EvUnroll: Neuromorphic Events based Rolling Shutter Image Correction

Xinyu Zhou[1#] Peiqi Duan[1#] Yi Ma[1] Boxin Shi[1,2,3] ✉

[1]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University

[2]Institute for Artificial Intelligence, Peking University [3]Beijing Academy of Artificial Intelligence

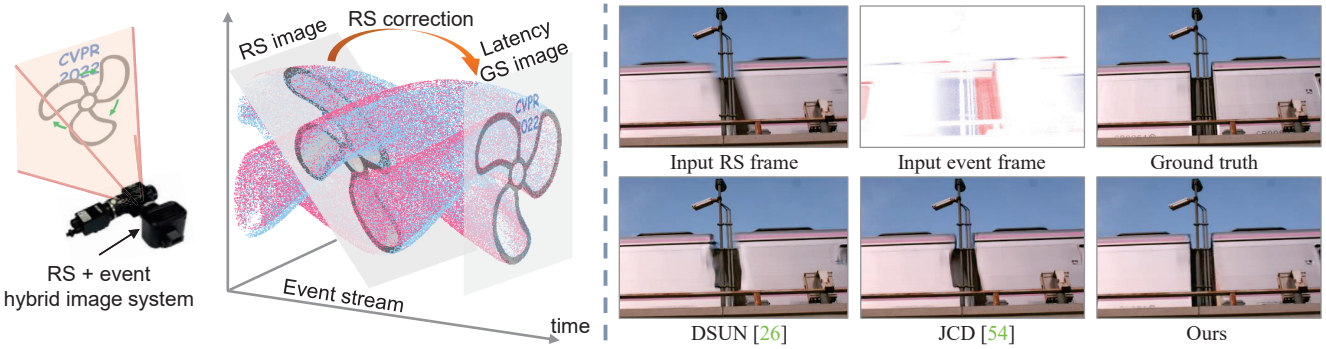


Figure 1: Left (illustration): We build a hybrid imaging system consisting of a rolling shutter (RS) sensor and an event sensor. The event sensor encodes motion and intensity change information, which are well explored by the proposed EvUnroll network to correct the edge distortion (*e.g.*, rotating blades) and restore the intra-frame region occlusion (*e.g.*, occluded logo) in RS images. Right (example result): RS correction results comparison among DSUN [26], JCD [54], and our method.

Abstract

This paper proposes to use neuromorphic events for correcting rolling shutter (RS) images as consecutive global shutter (GS) frames. RS effect introduces edge distortion and region occlusion into images caused by row-wise readout of CMOS sensors. We introduce a novel computational imaging setup consisting of an RS sensor and an event sensor, and propose a neural network called EvUnroll to solve this problem by exploring the high-temporal-resolution property of events. We use events to bridge a spatio-temporal connection between RS and GS, establish a flow estimation module to correct edge distortions, and design a synthesis-based restoration module to restore occluded regions. The results of two branches are fused through a refining module to generate corrected GS images. We further propose datasets captured by a high-speed camera and an RS-Event hybrid camera system for training and testing our network. Experimental results on both public and proposed datasets show a systematic performance improvement compared to state-of-the-art methods.

1. Introduction

CMOS imaging sensors are the mainstream choice for mobile phones and machine vision cameras due to low power consumption and cost [15]. However, commonly row-by-row readout scheme of CMOS sensors will always cause the rolling shutter (RS) effect (also known as the jelly effect) for captured images in scenes with camera or local object motion. Compared with the global shutter (GS) sensor that synchronizes the exposure period of each pixel, RS effects limit the applicability of CMOS sensors in consumer or industrial applications due to edge distortion and region occlusion [14, 22, 24, 54]. As such, the RS correction is a way to make up for such deficiencies.

A well-known challenge for RS correction is to estimate the transformation between RS and GS images [22, 26, 54]. Unlike many image restoration tasks (such as video frame interpolation [16, 32, 36] and image deblurring [17, 23]) which assume that the edge structures of local areas remain unchanged, RS correction needs to deal with the edge distortion. To address this problem, geometric model based methods [2, 12, 12, 29, 39] simplify the RS to GS (RS2GS) transformation via different assumptions, such as the scene is static [29, 30] and straight lines keep straight [43], and employ homography mixture or camera pose estimation to

# Contributed equally to this work as first authors
✉ Corresponding author: shiboxin@pku.edu.cn
Project page: https://github.com/zxyemo/EvUnroll

achieve RS correction [12]. However, these simplified assumptions lead to poor compatibility with complex motions, and the computational cost of such optimization problems is expensive [26]. Deep neural network since firstly demonstrated in [42] have revealed its effectiveness in RS correction by learning camera motion parameters [42, 57], optical flow maps [7], or direct mappings of RS2GS [26,54] from single or multiple consecutive RS frames. Nonetheless, even multi-frame images lack the ability to provide motion within the inter-frame period, which makes the problem still ill-posed.

Another bottleneck for RS correction is the intra-frame region occlusion, which is caused by hybrid models of global and local motion, or depth differences in 3D scenes. The depth-dependent RS distortion could be handled by modeling a 3D scene as layers of planar, and jointly estimating the depth and camera motion from more than three frames [49] at the cost of solving a complicated optimization problem. Deep neural networks could also be employed to learn the underlying camera motion properties and depth maps to restore intra-frame occluded regions [57], but it mainly deals with small occlusion artifacts due to the challenging nature of the single-image problem.

Neuromorphic event cameras are novel visual sensors that enable each pixel to work asynchronously to compare current/subsequent light intensity states and trigger a binary event whenever the log-intensity variation exceeds the preset thresholds [1, 10, 25, 47]. Thanks to their high-temporal-resolution property with microseconds-level sensitivity, event cameras are able to address several limitations of traditional frame-based tasks for dynamic scenes with fast motion. A particular body of the past methods have attended to event-based image reconstruction tasks [3, 5, 6, 35, 40, 51], and a branch of literatures prove the benefit that events could bring to high-frame rate video reconstruction [13, 48]. Hence, the bottlenecks of image-only based RS correction and the benefits of events motivate us to think about: Can we synergize the RS frame and event signals and make use of the high-speed characteristic of events to assist RS correction?

To answer this question, we propose EvUnroll, a neural network that synchronizes and fuses event signals to correct RS images as well as recover consecutive GS frames. Events encode the pixel-wise motion information and intensity change, so we use them to bridge a flow-based connection and a synthesis-based connection between RS and GS frames, and correspondingly establish a flow estimation module to correct edge distortions and a restoration module to restore occluded regions. These two branches are parallel and their outputs are fused through a refinement module to finally restore the GS image at any timestamp in the exposure period of the input RS image. An optional module for dealing with blurry RS images is designed to handle real scenes with blurs. We further collect a new training dataset generated from real videos captured at 5700 fps, and a testing dataset captured with an RS-event hybrid camera system. Overall, this paper makes the following contributions:

- EvUnroll is the first trial to improve RS correction with motion estimation and occlusion region restoration by involving event signals.
- We build a GS-event-RS triplet dataset called Gev-RS using RS-distortion-free frames from a high-speed camera to train the network, and build an RS-event hybrid camera (Fig. 1 left) to collect a real testing dataset.
- EvUnroll outperforms state-of-the-art RS correction methods on commonly used datasets, and obtains a numerical gain of 2.98 dB for PSNR, accompanied by visual quality improvements (Fig. 1 right).

2. Related work

Geometric model based RS correction. Existing geometric model based methods apply different assumptions to simplify the problem of RS correction, such as by assuming camera motion is simple rotational or translational [21, 29], or straight lines keep straight [43]. Meingast *et al*. [29] first develop the geometric model for translational-motion based RS effect. Grundmann *et al*. [12] propose a parameterized homography mixed model. Cho *et al*. [2] take into account the zooming motion, and Purkait *et al*. [39] utilize the underlying scene geometry with the Manhattan world assumption. To accurately estimate the camera motion [21, 27, 38], RANSAC [9] could be applied.

Learning based RS correction. RowColCNN [42] is the first deep learning based RS correction method by learning camera motion parameters. Zhuang *et al*. [57] further proposes the SMARSC network to correct a single RS image by learning to predict both the camera scanline velocity and depth map. Liu *et al*. [26] take two adjacent RS frames as input and learn a dense displacement field via a motion estimation network. To bridge gaps between synthetic and real data, JCD [54] collects a real captured dataset by setting up a GS-RS hybrid camera system, and proposes a network to handle both RS distortion and image blurring. More recently, Fan *et al*. propose an end-to-end RS correction network called SUNet [8], which is built up with a context-aware undistortion flow estimator and a symmetric consistency enforcement module. They also design a network called RSSR [7] to predict a GS video from two consecutive RS images based on scanline-dependent nature.

Event-based image enhancement. Thanks to the high-speed characteristic of the event camera, it is recently used to improve the performance of image enhancement tasks by

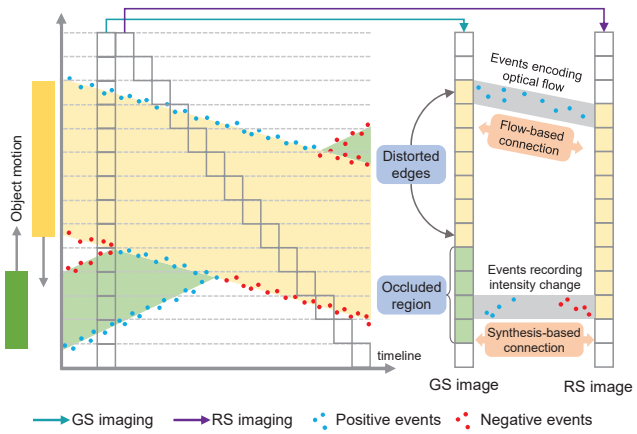


Figure 2: Two latent blocks are moving opposite each other in one dimension. Their time-space trajectory, event triggering, RS and GS frame imaging procedures are recorded along the timeline. We represent the intensity value by different colors (green<yellow<white). Note the RS image cannot precisely record the scene information due to edge distortion (incorrect edge position of the yellow pixels) and occluded region (missing the green pixels). We explore events-encoded optical flow and intensity change to build flow-based connection and synthesis-based connection between RS and GS frames to achieve RS correction.

an events-only or image + events fusion manners. By using only the event stream as input, Reinbacher *et al*. [41] use manifold regularization to reconstruct high-frame-rate videos, and Scheerlinck *et al*. [45] tackle the same problem by proposing a complementary filter. E2VID [40] proposes to learn a video frame synthesis network with LSTM modules. Pan *et al*. [34] make use of the high-speed advantage of events to deblur motion images via jointly optical flow estimation. Mostafavi *et al*. [31] and Han *et al*. [13] also attempt using the learning-based method to solve image super-resolution reconstruction task. Time Lens [48] sets up a hybrid camera system and uses events to assist an RGB camera to achieve video frame interpolation.

3. Methods

In this section, we briefly review the RS imaging and event sensing preliminaries in Sec. 3.1, demonstrate the relationship between the event formation model and its RS/GS frame-based counterpart in Sec. 3.2, and introduce EvUnroll network framework in Sec. 3.3. Implementation details are recorded in Sec. 3.4

3.1. RS imaging and event sensing preliminaries

Let's consider a 3D latent space-time volume ($\Omega \in \mathbb{R}^3$) that records the scene we want to capture in the time range $[0, T]$, and a virtual GS image $I^{\text{GS}}_{t=t_s}$ is formed at any moment $t_s \in [0, T]$. For the case of row-by-row readout RS imaging, we assume the readout direction is from top to bottom, and the resolution is $H \times W$; the exposure time delay between consecutive rows is $\frac{T}{H}$. Then the RS image can be formulated as:

$$I^{\text{RS}} = \sum_{y=1}^{H} M(I^{\text{GS}}_{t=y\frac{T}{H}}, y), \tag{1}$$

where y is the vertical coordinate, $t = y\frac{T}{H}$ means the scan moment of each row, and $M(I, y)$ is an operator to mask the y^{th} row from an image I.

On the event side, the event-triggered output at $t = t_s$ can be formulated as:

$$p_k = \Gamma\{\log(\frac{I_{t=t_s}(x_k, y_k) + b}{I_{t=t_s-1}(x_k, y_k) + b}), \epsilon\}, \tag{2}$$

where $\Gamma\{\theta, \epsilon\}$ is an event-triggering function, ϵ is the contrast threshold, and b is an infinitesimal positive number to prevent $\log(0)$. Events are triggered when $|\theta| \geq \epsilon$. Polarity $p_k \in \{1, -1\}$ indicates the direction (increase or decrease) of intensity change. The event stream output at this space-time volume can be described as a set $\{e_k\}_{k=1}^{N}$, where N denotes the number of events, and each event can be expressed as a four-attribute tuple $e_k = (x_k, y_k, t_s, p_k)$.

3.2. Connect RS and GS image via events

We show in Fig. 2 that the RS and GS imaging are connected by events via two ways: flow-based and synthesis-based connections, which are correspondingly formed by event-encoded motion information and intensity changes. These two connections are key constraints for us to link the GS and RS images via events and to achieve RS correction for removing edge distortion and fulfilling intra-frame occluded regions.

Flow-based connection We estimate an RS2GS flow map to warp a GS image at time $t_s \in [0, T]$ from an RS one. The inverse operation of Eq. (1) can be formulated as:

$$I^{\text{GS}}_{t=t_s}(\mathbf{p}) = I^{\text{RS}}(\mathbf{p} + V(x, y, t = \frac{yT}{H})S(y, t_s)), \tag{3}$$

where $\mathbf{p} = (x, y)$ is the pixel position, $V \in \mathbb{R}^{H\times W\times T\times 2}$ is defined as the velocity vector of each pixel at each moment with unit as pixel/s, and $S(y, t_s) = y\frac{T}{H} - t_s$ represents the time offset between the y^{th} row and the target GS frame timestamp t_s. Then $V(\cdot)S(\cdot)$ becomes the flow map of RS2GS that represents a coordinate translation.

We use events to assist in estimating the velocity vector via $V(x, y, t) = F(\{e_k\})$, where $F(\cdot)$ is the event-based flow estimation function that were previously formulated as a supervision network [55] or a photometric consistency formulation [34]. We propose a flow-based module to learn the velocity vector by events and use a warping-based constraint to correct edge distortions for each moving object.

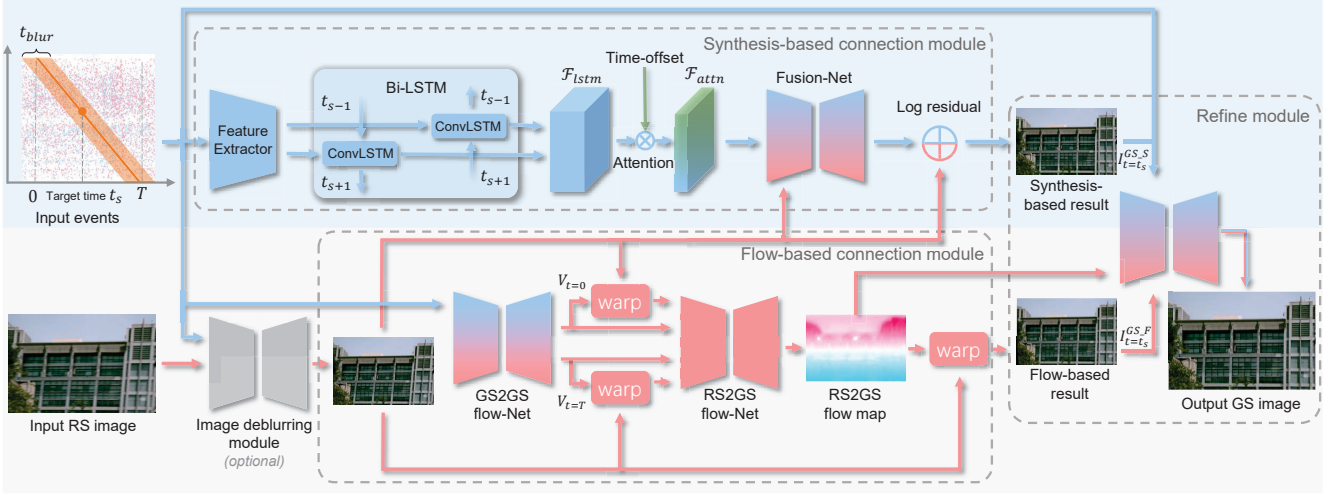


Figure 3: Network architecture of EvUnroll. It consists of three modules: flow-based connection module, synthesis-based connection module, and refine module. An optional deblurring module could be embedded before the flow-based connection module to handle motion blurs in the input RS image.

Synthesis-based connection For each pixel between the paired RS and GS images, intensity variations are encoded as an event stream by log-space thresholding operation. Following the events-to-image synthesis model such as EDI [35], we represent the synthesis-based connection as:

$$\log(I^{\mathrm{GS}}_{t=t_s}(x,y)) = \log(I^{\mathrm{RS}}(x,y)) + \epsilon \sum_{i=\frac{yT}{H}}^{t_s} p_{t=i}, \quad (4)$$

where the sum operation for each pixel refers to the integration of events triggered between the RS2GS time interval. We propose a synthesis-based module to learn a mapping to fulfill intra-frame occluded regions for each timeline.

3.3. EvUnroll framework

EvUnroll consists of three modules to accomplish the RS image correction task, which contains the flow-based connection module, synthesis-based connection module, and refine module, as shown in Fig. 3. A deblurring module handling motion blur in the input RS image can be optionally added. The EvUnroll network takes as input an RS image I^{RS}, the corresponding spatio-temporal events $\{e_k\}$, and a target time $t_s \in [0, T]$, to generate the corresponding GS image. Our network is able to output consecutive GS frames by setting different target time. The backbone of our network is build upon U-Net [44].

Flow-based connection module This module aims to learn an RS2GS mapping, which warps the RS image to a corrected GS image. In order to fully explore the motion priors carried in the event stream, the input $\{e_k\}$ is split into two intervals $[0, t_s]$ and $[t_s, T]$, and both event subsets are binned into an 8-channel event stack by pixel-wisely accumulating the event polarity. We first learn two velocity vectors $V_{t=0}$ and $V_{t=T}$ from the corresponding event subsets via the GS to GS (GS2GS) flow-Net, and then translate them to label the velocity vector for each pixel of the RS image, *i.e.* the time-varying velocity vector $V(x, y, t = y\frac{T}{H})$ described in Eq. (3). Following the optical flow assumption, one element $V(x_0, y_0, t = y_0\frac{T}{H})$ can be expressed as a vector mean of the $\{V_{t=0}(x', y')\}$, a set collecting elements of $V_{t=0}$ whose velocity direction pass through the spatio-temporal position $(x_0, y_0, t = y_0\frac{T}{H})$. The proof is included in the supplementary material. Through the same process, we can also calculate another result of $V(x, y, t = y\frac{T}{H})$ from $V_{t=T}$. Then the network performs warping process with the input RS image I^{RS} through Eq. (3), and outputs two roughly corrected GS images, which are subsequently fed into the RS2GS flow estimation network together with event subsets to further predict a refined RS2GS optical flow map. Finally, a flow-based GS image prediction $I^{\mathrm{GS\_F}}_{t=t_s}$ is warped by the refined RS2GS flow map.

Synthesis-based connection module This module applies the synthesis-based connection to restore a corrected GS image, with special focus on handling occluded regions. We learn log-domain residuals between RS and GS images from events. The input event stream $\{e_k\}$ is first binned into a 16-channel event stack through the same accumulation process as above and sent to a feature extractor to perform local event feature extraction. We employ the Bi-directional ConvLSTM (Bi-LSTM) [11, 46] architecture to correlate features of adjacent time periods, and fuse temporal information into the feature $\mathcal{F}_{\mathrm{lstm}}$. In order to allow the network to perceive the row-specific readout time differences in the RS image, an attention block is adopted to assign time-offset $y\frac{T}{H}$ to the $\mathcal{F}_{\mathrm{lstm}}$, and further obtain the fea-

ture $\mathcal{F}_{\text{attn}}$ that encodes the connection between the RS/GS image pair as shown in Eq. (4). Finally, we concatenate and feed $\mathcal{F}_{\text{attn}}$ and the input RS image into a fusion network, and obtain the log-domain residual to predict the synthesis-based GS image $I^{\text{GS\_S}}_{t=t_s}$, as shown in Fig. 3.

Refine module An attention U-Net [33] based module is introduced to fuse rough prediction results $I^{\text{GS\_F}}_{t=t_s}$ and $I^{\text{GS\_S}}_{t=t_s}$. We use the generated mask m and residual image $I^{r}_{t=t_s}$ to blend the final GS result $I^{\text{GS}}_{t=t_s}$ by

$$I^{\text{GS}}_{t=t_s} = m \cdot I^{\text{GS\_F}}_{t=t_s} + (1-m) \cdot I^{\text{GS\_S}}_{t=t_s} + I^{r}_{t=t_s}. \quad (5)$$

Deblurring module This module aims to recover the RS clear image corresponding to the midpoint of the exposure time of each row. We clip events triggered between the exposure time of each row, and offset the timestamp of events to make events of all rows fall into the same time interval. In this way, the deblurring of RS images is exactly the same as that of GS images.

3.4. Implementation details

Instead of end-to-end training, we empirically find training each module independently works better in our context. A pre-trained image deblurring module to handle motion blur accompanied with the RS effect could be optionally inserted between the input RS image and the flow-based connection module, as a preprocessing to enhance the quality of input images. The flow-based and synthesis-based connection modules are subsequently trained and the refine module is finally trained with the weights of the previous modules fixed. We define the loss between the ground truth and the predicted result as a hybrid of Charbonnier loss [20], perceptual loss [18], and total variation (TV) loss [28]:

$$L = \lambda_1 L_c + \lambda_2 L_p + \lambda_3 L_{tv}. \quad (6)$$

We employ the perceptual loss to preserve details of predictions, and add TV loss to encourage smoothness in the estimated flow map. The hyperparameters $\{\lambda_1, \lambda_2, \lambda_3\}$ are set as: $\{1, 0.05, 0.05\}$ for the flow-based connection module and $\{1, 0.05, 0\}$ for others. Our network is implemented in PyTorch [37] with an NVIDIA TITAN RTX. The Adam optimizer [19] is used for minimizing the loss with an initial learning rate of 0.001, decayed by a factor of 0.2 every 10 epochs. Each module is trained for 30 epochs. The data augmentation is applied by 256×256 random crop for both RS images and corresponding events.

4. Experiment

In this section, we introduce our collected dataset in Sec. 4.1, and qualitatively and quantitatively compare our method with state-of-the-art RS correction methods on public dataset Fastec-RS [26] (Sec. 4.2), our collected dataset Gev-RS (Sec. 4.3), and our real-captured data (Sec. 4.4). In Sec. 4.5, ablation studies are conducted to evaluate the effectiveness of the proposed modules.

4.1. Gev-RS dataset collection

While the majority of existing RS correction datasets have effectively improved the performance of RS correction, there are still unrealistic cases. As a popular RS correction dataset, Fastec-RS [26] collects GS images at a resolution of 640×480 and the frame rate of 2400 fps, and then synthesizes simulated RS images. Although Fastec-RS [26] dataset shows great improvement over previous datasets such as [56], the captured images suffer from quality issues. JCD [54] releases the BS-RSCD dataset captured by a GS-RS hybrid camera system, but the frame rate is only 15 fps, which are not suitable for simulating event stream. To this end, we use a high-speed camera (Phantom VEO 640, F/1.8 85mm lens) to collect high-quality GS frame sequences with 1280×720 resolution at 5700 fps. We capture a total of 29 sequences for both indoor and outdoor scenarios from camera (global) motion to object (local) motion, to cover real challenging scenarios with object occlusion and high-speed motion. The original resolution was downsampled to half (640×360) to suppress the noise level of ground truth. Then we feed the captured videos into the event simulator V2E [4] to generate corresponding event streams under the default parameter settings, and apply the same RS effect simulation process as Fastec-RS [26] to generate RS frames. Eventually, we obtain 3700 "GS-event-RS" triplet clips, and we refer to this dataset as "Gev-RS".

4.2. Comparison on Fastec-RS dataset

We compare EvUnroll with recent RS correction methods DSUN [26], JCD [54], RSSR [7], and SUNet [8] on Fastec-RS [26] dataset. The input settings of the above methods are shown in Fig. 4. For a fair comparison, we set a target time t for testing samples and each method outputs a corrected GS image at this time. We evaluate DSUN [26] and JCD [54] with their released testing code, and obtain the results of RSSR [7] and SUNet [8] from the authors. We simulate the corresponding events stream by V2E [4] and EvUnroll is also retrained using the same Fastec-RS

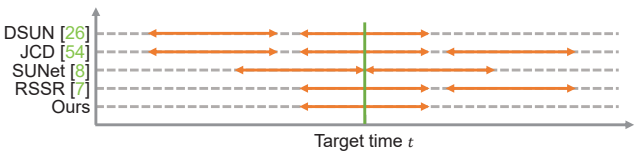


Figure 4: Input settings for comparison methods. The orange two-way arrows represent the total imaging time of an RS frame. Each method outputs a corrected GS image at the target time t.

(a) RS Frame (b) DSUN [26] (c) JCD [54] (d) SUNet [8] (e) Ours (f) GS Frame

Figure 5: Rolling shutter correction results on Fastec-RS [26] dataset. Objects in color boxes (red: lamp pole; yellow: balcony) indicate regions with noticeable differences. (a) Frames with rolling shutter effect. (b)-(e) Correction results for (a) by different methods. (f) Global shutter frames corresponding to (a).

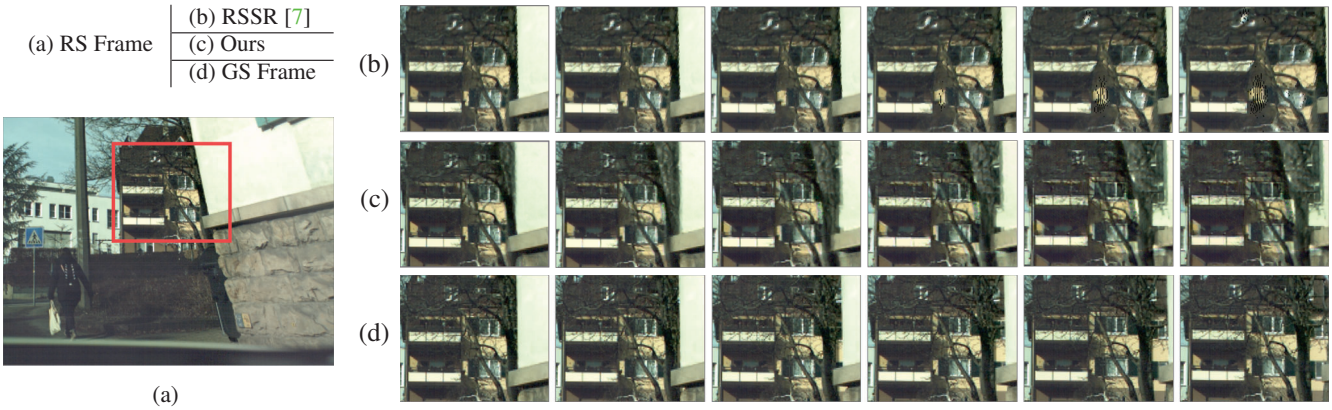


Figure 6: Multi-frame GS correction results on Fastec-RS [26] dataset.

Table 1: Quantitative comparison in PSNR, SSIM, and LPIPS on the Fastec-RS [26] dataset. Lower LPIPS and higher PSNR/SSIM values mean better performance.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
Zhuang *et al*. [56]	21.44	0.71	0.218
DSUN [26]	26.52	0.79	0.122
ESTRNN [53]	27.41	0.84	0.189
JCD [54]	24.84	0.78	0.107
RSSR [7]	21.26	0.78	0.142
SUNet [8]	28.34	0.84	-
EvUnroll (Ours)	**31.32**	**0.88**	**0.084**

[26] training data as above. Visual quality comparisons are shown in Fig. 5 and Fig. 6. In Fig. 5, EvUnroll corrects the distorted poles of the input RS frame (red boxes) in both two examples due to its capability to correct edge distortion and also effectively restores the occluded balcony (yellow boxes) in the second example. Note that we only use a single RS frame as well as the corresponding event stream as input, while others use at least two frames. In Fig. 6, we compare the multi-frame output of our method with a high-frame-rate GS frame reconstruction method RSSR [7]; EvUnroll corrects the distorted edges and restores occluded regions without generating distortion or black edges like RSSR [7] does. The quantitative comparison results are listed in Table 1 (the results of Zhuang *et al*. [57] and ESTRNN [53] are from JCD [54]). We evaluate the average Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [50], and Learned Perceptual Image Patch Similarity (LPIPS) [52] between GS frame and the restoration results of each method. EvUnroll outperforms other methods

(a) RS Frame (b) DSUN [26] (c) JCD [54] (d) Ours (e) GS Frame

Figure 7: Rolling shutter correction results on our Gev-RS dataset. (a) Frames with rolling shutter effect. (b)-(d) Correction results for (a) by different methods. (e) Global shutter frames corresponding to (a).

Table 2: Quantitative comparison in PSNR, SSIM, and LPIPS on our simulation dataset.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
DSUN [26]	23.10	0.70	0.166
JCD [54]	24.90	0.82	0.105
EvUnroll (Ours)	**30.14**	**0.91**	**0.061**

in all three metrics and has a gain of at least 2.98 dB for PSNR.

4.3. Comparison on Gev-RS dataset

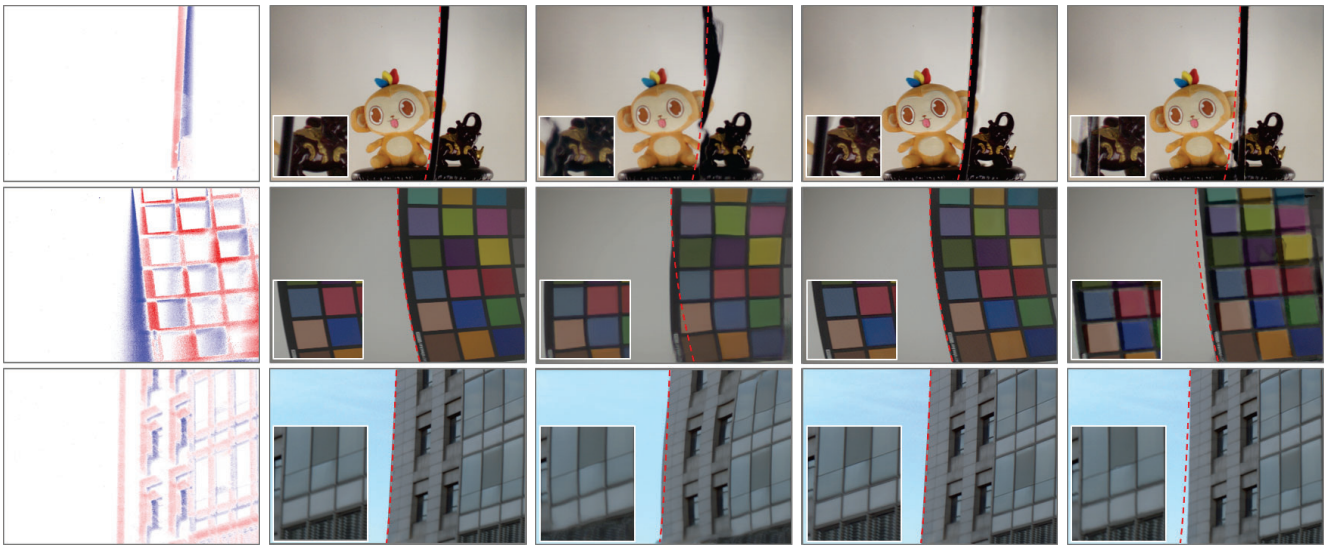
We use our collected Gev-RS dataset to evaluate EvUnroll by comparing with DSUN [26] and JCD [54] whose testing codes are available. We divide the Gev-RS dataset into the training, and testing datasets at a ratio of 7 : 3. We train EvUnroll and retrain DSUN [26] and JCD [54] with the divided training data. Figure 7 shows the qualitative comparison results on some challenging scenarios. The first example is a roadside street scene taken on a moving vehicle in parallel directions and the second one is a building severely distorted by the RS effect. It can be seen that EvUnroll restores the textures and shapes at different depths in the first example and rectifies the vertical edges of the building in the second example. The right region of Fig. 1 also shows a high-speed train example shot by a still camera and our method restores the train compartment intra-frame occluded by the street light. Our deblurring module handles the widespread motion blur in real scenes effectively, outperforming JCD [54], which also deals with simultaneous RS correction and deblurring for dynamic scenes. The quantitative comparison results listed in Table 2 show that EvUnroll outperform DSUN [26] and JCD [54] across all metrics on average. Additional results are included in the supplementary material.

4.4. Comparison on real-captured data

To test EvUnroll for real-world scenarios, we build a hybrid camera system consisting of an RS machine vision camera (LUCID TRI054S IMX490, with 2880×1860 resolution at 20 fps) and an event camera (PROPHESEE GEN4.0, with 1280×720 resolution and about $1\mu s$ latency) via a beam splitter (Thorlabs CCM1-BS013) mounted in front of the two cameras with 50% optical splitting (details can be found in the supplementary material). We capture both indoor and outdoor scenarios with global or local motion. We compare our method with the state-of-the-art methods DSUN [26] and JCD [54], and the visual comparisons are shown in Fig. 8. We correct the distorted stick in the first example and recover the background scene occluded by the distorted area, and restore the shape of square color checkers as well as the building edge in the last two examples. In comparison, DSUN [26] introduces recovery errors and partially distorted edges, and the correction effect of JCD [54] is not obvious due to the challenging motion scenes of our test data.

4.5. Ablation

In this section, we evaluate the effectiveness of the proposed flow-based connection module and synthesis-based connection module and also validate the optional deblurring module by respectively adding it to the above two modules. Therefore, we consider four baseline cases with each of them disabling one/two modules. The minimal loss values during training are used as the evaluation metric, as summarized in Table 3. The qualitative ablation results and

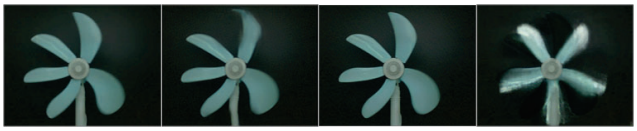


(a) Event Frame (b) Input RS Frame (c) DSUN [26] (d) JCD [54] (e) Ours

Figure 8: Rolling shutter correction results on our real-captured test dataset. (a) Event frames binned in the total readout time of the input RS images (b). (c)-(e) Correction results for (b) by different methods. The red dashed curves (with the same position and shape in (b)-(e)) indicate the distorted edges in the RS images as a reference.

Table 3: Ablation study on different module combinations in EvUnroll.

Case	Flow	Syn.	Deblur	PSNR ↑	SSIM ↑	LPIPS ↓
#1	✓	×	×	23.45	0.805	0.139
#2	×	✓	×	29.02	0.898	0.066
#3	✓	×	✓	26.02	0.832	0.082
#4	×	✓	✓	29.50	0.903	0.065
EvUnroll	✓	✓	✓	30.14	0.912	0.061



(a) RS frame (b) DSUN [26] (c) JCD [54] (d) Ours

Figure 9: Failure case: Recovering the blades of the high-speed rotating fan from an RS image.

analysis are presented in the supplementary material.

5. Conclusion

This paper proposes to use neuromorphic events for correcting RS images as consecutive GS frames. We introduce a novel imaging setup consisting of an RS sensor and an event sensor, and propose a neural network called EvUnroll to solve this problem. We use events to bridge a spatio-temporal connection between RS and GS, establish a flow estimation module to correct edge distortions, and design a synthesis-based connection module to restore occluded regions. The intermediate results of two branches are fused through a refine module to generate corrected GS images. Results on newly collected Gev-RS and real-captured datasets demonstrate the advantage of EvUnroll.

Limitations With our current prototype of a simple hybrid camera system, it is difficult to ensure the microsecond-level synchronization of the RS image with the event stream when shooting high-speed motion scenes, which affects the RS correction performance. A failure case is shown in Fig. 9. Although EvUnroll recovers the blade shape and position with a closer appearance to the real situation than other state-of-the-art methods, there are still obvious artifacts caused by a misalignment between frames and events. Besides, we do not consider the dynamic range gap between RS and event cameras, which may affect the effectiveness of our method in over- or under-exposed regions in images.

Acknowledgement

This work was supported by National Key R&D Program of China (2021ZD0109803) and National Natural Science Foundation of China under Grant No.62136001, 62088102.

References

[1] Shoushun Chen and Menghan Guo. Live demonstration: CeleX-V: a 1m pixel multi-mode event-based sensor. In *Proc. of Computer Vision and Pattern Recognition Workshops*, 2019. 2

[2] Won-ho Cho, Dae-Woong Kim, and Ki-Sang Hong. CMOS digital image stabilization. *IEEE Transactions on Consumer Electronics*, 53:979 – 986, 2007. 1, 2

[3] Jonghyun Choi, Kuk-Jin Yoon, et al. Learning to super resolve intensity images from events. In *Proc. of Computer Vision and Pattern Recognition*, 2020. 2

[4] Tobi Delbruck, Hu Yuhuang, and He Zhe. V2E: From video frames to realistic DVS event camera streams. *arxiv*, June 2020. 5

[5] Peiqi Duan, Zihao Wang, Boxin Shi, Oliver Cossairt, Tiejun Huang, and Aggelos Katsaggelos. Guided event filtering: Synergy between intensity images and neuromorphic events for high performance imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[6] Peiqi Duan, Zihao Wang, Xinyu Zhou, Yi Ma, and Boxin Shi. EventZoom: Learning to denoise and super resolve neuromorphic events. In *Proc. of Computer Vision and Pattern Recognition*, 2021. 2

[7] Bin Fan and Yuchao Dai. Inverting a rolling shutter camera: Bring rolling shutter images to high framerate global shutter video. In *Proc. of International Conference on Computer Vision*, 2021. 2, 5, 6

[8] Bin Fan, Yuchao Dai, and Mingyi He. SUNet: Symmetric undistortion network for rolling shutter correction. In *Proc. of International Conference on Computer Vision*, 2021. 2, 5, 6

[9] Martin A. Fischler and Oscar Firschein. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in Computer Vision*, pages 726–740, 1987. 2

[10] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[11] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18:602–10, 2005. 4

[12] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *Proc. of International Conference on Computational Photography*, 2012. 1, 2

[13] Jin Han, Yixin Yang, Chu Zhou, Chao Xu, and Boxin Shi. EvIntSR-Net: Event guided multiple latent frames reconstruction and super-resolution. In *Proc. of International Conference on Computer Vision*, 2021. 2, 3

[14] Johan Hedborg, Per-Erik Forssén, Michael Felsberg, and Erik Ringaby. Rolling shutter bundle adjustment. In *Proc. of Computer Vision and Pattern Recognition*, 2012. 1

[15] James Janesick, Jeff H. Pinter, Robert Potter, Tom S. Elliott, James Andrews, J. R. Tower, John Cheng, and Jeanne Bishop. Fundamental performance differences between cmos and ccd imagers: part iii. In *Optical Engineering + Applications*, 2009. 1

[16] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *Proc. of Computer Vision and Pattern Recognition*, 2018. 1

[17] Meiguang Jin, Stefan Roth, and Paolo Favaro. Noise-blind image deblurring. In *Proc. of Computer Vision and Pattern Recognition*, 2017. 1

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. of European Conference on Computer Vision*, 2016. 5

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(11):2599–2613, 2019. 5

[21] Yizhen Lao and Omar Ait-Aider. A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *Proc. of Computer Vision and Pattern Recognition*, 2018. 2

[22] Yizhen Lao and Omar Ait-Aider. Rolling shutter homography and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2780–2793, 2021. 1

[23] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proc. of Computer Vision and Pattern Recognition*, 2021. 1

[24] Chia-Kai Liang, Li-Wen Chang, and Homer H. Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330, 2008. 1

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

[26] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *Proc. of Computer Vision and Pattern Recognition*, 2020. 1, 2, 5, 6, 7, 8

[27] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics*, 32, 07 2013. 2

[28] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proc. of Computer Vision and Pattern Recognition*, 2015. 5

[29] Marci Meingast, Christopher Geyer, and S. Shankar Sastry. Geometric models of rolling-shutter cameras. *ArXiv*, abs/cs/0503076, 2005. 1, 2

[30] Mahesh Mohan M.R., A.N. Rajagopalan, and Gunasekaran Seetharaman. Going unconstrained with rolling shutter deblurring. In *Proc. of International Conference on Computer Vision*, 2017. 1

[31] S. M. Mostafavi I., J. Choi, and K. J. Yoon. Learning to super resolve intensity images from events. In *Proc. of Computer Vision and Pattern Recognition*, 2020. 3

[32] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proc. of Computer Vision and Pattern Recognition*, 2018. 1

[33] Ozan Oktay, Jo Schlemper, Loïc Le Folgoc, M. J. Lee, Mattias P. Heinrich, Kazunari Misawa, Kensaku Mori, Steven G. McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning where to look for the pancreas. *ArXiv*, abs/1804.03999, 2018. 5

[34] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *Proc. of Computer Vision and Pattern Recognition*, 2020. 3

[35] 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 *Proc. of Computer Vision and Pattern Recognition*, 2019. 2, 4

[36] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proc. of International Conference on Computer Vision*, 2021. 1

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[38] Pulak Purkait and Christopher Zach. Minimal solvers for monocular rolling shutter compensation under ackermann motion. In *Proc. of Winter Conference on Applications of Computer Vision*, 2018. 2

[39] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in manhattan world. In *Proc. of International Conference on Computer Vision*, 2017. 1, 2

[40] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2, 3

[41] Christian Reinbacher, Gottfried Graber, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *arXiv*, abs/1607.06283, 2016. 3

[42] Vijay Rengarajan, Yogesh Balaji, and A. N. Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *Proc. of Computer Vision and Pattern Recognition*, 2017. 2

[43] Vijay Rengarajan, Ambasamudram N. Rajagopalan, and Rangarajan Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *Proc. of Computer Vision and Pattern Recognition*, 2016. 1, 2

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention 2015*, pages 234–241. 4

[45] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Proc. of Asian Conference on Computer Vision*, 2018. 3

[46] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proc. of Neural Information Processing Systems*, 2015. 4

[47] 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 Trans. Circuit Syst. II: Express Briefs*, 65(5):677–681, 2018. 2

[48] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time Lens: Event-based video frame interpolation. In *Proc. of Computer Vision and Pattern Recognition*, 2021. 2, 3

[49] Subeesh Vasu, Mahesh Mohan M.R., and A.N. Rajagopalan. Occlusion-aware rolling shutter rectification of 3d scenes. In *Proc. of Computer Vision and Pattern Recognition*, 2018. 2

[50] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[51] Zihao Winston 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 *Proc. of Computer Vision and Pattern Recognition*, 2020. 2

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of Computer Vision and Pattern Recognition*, 2018. 6

[53] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *Proc. of European Conference on Computer Vision*, 2020. 6

[54] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *Proc. of Computer Vision and Pattern Recognition*, 2021. 1, 2, 5, 6, 7, 8

[55] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proc. of Computer Vision and Pattern Recognition*, 2019. 3

[56] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *Proc. of International Conference on Computer Vision*, 2017. 5, 6

[57] Bingbing Zhuang, Quoc-Huy Tran, Pan Ji, Loong-Fah Cheong, and Manmohan Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *Proc. of Computer Vision and Pattern Recognition*, 2019. 2, 6