Turning Frequency to Resolution: Video Super-resolution via Event Cameras

Yongcheng Jing[1], Yiding Yang[2], Xinchao Wang[3,2], Mingli Song[4], Dacheng Tao[1]
[1]The University of Sydney, [2]Stevens Institute of Technology,
[3]National University of Singapore, [4]Zhejiang University
{yjin9495, dacheng.tao}@sydney.edu.au, yyang99@stevens.edu,
xinchao@nus.edu.sg, brooksong@zju.edu.cn

Figure 1. Influence of temporal frequency on the performance of VSR. A higher frame rate, and hence a smaller displacement between consecutive frames, yields enhanced VSR results obtained by a state-of-the-art approach [53], as evidenced by the rising PSNRs.

Abstract

State-of-the-art video super-resolution (VSR) methods focus on exploiting inter- and intra-frame correlations to estimate high-resolution (HR) video frames from low-resolution (LR) ones. In this paper, we study VSR from an exotic perspective, by explicitly looking into the role of temporal frequency of video frames. Through experiments, we observe that a higher frequency, and hence a smaller pixel displacement between consecutive frames, tends to deliver favorable super-resolved results. This discovery motivates us to introduce Event Cameras, a novel sensing device that responds instantly to pixel intensity changes and produces up to millions of asynchronous events per second, to facilitate VSR. To this end, we propose an Event-based VSR framework (E-VSR), of which the key component is an asynchronous interpolation (EAI) module that reconstructs a high-frequency (HF) video stream with uniform and tiny pixel displacements between neighboring frames from an event stream. The derived HF video stream is then encoded into a VSR module to recover the desired HR videos. Furthermore, an LR bi-directional interpolation loss and an HR self-supervision loss are also introduced to respectively regulate the EAI and VSR modules. Experiments on both real-world and synthetic datasets demonstrate that the proposed approach yields results superior to the state of the art.

1. Introduction

The goal of video super-resolution (VSR) is to recover a high-resolution (HR) video frame from a sequence of low-resolution (LR) frames. With the prevalence of recent HR display technology, VSR techniques have been attracting increasing attention from both the academic and the industrial community. Various applications of VSR include entertainment [21], surveillance [26], as well as medical and satellite imaging [8]. Recently, VSR has also been applied to facilitate high-level vision tasks like action recognition [63].

State-of-the-art VSR techniques have relied on deep neural networks to model intra-frame correlations and inter-frame coherence, so as to recover HR frames. They can be broadly categorized into two streams, the ones based on convolutional neural network (CNN) [53, 50, 24, 61, 18, 13] and those based on recurrent neural network (RNN) [11, 48, 37, 4, 12, 9]. The former category retains temporal information by concatenating multiple consecutive frames as inputs to produce a single HR estimate. The latter category, on the other hand, relies on recurrent connections to capture the temporal dependencies across a sequence of video frames. Both categories have demonstrated visually plausible and quantitatively encouraging results.

Unlike existing approaches that focus on spatial and temporal dependencies, we study the VSR task from an exotic

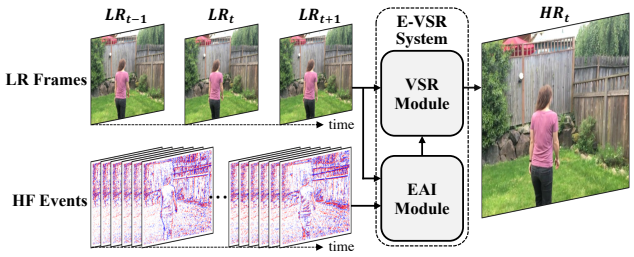


Figure 2. The proposed event-based VSR (E-VSR) system receives low-resolution (LR) videos and high-frequency (HF) event streams as input, and produces high-resolution (HR) video frames. It comprises a general VSR module as well as a EAI module, which exploits event data to generate asynchronous frames with tiny pixel displacements to facilitate VSR.

aspect by exploring the impact of *temporal frequency* on super-resolved results. We found that a higher frequency or frame rate, and consequently a smaller pixel displacement between successive frames, leads to superior super-resolved results. A couple of examples are demonstrated in Fig. 1, where we simulate the varying frame rates of the same videos by sampling frames with different intervals from a high-speed camera [46]. We observe that as the frame rate increases, the super-resolved results derived from a state-of-the-art VSR approach [9] also improve, both visually and quantitatively. This is not totally unexpected, since the larger pixel displacements would, intuitively, make the VSR system harder to capture longer-range temporal dependencies and to utilize the contextual information between frames, resulting in inferior results.

Inspired by this discovery, we introduce to the VSR task a novel sensing modality, *Event Camera*, in the aim to boost the VSR performance by injecting high-frequency (HF) event data into the super-resolving process. In contrast to conventional cameras that capture images at a fixed frame rate, Event Cameras *asynchronously* respond to intensity changes of each pixel in the microsecond level [5, 32], and produce up to millions of asynchronous events each second. The generated asynchronous event data, therefore, precisely measure the pixel variants within an extremely short temporal interval. Apart from the very high speed, Event Cameras also offer numerous other benefits, such as high dynamic range and ultra-low power.

We further propose a novel event-based VSR (E-VSR) system that explicitly accounts for event data, as shown in Fig. 2. It takes as input both a high-frequency (HF) event stream and a regular LR video stream, and then feeds the data of two modalities to a general VSR module alongside with an *event-based asynchronous interpolation* (EAI) module. The goal of the EAI module is to leverage the HF event stream to synthesize asynchronous neighboring frames with tiny and uniform RGB pixel displacements in a given video context. Specifically, within EAI we introduce an event-based dynamic conventional layer to handle the spatially- and temporally-varying thresholds in event data. The outputs of EAI are encoded into the VSR module to establish correspondences between consecutive frames. Moreover, we pose a novel LR bi-directional interpolation loss and an HR self-supervised loss, so as to enforce the consistency between the predicted HR frames and the event stream.

We evaluate the proposed E-VSR on the CED color event dataset [42], where the RGB frames and the corresponding color events are collected in a wide range of natural scenes. Due to the novelty of color event data, the CED dataset is currently the only public dataset that contains real color events captured in practical scenarios. To address this issue of limited event data, we build a simulated color event dataset, which is publicly available at https://osf.io/6c3d9/, using an event simulator [34, 6]. Experiments on both datasets demonstrate that E-VSR yields super-resolved results superior to the state of the art, both qualitatively and quantitatively.

In sum, our contribution is a novel scheme that exploits the asynchronous HF event data, delivered by event cameras, to boost the VSR performance. Its rationale is grounded by the observation that HF streams, and hence smaller pixel displacements, tend to yield favorable VSR results. The proposed E-VSR system generates neighboring frames with tiny and uniform pixel displacements derived from the event streams, which facilitate the establishment of temporal correspondence and further strengthen the VSR process. Both quantitative and qualitative results showcase that the proposed E-VSR consistently outperforms the state of the art.

2. Related Work

We briefly review here several topics that are related to our work, including video super-resolution, event cameras, as well as event-based vision algorithms.

Video Super-resolution. VSR has been a long-standing research topic in computer vision [62, 3, 17, 44, 60, 59]. In recent years, various VSR approaches have been proposed to effectively utilize the contextual information among successive frames [2, 27, 48, 9, 53, 61, 18, 23, 13, 50, 12, 24]. These algorithms can be divided into two groups. The idea of the first group is to use an elaborated pipeline, including feature extraction, alignment, fusion and feature reconstruction, to *explicitly* establish accurate correspondences among different frames, so as to effectively utilize the temporal information [2, 48, 37]. For example, in [53], Wang *et al.* design a pyramid, cascading and deformable module for the superior feature alignment, and also propose a temporal and spatial attention module for the effective feature fusion.

By contrast, the second group of VSR *implicitly* utilizes the motion information among neighboring frames

[9, 18, 61]. For example, the work in [18] proposes to exploit dynamic upsampling filters to implicitly compensate motion among neighboring frames. Haris *et al*. also propose a recurrent back-projection network (RBPN) to effectively exploit the temporal contexts for high-quality results [9]. However, most existing VSR algorithms do not consider the case where there are large pixel displacements among adjacent frames, which remains an open challenge.

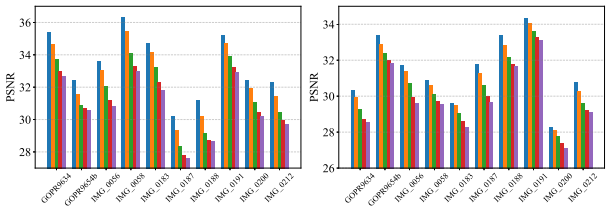
Event Cameras and Event-based Vision. Event cameras are neuromorphically inspired sensing devices [5]. In contrast to conventional cameras that capture synchronized frames at a fixed frame rate, event cameras produce event outputs asynchronously at the exact time they occur with a microsecond-level latency [55]. The most recent type of event camera is *Color-DAVIS346*, which combines a conventional RGB camera sensor to simultaneously produce RGB frames and color event streams without viewpoint differences [49]. The CED dataset [42], which will be used in the experiment, is collected with this *Color-DAVIS346*.

With the nature of extreme low latency ($\sim 1\mu s$), very high temporal resolution, significantly larger dynamic range (140dB) and low power consumption (order of $10mW$) [38, 40], event cameras have been used in a wide range of applications [66, 56, 31, 57, 15, 28, 45, 41, 33, 47, 51, 10]. For example, in [55], Wang *et al*. propose the first event-based gait recognition network, termed as *EV-Gait*, to recognize gait from pure event data. Another work proposed by Pan *et al*. [32, 30] pioneers the use of event data in video deblurring. Specifically, they propose an event-based double integral model, which can generate sharp and high-frame-rate videos using a DAVIS event camera. In [64], Zhu *et al*. present a self-supervised optical flow estimation pipeline using only the event streams, which achieves comparable performance with image-based self-supervised estimation method. They further develop a novel event-based framework [65], which can be used to predict depth, egomotion and optical flow. Another work by Kim *et al*. [20] utilizes event data for 3D reconstruction. A summary of more event-based vision algorithms can be found in [5]. However, to our knowledge, there is no literature that investigates the use of event cameras to benefit the VSR task.

3. Problem Formulation and Pre-analysis

The goal of a VSR model g is to estimate a high-resolution frame $\mathcal{I}_t^{HR}$ from a set of corresponding consecutive low-resolution frames $\{\mathcal{I}_t^{LR}\}$, which can be modeled as $\mathcal{I}_t^{HR} = g_{\theta^*}(\{\mathcal{I}_t^{LR}\})$.

Fig. 1 has qualitatively shown that the size of pixel displacements among $\{\mathcal{I}_t^{LR}\}$ has an influence on the quality of $\mathcal{I}_t^{HR}$. To further validate this observation, in Fig. 3, we conduct more comprehensive experiments with two state-of-the-art VSR algorithms, termed as *Recurrent Back-Projection Network (RBPN)* [9] and *Enhanced Deformable Convolutional Network (EDVR)* [53]. We simulate different sizes of pixel displacements from ten high-quality 240 FPS video sequences [46]. As shown in Fig. 3, the peak signal-to-noise ratios (PSNR) of both RBPN and EDVR decrease with the larger pixel displacements in input videos.



(a) Results of RBPN [9] (2×) (b) Results of EDVR [53] (4×)

Figure 3. Quantitative results of the influence of pixel displacements among neighboring frames on VSR for 2× upscaling (a) and 4× upscaling (b), respectively. The horizontal axis represents different 240 FPS video clips [46]. The blue, orange, green, red, and purple blocks (*i.e*., from left to right) correspond to the video sequences with sampling intervals of 5, 10, 20, 35, and 50 from the same 240 FPS video. Larger sampling intervals correspond to larger pixel displacements among consecutive frames.

This work aims to alleviate this deficiency by introducing auxiliary high-temporal-resolution event data $\mathcal{E}$. The feed-forward super-resolved process of the proposed unified event-based model g' can be formulated as: $\mathcal{I}_t^{HR} = g'_{\theta^*}(\{\mathcal{I}_t^{LR}\}, \{\mathcal{E}\})$. Unlike existing VSR models that learn the temporal correlations by optical flow estimation, the proposed event-based VSR model explicitly uses high-frequency event streams at microsecond resolutions for a more effective utilization of contextual information.

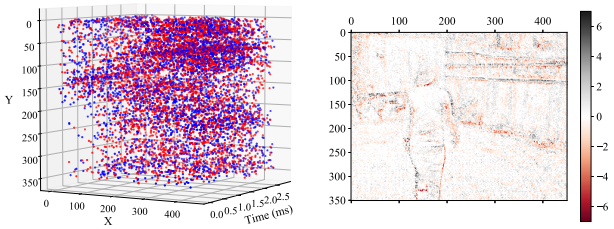
4. Event-based Video Super-resolution

4.1. Overview

To incorporate event streams into VSR, the first issue to be considered is how to convert asynchronous and sparse event data (Fig. 4(a)) into fixed-size representations. To address this issue, we adopt a uniform event aggregation scheme, which will be detailed in Sect. 4.2.

With the obtained event representations, in Sect. 4.3, we introduce the network design of the proposed event-based VSR system. At the heart of the proposed network is an event-based asynchronous interpolation (EAI) module and a VSR module. Specifically, EAI acts as an assistant for VSR, which manipulates raw event representations and feeds the obtained event-based features to the VSR module.

For the training of the two modules, we propose a self-supervised asynchronous interpolation loss upon the predicted consecutive HR frames and HR events, as well as an event-based hierarchical training strategy, which will be further explained in Sect. 4.4.



(a) Raw Events (b) Uniform Event Aggregation

Figure 4. (a) Visualization of asynchronous raw events. Red and blue points correspond to the polarity $p = 1$ and $p = -1$, respectively. (b) Visualization of the uniform event representation. The corresponding RGB frame can be found in Fig. 2.

4.2. Preliminaries and Event Representations

We briefly introduce here the event sensing preliminaries and then derive the uniform event representations for the use of the event-based VSR system. Event cameras are neuromorphically inspired devices that asynchronously respond to logarithmic image intensity changes in microseconds. Let d denote the intensity change at a specific pixel (x, y), which can be formulated as:

$$d = \log \mathcal{I}(x, y, t) - \log \mathcal{I}(x, y, t - \Delta t). \tag{1}$$

An event $\mathcal{E}$ will be triggered whenever d exceeds a specific threshold C in logarithm. The output format of the event data $\mathcal{E}$ is a tuple containing four elements, defined as:

$$\mathcal{E} = (x, y, t, p), \quad p = \begin{cases} 1, & d \geq C \\ -1, & d \leq -C \end{cases}, \tag{2}$$

where p is the polarity that denotes the changing direction. t is the timestamp of the corresponding event. Compared with conventional camera outputs, one distinctive property of event data is that each pixel acts independently and asynchronously, as can be observed in Fig. 4(a). Because of this asynchronous property of event data, it remains a challenging problem in designing an effective event representation for different event-based tasks [5, 7].

To process the asynchronous events with the proposed network, we use here a specialized uniform event aggregation scheme. As analyzed in Sect. 3, for better extraction and utilization of temporal dependencies, there should *not* be large pixel displacements among successive frames. This concept of pixel displacement just matches the principle of event data, where each event corresponds to one single intensity change at a specific pixel. Thus, for the use of events in VSR, we should guarantee that the number of triggered events in each consecutive event representation along the time axis would not differ too much from each other.

Based on this observation, we design a uniform event aggregation scheme to uniformly distribute event streams. Assume that in a given time interval Δt, there is a sequential list of N_e events, sorted by the time each event occurs. We first uniformly divide these N_e events into B bins. A uniform event representation can then be derived by aggregating every $\frac{N_e}{B}$ events. The set of uniform event representations during Δt can be formulated as:

$$\{E_u(x, y)\}_{\Delta t} = \{ \sum_{i=n_e}^{n_e + \frac{N_e}{B}} p_i(x, y) \}, \tag{3}$$

where $n_e = (k-1)\frac{N_e}{B}$, $k = \{1, 2, \ldots, B\}$. $p_i(x, y)$ means that the corresponding event of p_i is triggered at the pixel (x, y). The visualization of an example uniform event representation is shown in Fig. 4(b).

4.3. Network Architecture

The network architecture of the proposed event-based VSR system is shown in Fig. 5. There are primarily three components in the proposed architecture, termed as *Pre- and Post-processing Module*, *Event-based Asynchronous Interpolation (EAI) Module*, and *Video Super-resolution (VSR) Module*.

Pre- and Post-processing Module. The pre- and post-processing module aims to handle the specialized *4-channel* data from the event camera. The imaging system in ColorDAVIS346 event camera relies on an 8×6mm CMOS chip patterned with RGBG filters [49]. As a result, the captured raw color events generally have 4 event channels (*i.e.*, RGBG). In correspondence to the 4-channel events, the associated color image frames should also have the raw image form with 4 image channels, differing from normal 3-channel data.

To process these 4-channel events and image frames, one possible solution is to address this issue on the input side, *i.e.*, converting 4-channel events and image frames into 3 channels (RGB) by developing a specialized event-based ISP system [43, 58]. However, an effective event-based ISP pipeline is an independent research topic, which is beyond the focus of this work. Instead, we resort to another more straightforward solution, which is to directly feed the 4-channel data into the subsequent modules. Then, on the final output side, we conduct image demosaicking and gamma correction to transform the 4-channel super-resolved results into the RGB ones, as done in [36].

Event-based Asynchronous Interpolation (EAI) Module. The primary challenge towards an event-based VSR system is how to incorporate the information contained in high-temporal-resolution event streams to facilitate the learning process of the VSR module. To address this issue, we propose here a baseline method for the utilization of event data. Specifically, based on the analysis in Sect. 3, we build a specialized EAI module, which aims to utilize high-frequency asynchronous events to synthesize the corresponding consecutive asynchronous frames with uniform and tiny pixel displacements among each other. The archi-

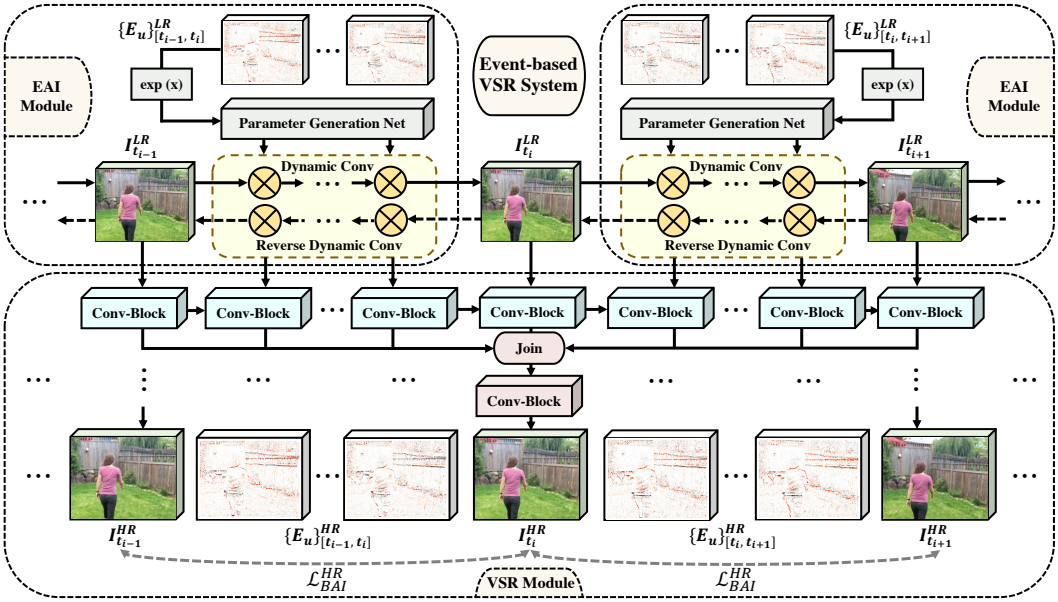


Figure 5. Network Architecture of the proposed event-based VSR system, where the pre- and post-processing module is omitted for clarity. $\mathcal{I}_{t_{i-1}}$, $\mathcal{I}_{t_i}$, and $\mathcal{I}_{t_{i+1}}$ represent the consecutive video frames. $\{E_u\}_{[t_{i-1},t_i]}$ is the corresponding uniform event representation for the time interval $[t_{i-1}, t_i]$. $\mathcal{L}_{BAI}^{HR}$ denotes the HR bi-directional asynchronous interpolation loss, as described in Sect. 4.4. For the sake of clarity, we also omit the illustration of $\mathcal{L}_{BAI}^{LR}$, which is similar to $\mathcal{L}_{BAI}^{HR}$ but replaces all the HR ones with the input LR ones.

tecture of the proposed EAI module is shown in the upper part of Fig. 5.

To further demonstrate the design of EAI, we firstly propose a mathematical model for event-based asynchronous interpolation using the uniform event representations in Sect. 4.2. Assume that E_{u_i} is the i-th uniform event representation in a given time interval Δt. The corresponding image frames are $\mathcal{I}(x, y, t)$ and $\mathcal{I}(x, y, t-\Delta t)$, respectively. From Eq. 1 and Eq. 3, we can derive the relationship between the uniform event representations and image frames as:

$$\mathcal{I}(x, y, t) = \mathcal{I}(x, y, t - \Delta t) \prod_{i=1}^{B} \exp\left(C E_{u_i}(x, y)\right), \quad (4)$$

where B is the number of bins, defined in Eq. 3. C is the intensity threshold. Eq. 4 is, in fact, an approximated form that holds except for extreme conditions [32].

Based on Eq. 4, one possible solution to obtain the k-th asynchronous frame (*i.e.*, the target output of EAI), is to simply compute $\mathcal{I}(x, y, t - \Delta t) \prod_{i=1}^{k} \exp(C E_{u_i}(x, y))$ $(k \leq B)$ by using a manually selected threshold C. However, this possible approach of incorporating event information is not optimal, since the threshold C is not a constant [5, 35, 39]. In practice, C would vary with many factors, including temperature, lighting conditions, and electronic noise [29]. In particular, the intensity threshold C would even vary both temporally and spatially from pixel to pixel [1]. To address this issue, Pan *et al.* [32] propose to model the estimation of C as an optimization problem. In their approach, C is regularized based on image priors like total variation. However, this iterative optimization process is computationally expensive, which is not feasible for the task of VSR.

To alleviate this deficiency, we approximate the burdensome optimization process by using dynamic convolutional operations [14, 16]. As shown in Fig. 5, a parameter generation network receives the exponential uniform event representations as inputs and produces the corresponding parameters for the dynamic convolutional layers, which are then used to convolve with image frames to generate the event-based asynchronous frames. In this way, the per-pixel value of C is learned adaptively in a data-driven manner, requiring only one single network forward pass for estimation in inference. Also, the proposed approach has the additional benefit of VSR-orientated estimation of the intensity threshold C. Specifically, in training, the dynamic convolutional part will be jointly optimized with the rest of the whole event-based VSR network. As a result, C can be learned in a VSR-orientated manner.

In summary, the proposed EAI module receives uniform event representations and image frames as inputs. The outputs of EAI are directly forwarded to the subsequent VSR module to facilitate the utilization of inter-frame temporal information. The reverse dynamic convolutional operation in Fig. 5 is designed for the use in the loss functions, which will be detailed in Sect. 4.4.

Video Super-resolution (VSR) Module. The VSR module

in the proposed event-based VSR system is designed primarily based on a state-of-the-art model termed as RBPN [9], but removes the optical flow inputs. Here, we need to clarify that the proposed event-based VSR system can be equally applicable to many existing VSR networks. Considering that RBPN is currently one of the state-of-the-art VSR methods, we use the architecture of RBPN as a preliminary example to validate the effectiveness of the proposed event-based system. As illustrated in Fig. 5, the RBPN-based VSR module receives the *asynchronous* outputs of the EAI module and also the *synchronous* video frames as inputs. The output of the VSR module is the target super-resolved video frame.

4.4. Loss Function and Training Strategy

Loss Function. The proposed loss function comprises three components, termed as *Low-resolution Bi-directional Asynchronous Interpolation (LR-BAI)* loss, *Self-supervised High-resolution Interpolation (HR-BAI)* loss, and *Mean-square-error (MSE)* VSR loss. Specifically, LR-BAI aims to regularize the EAI module, while both the HR-BAI loss and MSE loss are designed to regularize the VSR module.

Derived from the input LR video frames $\mathcal{I}^{LR}$, *Low-resolution Bi-directional Asynchronous Interpolation (LR-BAI)* loss is devised in correspondence to the proposed EAI module on the LR side. LR-BAI loss aims to regularize the parameter generation network in EAI to produce the feasible parameters for dynamic convolutions, which are further used to generate event-based asynchronous LR frames that can facilitate the learning in VSR module.

Specifically, given a set of successive LR frames $\{\mathcal{I}^{LR}_{t_i}\}$ and also the uniform event representations $\{E_u\}^{LR}_{[t_{i-1},t_i]}$ in the corresponding time intervals, the proposed LR-BAI loss can then be formulated as:

$$\mathcal{L}^{LR}_{BAI} = \sum_i \Big(\Big\| f_{\theta^*}\left(\mathcal{I}^{LR}_{t_{i-1}}, \{E_u\}^{LR}_{[t_{i-1},t_i]}\right) - \mathcal{I}^{LR}_{t_i} \Big\| + \Big\| f^{-1}_{\theta^*}\left(\mathcal{I}^{LR}_{t_i}, \{E_u\}^{LR}_{[t_{i-1},t_i]}\right) - \mathcal{I}^{LR}_{t_{i-1}} \Big\| \Big), \tag{5}$$

where f_{θ^*} is the dynamic convolutional operation. According to the property of event data, with the optimal adaptively generated parameters θ^* from $\{E_u\}^{LR}_{[t_{i-1},t_i]}$, the LR frame $\mathcal{I}^{LR}_{t_{i-1}}$ at a specific timestamp can be convolved into the frame $\mathcal{I}^{LR}_{t_i}$ at the next timestamp, which inspires the design of the proposed loss function. Furthermore, we impose a bi-directional interpolation constraint in Eq. 5, where the aforementioned transformation should be conducted in both directions via a pair of dynamic convolution f_{θ^*} and reverse dynamic convolution $f^{-1}_{\theta^*}$, as shown in Fig. 5.

Also, a *Self-supervised High-resolution Interpolation (HR-BAI)* loss is proposed to regularize the VSR module, based on the predicted HR video frames and HR event data. The idea of HR-BAI is to encourage the generated neighboring HR frames to be transformed into each other with HR events, as shown in Fig. 5. Specifically, the mathematical model of HR-BAI, termed as $\mathcal{L}^{HR}_{BAI}$, has a similar form to that of LR-BAI in Eq. 5, but replaces all the LR ones with the predicted HR ones.

In addition to LR-BAI and HR-BAI loss terms, we also use a mean-square-error (MSE) loss like other VSR algorithms. The total loss for the event-based VSR system is a weighted sum of these three loss terms, formulated as $\mathcal{L}_{total} = \mathcal{L}_{MSE} + \alpha\mathcal{L}^{LR}_{BAI} + \beta\mathcal{L}^{HR}_{BAI}$, where α and β are balancing factors.

Hierarchical Training Strategy. We design a hierarchical training strategy to train the proposed multi-module event-based VSR system, as done in the multimodal learning method of [54]. Specifically, for the first K epochs, we train the EAI module and the VSR module separately with the corresponding MSE loss and LR-BAI loss. This design aims to avoid the situation where the optimizations of the EAI and VSR modules compete with each other at the initial stage. Then, in the following $K/2$ epochs, we jointly optimize the entire system with the loss terms $\mathcal{L}_{MSE}+\alpha\mathcal{L}^{LR}_{BAI}$, such that the information contained in the high-frequency event data is utilized to facilitate the learning of the target VSR process. Afterwards, the proposed VSR system is trained for another $K/2$ epochs with the total loss $\mathcal{L}_{MSE} + \alpha\mathcal{L}^{LR}_{BAI} + \beta\mathcal{L}^{HR}_{BAI}$ to further improve the VSR performance with the self-supervised loss term.

5. Experiments

5.1. Experimental Settings

Datasets. We train and primarily evaluate the proposed event-based system on the first and the only *real* color event dataset, the *Color Event Camera Dataset (CED)*. The CED dataset contains a set of color event streams and video sequences in real scenes, as illustrated in Fig. 6(a). More details of the CED dataset can be found in [42]. Also, we try to build an open event-based VSR dataset by using the event simulator [34, 6]. In particular, the simulation of color events requires HF video inputs, where existing VSR datasets do not meet this requirement. Therefore, we first use an iPhone 11 to capture 240 FPS videos with fast motions and challenging textures (Fig. 6(b)) and then apply the event simulator [34, 6] with the setting of random thresholds to generate the corresponding simulated events. Specifically, the set of positive and negative contrast thresholds for each sequence is randomly sampled from a normal distribution, according to the measurements in [25].

Since the variations of the threshold and the distributions of the event data in the real CED dataset are much closer to the practical situation, we primarily use this dataset to evaluate the performance of the proposed system in real-

Table 1. Quantitative results (PSNR/SSIM) of the proposed event-based system (**E-VSR**) and other methods on the CED dataset for 2×.

Clip Name	SPMC* [48]	DUF* [18]	SOF [52]	TDAN [50]	RBPN [9]	**E-VSR (Ours)**
people_dynamic_wave	24.87 / 0.7915	32.02 / 0.9333	33.32 / 0.9360	35.83 / 0.9540	40.07 / 0.9868	**41.08 / 0.9891**
indoors_foosball_2	23.16 / 0.7548	30.55 / 0.9262	30.86 / 0.9253	32.12 / 0.9339	34.15 / 0.9739	**34.77 / 0.9775**
simple_wires_2	22.60 / 0.7789	30.08 / 0.9387	30.12 / 0.9326	31.57 / 0.9466	33.83 / 0.9739	**34.44 / 0.9773**
people_dynamic_dancing	24.44 / 0.7973	31.64 / 0.9369	32.93 / 0.9388	35.73 / 0.9566	39.56 / 0.9869	**40.49 / 0.9891**
people_dynamic_jumping	24.42 / 0.7886	31.57 / 0.9334	32.79 / 0.9347	35.42 / 0.9536	39.44 / 0.9859	**40.32 / 0.9880**
simple_fruit_fast	29.87 / 0.8575	37.46 / 0.9442	37.22 / 0.9390	37.75 / 0.9440	40.33 / 0.9782	**40.80 / 0.9801**
outdoor_jumping_infrared_2	19.35 / 0.6463	25.33 / 0.8162	26.67 / 0.8746	28.91 / 0.9062	30.36 / 0.9648	**30.70 / 0.9698**
simple_carpet_fast	25.91 / 0.6883	31.43 / 0.8811	31.83 / 0.8774	32.54 / 0.9006	34.91 / 0.9502	**35.16 / 0.9536**
people_dynamic_armroll	24.41 / 0.7885	31.38 / 0.9311	32.79 / 0.9345	35.55 / 0.9541	40.05 / 0.9878	**41.00 / 0.9898**
indoors_kitchen_2	23.45 / 0.7732	29.92 / 0.9273	29.61 / 0.9192	30.67 / 0.9323	31.51 / 0.9551	**31.79 / 0.9586**
people_dynamic_sitting	23.56 / 0.7842	30.62 / 0.9331	32.13 / 0.9367	35.09 / 0.9561	39.03 / 0.9862	**39.97 / 0.9884**
Average PSNR/SSIM	24.19 / 0.7681	31.09 / 0.9183	31.84 / 0.9226	33.74 / 0.9398	36.66 / 0.9754	**37.32 / 0.9783**

Note: * denotes that values are obtained from the pre-trained model released by the authors, since the official training code is unavailable.

Table 2. Quantitative results of the proposed E-VSR and other methods on the simulated color event dataset for the scale of 2×.

VSR Methods	SPMC* [48]	DUF* [18]	SOF [52]
Average PSNR/SSIM	21.98 / 0.7581	28.34 / 0.9081	28.56 / 0.9135
VSR Methods	TDAN [50]	RBPN [9]	**E-VSR (Ours)**
Average PSNR/SSIM	29.86 / 0.9236	31.57 / 0.9526	**32.10 / 0.9557**

Table 3. Quantitative results of the proposed E-VSR and other VSR methods for the scale of 4×, corresponding to Tab. 1.

VSR Methods	SPMC* [48]	DUF* [18]	SOF [52]
Average PSNR/SSIM	18.32 / 0.4831	24.43 / 0.8177	27.00 / 0.8050
VSR Methods	TDAN [50]	RBPN [9]	**E-VSR (Ours)**
Average PSNR/SSIM	27.88 / 0.8231	29.80 / 0.8975	**30.15 / 0.9053**

scene situations. We randomly split the sequences in CED into training, validation and testing sets, and report the corresponding comparison results against the state-of-the-art models by retraining them with the same setting.

Implementation Details. The event-based system is trained on two NVIDIA Tesla V100 GPUs with a mini-batch size of 2 per GPU. In our implementation, we set α, β, and K as 0.1, 0.01, and 100, respectively. For uniform event representations, the number of bins B is set to 2. During training, we adopt the Adam optimizer [22]. The learning rates for both the EAI and VSR modules are set to 0.0001. We downscale the HR video frames for 2× with bicubic interpolation to produce the LR input frames. In evaluations, following the settings in [9, 18, 48], we exclude 8 pixels near the image boundary and use the RGB channels for measurements.

5.2. Experimental Results

Comparison with State-of-the-art Methods. We compare the proposed VSR system (E-VSR) with several state-of-the-art VSR methods, including SPMC [48], DUF [18], RBPN [9], SOF [52], and TDAN [50]. Specifically, we use the authors' official implementations to retrain the models of [9, 52, 50] on the CED dataset. For [48, 18], since the official training code is not available, we report the results by using the authors' provided models.

Quantitative comparison results on the CED dataset and the simulated dataset for the upsampling factor of 2 are shown in Tab. 1 and Tab. 2, respectively. Essentially, most of the state-of-the-art VSR methods implicitly or explicitly utilize estimated optical flow to capture motion cues,

Figure 6. Examples of (a) CED dataset and (b) simulated dataset.

so as to incorporate multi-frame contextual information in the process of VSR. The proposed event-based VSR system, however, explicitly learns to exploit high-temporal-resolution and high-dynamic-range event streams for VSR, making it possible to capture fast and accurate motions for more effective utilization of inter-frame temporal information. As shown in Tab. 1 and Tab. 2, the proposed E-VSR outperforms other approaches in terms of PSNR and SSIM, which is consistent with our assumption. Tab. 3 shows the corresponding quantitative comparison results for 4× upscaling on the CED dataset, where the proposed E-VSR also achieves the state-of-the-art VSR performance.

Fig. 7 shows the qualitative results of different methods, corresponding to Tab. 1. We zoom in on the same region (*i.e.*, red and blue frames) to observe the details. The proposed event-based VSR system, as shown in Fig. 7, recovers finer and more accurate details and textures, such as the yellow edges in the first line and also the black wires in the second line. In Fig. 8, we also try to explain this favorable qualitative performance by providing the visualizations of the corresponding uniform event representations used in the proposed event-based system. The event streams, as can be observed from the zoom-in regions, have a higher event

Table 4. Ablation study of the proposed modules, loss terms, and the uniform event representations with different numbers of bins.

Video Super-resolution (VSR) Module	Event-based Asynchronous Interpolation (EAI) Module			Uniform Event Representation		Average PSNR/SSIM
	[LR-BAI Loss]	[HR-BAI Loss]	[Dynamic Conv]	[Bins (B) = 2]	[Bins (B) = 3]	
√	×	×	×	×	×	36.66 / 0.9754
√	√	×	√	√	×	37.04 / 0.9771
√	√	√	×	√	×	36.98 / 0.9773
√	√	√	√	×	√	37.18 / 0.9782
√	√	√	√	√	×	**37.32 / 0.9783**

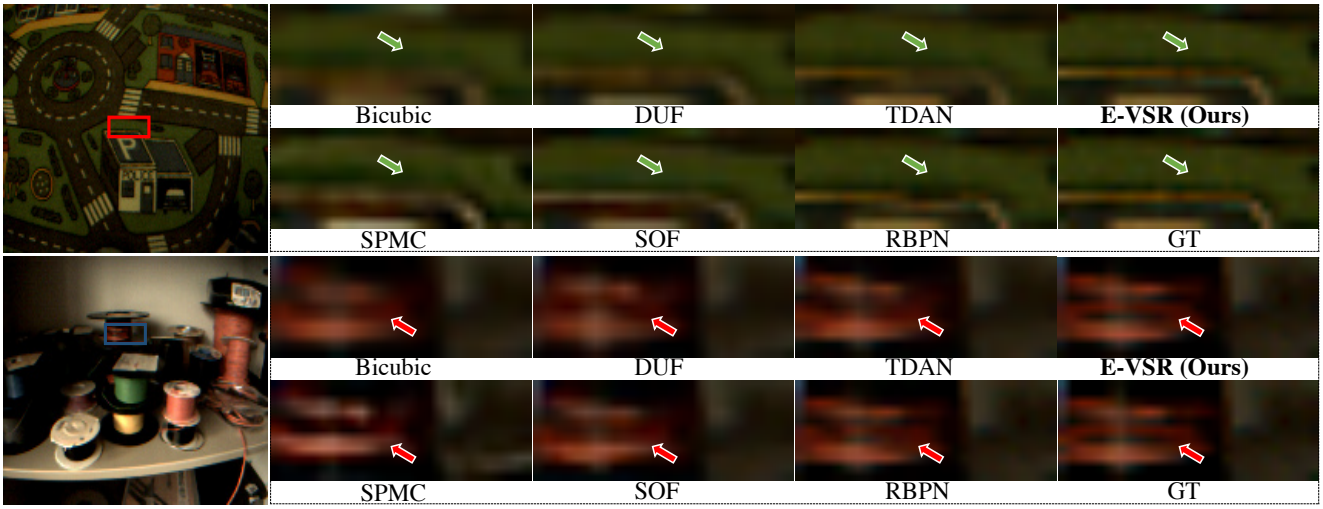


Figure 7. Qualitative VSR results of the proposed E-VSR and other approaches [48, 18, 52, 50, 9] on the CED dataset for the scale of 2×.

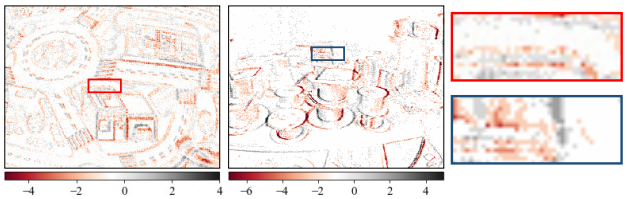


Figure 8. Visualizations of the uniform event representations, corresponding to the qualitative VSR results shown in Fig. 7.

density for the textural and challenging regions, which possibly contributes to the establishment of more accurate correspondences among consecutive frames.

Abalation Study. We perform extensive ablation studies to further validate the effectiveness of the proposed event-based VSR system. Tab. 4 shows the results of the ablation studies on each individual module and loss function. Specifically, by using the event-based asynchronous interpolation module and the LR bi-directional interpolation loss for training, the corresponding model outperforms the one with only the VSR module by about 0.4 dB in PSNR on average. Furthermore, by combining the proposed self-supervised HR interpolation loss to regularize the outputs of VSR, the event-based model can gain further boost by about 0.3 dB in terms of average PSNR.

We also conduct more ablation studies on the design of dynamic convolutions in the EAI module and the number of bins in the uniform event representation in Tab. 4. It can be observed that having more bins does not necessarily lead to superior performance, possibly due to the influence of event noise [5, 19].

6. Conclusions

In this paper, we propose a novel video super-resolution (VSR) scheme that explicitly looks into the role of temporal frequency and utilizes Event Cameras to enhance VSR. The proposed approach exploits the high-frequency and event-aware asynchronous property of event data to reconstruct successive frames with tiny and uniform pixel displacements, leading to the establishment of precise correspondence among consecutive frames in a given video context. Moreover, to address the issue of limited event data, we make an open simulated color event dataset with the event simulator, which will be released for further research. Experimental results demonstrate that the proposed approach achieves performance superior to the state of the art on real-world and synthetic datasets. The proposed system also sheds light on the potential of using both event and RGB sensors on mobile and embedded devices for raising low-level vision performance, which we also look forward to exploring in our future work.

Acknowledgements. This work was supported by Australian Research Council Projects FL-170100117, DP-180103424, IH-180100002, and IC-190100031.

References

[1] Christian Brandli, Lorenz Muller, and Tobi Delbruck. Real-time, high-speed video decompression using a frame-and event-based davis sensor. In *ISCAS*, 2014. 5

[2] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*, 2017. 2

[3] Jianrui Cai, Shuhang Gu, Radu Timofte, and Lei Zhang. Ntire 2019 challenge on real image super-resolution: Methods and results. In *CVPRW*, 2019. 2

[4] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *ICCVW*, 2019. 1

[5] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Joerg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *TPAMI*, 2020. 2, 3, 4, 5, 8

[6] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *CVPR*, 2020. 2, 6

[7] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *ICCV*, 2019. 4

[8] Hayit Greenspan. Super-resolution in medical imaging. *The Computer Journal*, 2009. 1

[9] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, 2019. 1, 2, 3, 6, 7, 8

[10] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *3DV*, 2020. 3

[11] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *NeurIPS*, 2015. 1

[12] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *ECCV*, 2020. 1, 2

[13] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *CVPR*, 2020. 1, 2

[14] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 5

[15] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *CVPR*, 2020. 3

[16] Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen. Dynamic instance normalization for arbitrary style transfer. In *AAAI*, 2020. 5

[17] Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. Stroke controllable fast style transfer with adaptive receptive fields. In *ECCV*, 2018. 2

[18] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 1, 2, 3, 7, 8

[19] Alireza Khodamoradi and Ryan Kastner. O (n)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors. *TETC*, 2018. 8

[20] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *ECCV*, 2016. 3

[21] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Deep sr-itm: Joint learning of super-resolution and inverse tone-mapping for 4k uhd hdr applications. In *ICCV*, 2019. 1

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

[23] Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. Fast spatio-temporal residual network for video super-resolution. In *CVPR*, 2019. 2

[24] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In *ECCV*, 2020. 1, 2

[25] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 $\times$ 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In *ISSCC*, 2006. 6

[26] Frank Lin, Clinton Fookes, Vinod Chandran, and Sridha Sridharan. Super-resolved faces for improved face recognition from surveillance video. In *ICB*, 2007. 1

[27] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, Xinchao Wang, and Thomas S Huang. Learning temporal dynamics for video super-resolution: A deep learning approach. *TIP*, 2018. 2

[28] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *ECCV*, 2020. 3

[29] Yuji Nozaki and Tobi Delbruck. Temperature and parasitic photocurrent effects in dynamic vision sensors. *TED*, 2017. 5

[30] Liyuan Pan, Richard Hartley, Cedric Scheerlinck, Miaomiao Liu, Xin Yu, and Yuchao Dai. High frame rate video reconstruction based on an event camera. *TPAMI*, 2020. 3

[31] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *CVPR*, 2020. 3

[32] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *CVPR*, 2019. 2, 3, 5

[33] Christian Pfeiffer and Davide Scaramuzza. Human-piloted drone racing: Visual processing and control. *RA-L*, 2021. 3

[34] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *CoRL*, 2018. 2, 6

[35] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, 2019. 5

[36] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *TPAMI*, 2019. 4

[37] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *CVPR*, 2018. 1, 2

[38] Cedric Scheerlinck. *How to See with an Event Camera*. PhD thesis, College of Engineering and Computer Science, The Australian National University, 2021. 3

[39] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *ACCV*, 2018. 5

[40] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Asynchronous spatial image convolutions for event cameras. *RA-L*, 2019. 3

[41] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert E Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *WACV*, 2020. 3

[42] Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *CVPRW*, 2019. 2, 3, 6

[43] Eli Schwartz, Raja Giryes, and Alex M Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *TIP*, 2018. 4

[44] Chengchao Shen, Youtan Yin, Xinchao Wang, Xubin Li, Jie Song, and Mingli Song. Training generative adversarial networks in one stage. *arXiv preprint arXiv:2103.00430*, 2021. 2

[45] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *ECCV*, 2020. 3

[46] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, 2017. 2, 3

[47] Sihao Sun, Giovanni Cioffi, Coen de Visser, and Davide Scaramuzza. Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events. *RA-L*, 2021. 3

[48] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, 2017. 1, 2, 7, 8

[49] Gemma Taverni, Diederik Paul Moeys, Chenghan Li, Celso Cavaco, Vasyl Motsnyi, David San Segundo Bello, and Tobi Delbruck. Front and back illuminated dynamic and active pixel vision sensors comparison. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2018. 3, 4

[50] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *CVPR*, 2020. 1, 2, 7, 8

[51] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *RA-L*, 2018. 3

[52] Longguang Wang, Yulan Guo, Li Liu, Zaiping Lin, Xinpu Deng, and Wei An. Deep video super-resolution using hr optical flow estimation. *TIP*, 2020. 7, 8

[53] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019. 1, 2, 3

[54] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *CVPR*, 2017. 6

[55] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In *CVPR*, 2019. 3

[56] Zihao W Wang, Peiqi Duan, Oliver Cossairt, Aggelos Katsaggelos, Tiejun Huang, and Boxin Shi. Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging. In *CVPR*, 2020. 3

[57] Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt. Eventcap: Monocular 3d capture of high-speed human motions using an event camera. In *CVPR*, 2020. 3

[58] Xiangyu Xu, Yongrui Ma, and Wenxiu Sun. Towards real scene super-resolution with raw images. In *CVPR*, 2019. 4

[59] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *CVPR*, 2020. 2

[60] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Learning propagation rules for attribution map generation. In *ECCV*, 2020. 2

[61] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, 2019. 1, 2, 3

[62] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018. 2

[63] Haochen Zhang, Dong Liu, and Zhiwei Xiong. Two-stream action recognition-oriented video super-resolution. In *ICCV*, 2019. 1

[64] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *RSS*, 2018. 3

[65] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *CVPR*, 2019. 3

[66] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based optical flow using motion compensation. In *ECCVW*, 2018. 3