and Delbruck, T. (2006). A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pages 2060–2069. IEEE.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*. Springer.

Lungu, I.-A., Corradi, F., and Delbrück, T. (2017). Live demonstration: Convolutional neural network driven by dynamic vision sensor playing roshambo. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–1. IEEE.

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

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mitrokhin, A., Fermuller, C., Parameshwara, C., and Aloimonos, Y. (2018). Event-based moving object detection and tracking. *arXiv preprint arXiv:1803.04523*.

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

Pini, S., Borghi, G., Vezzani, R., and Cucchiara, R. (2019). Video synthesis from intensity and event frames. In *International Conference on Image Analysis and Processing*, pages 313–323. Springer.

Ramesh, B., Zhang, S., Lee, Z. W., Gao, Z., Orchard, G., and Xiang, C. (2018). Long-term object tracking with a moving event camera. In *British Machine Vision Conference*.

Rebecq, H., Gallego, G., and Scaramuzza, D. (2016). Emvs: Event-based multi-view stereo. In *British Machine Vision Conference*.

Rebecq, H., Ranftl, R., Koltun, V., and Scaramuzza, D. (2019). Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3857–3866.

Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Reinbacher, C., Graber, G., and Pock, T. (2016). Real-time intensity-image reconstruction for event cameras using manifold regularisation. In *British Machine Vision Conference*.

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

Rota Bulò, S., Porzi, L., and Kontschieder, P. (2018). In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Neural Information Processing Systems*, pages 2234–2242.

Scheerlinck, C., Barnes, N., and Mahony, R. (2018). Continuous-time intensity estimation using event cameras. *Asian Conf. Comput. Vis. (ACCV)*.

Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., and Geiger, A. (2017). Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*.

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

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Neural Information Processing Systems*, pages 802–810.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

Zhou, Y., Gallego, G., Rebecq, H., Kneip, L., Li, H., and Scaramuzza, D. (2018). Semi-dense 3d reconstruction with a stereo event camera. In *European Conference on Computer Vision*.

Zhu, A. Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., and Daniilidis, K. (2018). The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039.