Neural Global Shutter: Learn to Restore Video from a Rolling Shutter Camera with Global Reset Feature

Zhixiang Wang[1,2,3] Xiang Ji[1] Jia-Bin Huang[4] Shin'ichi Satoh[3,1] Xiao Zhou[5†] Yinqiang Zheng[1†]
[1]The University of Tokyo [2]RIISE [3]National Institute of Informatics
[4]University of Maryland College Park [5]Hefei Normal University

Abstract

Most computer vision systems assume distortion-free images as inputs. The widely used rolling-shutter (RS) image sensors, however, suffer from geometric distortion when the camera and object undergo motion during capture. Extensive researches have been conducted on correcting RS distortions. However, most of the existing work relies heavily on the prior assumptions of scenes or motions. Besides, the motion estimation steps are either oversimplified or computationally inefficient due to the heavy flow warping, limiting their applicability. In this paper, we investigate using rolling shutter with a global reset feature (RSGR) to restore clean global shutter (GS) videos. This feature enables us to turn the rectification problem into a deblur-like one, getting rid of inaccurate and costly explicit motion estimation. First, we build an optic system that captures paired RSGR/GS videos. Second, we develop a novel algorithm incorporating spatial and temporal designs to correct the spatial-varying RSGR distortion. Third, we demonstrate that existing image-to-image translation algorithms can recover clean GS videos from distorted RSGR inputs, yet our algorithm achieves the best performance with the specific designs. Our rendered results are not only visually appealing but also beneficial to downstream tasks. Compared to the state-of-the-art RS solution, our RSGR solution is superior in both effectiveness and efficiency. Considering it is easy to realize without changing the hardware, we believe our RSGR solution can potentially replace the RS solution in taking distortion-free videos with low noise and low budget.

1. Introduction

Image sensor is the important component converting photons into digital signals, for machine to see [1, 27], to understand [5, 38], and to recreate [10, 19] the visual world. It comprises millions of spatially distributed photodiodes, namely *pixels*, performing photoelectric conversion and charge accumulation when photons arrive during exposure duration. Readout circuits read these accumulations out and converting them into spatially distributed digital signal, *i.e.*, image, when pixels complete charging. Since simultaneously reading all pixels out requires millions of circuits, leading to unaffordable costs, a key design of image sensors is scheduling the exposure time and readout time of different pixels to reuse limited readout circuits. This function is based on the on-chip electronic shutters dominated by two modes: *global* shutter (GS) and *rolling* shutter (RS).

Image sensors with different on-chip electronic shutters hold contrast characteristics. GS-based image sensors expose all pixels simultaneously and transfer accumulated charges to a storage area before readout. In this way, they can read out charges sequentially with a few readout circuits (Figure 1a). But the requirement for additional storage increases their expense and power consumption and leads to more noises. Differently, RS-based image sensors expose pixels scanline by scanline with a time delay (Figure 1b). This delay enables RS sensors to overlap exposure and readout time, leading to a higher frame rate. Besides, exemption from additional storage area attributes to RS sensors lower cost and fewer noises. Unfortunately, they bring distortions when scenes or cameras undergo motion. The faster the movement, the larger the distortion. The distortion obstructs vision systems equipped with RS sensors to precision-sensitive applications, *e.g.*, localization [37], optical flow [34], and reconstruction [14]. It naturally poses a research question: if there exists a solution for taking distortion-*free* videos with *low* noise and *low* budget.

We categorize existing solutions into two folds: hardware-based and computational. Hardware-based solutions implement the GS function on RS sensors by placing additional memory nodes to store pixels on the charge [35], the voltage [30], even the digital domain [28]. The disadvantages in sensor size, cost and noise are obvious. Computational methods operate directly on RS outputs by correcting the RS distortions [12, 15, 16, 22, 25, 43, 45]. Though existing methods are different in either the input forms: single-image *vs.* multi-image, or the method types: classical *vs.* learning-

[†] Corresponding authors

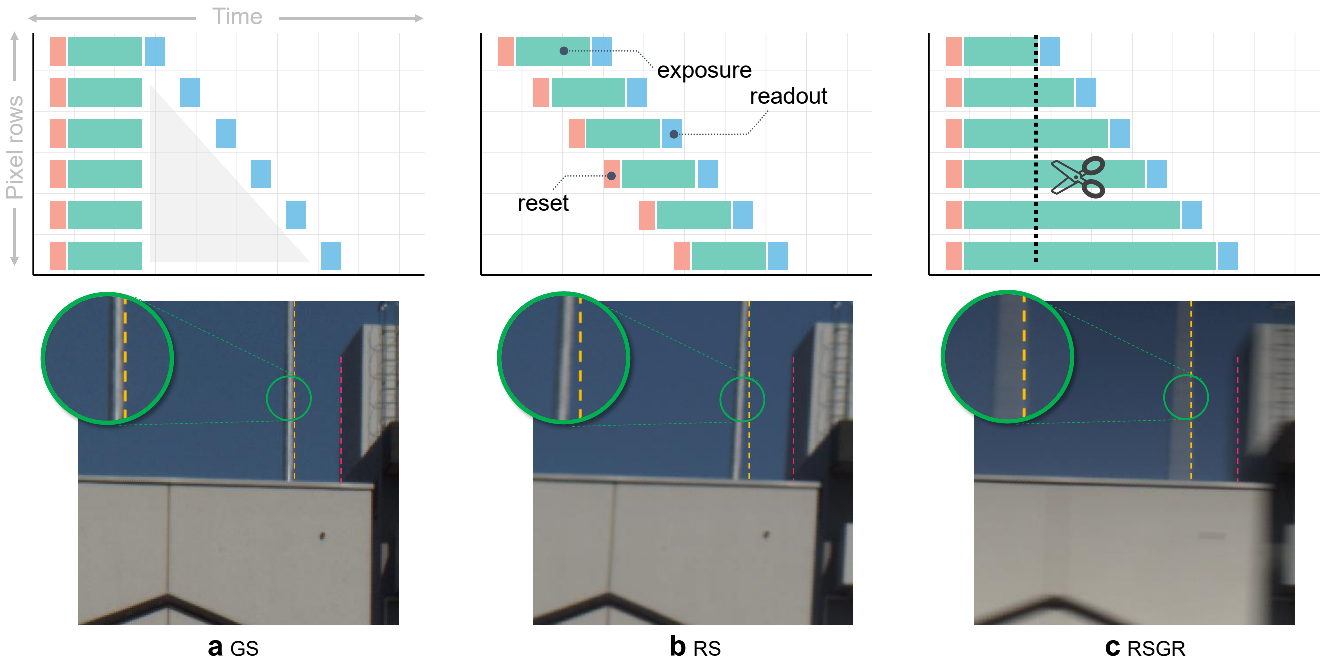

Figure 1. **Different exposure modes.** **a**, GS exposes all pixels simultaneously. This manner requires additional memory nodes to store charged pixels before readout so that it is costly and often suffers from noises. **b**, RS exposes pixels scanline by scanline with a time delay, featured in less noise, higher frame rate, and lower cost. Unfortunately, it leads to distortions when the scene or the camera undergoes motion. Reducing these distortions needs to estimate the motion and complement pixels with the estimation. These steps rely on various assumptions and are time-consuming, limiting RS sensors' applicability. **c**, RSGR is a widely ignored feature of RS. It begins exposure of all pixels at the same time and ends them scanline by scanline. The varying exposure duration of different scanlines yields spatial-varying brightness and blur. RSGR enables us to turn the old RS rectification problem into a deblur-like one.

based, their basic ideas that complement the distorted pixels through estimating pixel-wise motions are the same. In the case of *single* image input, due to the ill-posedness, classical methods estimate motion based on additional assumptions on the scenes [12, 25] or the camera motions [22]. Learning-based approaches relax the assumptions on scenes by digesting the implicit prior, but those on camera motions [24, 45] still hold. The assumptions on either scenes or camera motions compromise their practical applicability. When the input comprises *multiple* images, this problem becomes well-posed, since one can directly estimate the motion between two consecutive frames with classical [43, 44] or learning-based methods [16]. Then, they approximate the displacement between the RS frame and the *virtual* GS frame and use the approximation for warping. But, they usually cannot work under large and complicated camera motions since complementing information is essentially hard, especially when they encounter *oversimplified* motion models. Moreover, the motion estimation steps are often computationally *inefficient* and is unacceptable by real-time applications.

In this paper, we propose a new solution based on a widely ignored feature of RS sensors—*global reset* (GR). This feature allows us to *convert* the old RS rectification problem into a deblur-like one. Thus, we can throw away inaccurate and time-consuming motion estimation steps and make the problem easier to solve. It is because RSGR exposes all pixels at the same time like GS rather than scanline by scanline with a constant time delay like conventional RS [21], as Figure 1c shows. The varying exposure duration of different scanlines yields spatial-varying blur and brightness when capture undergoes motion. This distortion is different from pixel shifts of the RS sensor. Most RS sensors come with this feature that allows them to use mechanical shutters or strobe lights to overcome the distortion, like what Bradley *et al*. [2] does to solve the RS distortion. We relax the hardware requirements in a computational way, as shown in Figure 2. To facilitate the development and evaluation of data-driven algorithms, we build an optic system to take paired RSGR/GS videos simultaneously. This system offers our community a new dataset consisting of 79 paired RSGR/GS video sequences captured under real scenes. We further propose an new algorithm to tackle the unique distortion. Our method contains three main components: **1)** a spatial-aware feature encoder that extracts low-dimensional feature representations for each input RSGR frame. It has two special designs: exposure encoding (EE) and spatial at-

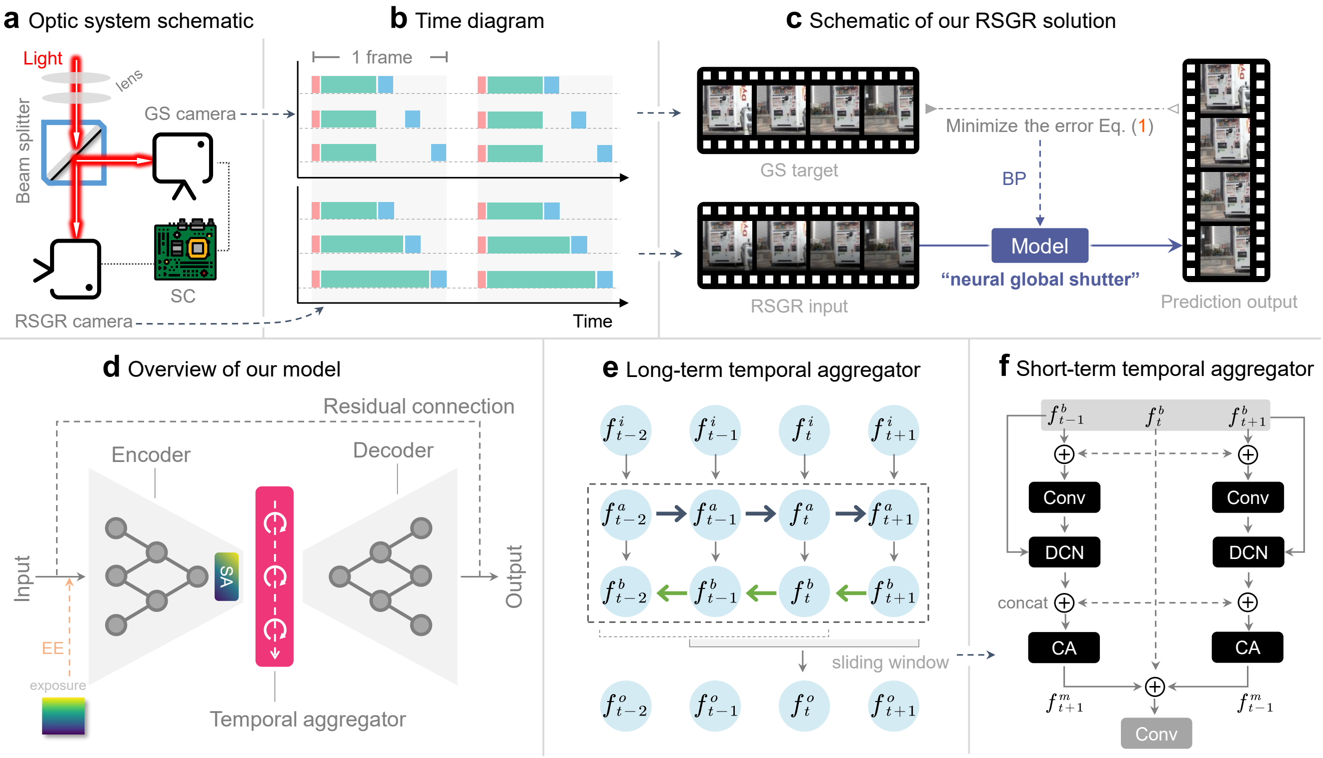
a Optic system schematic
Light
lens
Beam splitter
GS camera
SC
RSGR camera
b Time diagram
1 frame
Time
c Schematic of our RSGR solution
GS target
Minimize the error Eq. (1)
BP
RSGR input
Model
"neural global shutter"
Prediction output
d Overview of our model
Residual connection
Encoder
Decoder
Input
Output
SA
EE
exposure
Temporal aggregator
e Long-term temporal aggregator
f^i_{t-2} f^i_{t-1} f^i_t f^i_{t+1}
f^a_{t-2} f^a_{t-1} f^a_t f^a_{t+1}
f^b_{t-2} f^b_{t-1} f^b_t f^b_{t+1}
sliding window
f^o_{t-2} f^o_{t-1} f^o_t f^o_{t+1}
f Short-term temporal aggregator
f^b_{t-1} f^b_t f^b_{t+1}
Conv
Conv
DCN
DCN
concat
CA
CA
f^m_{t+1}
f^m_{t-1}
Conv

Figure 2. **The proposed method.** **a**, We divide an incoming light into two same parts through a beam splitter and feed these splits into a GS camera and an RSGR camera, respectively. Since the cameras are spatially aligned using a calibration tool and synchronized through synchronization circuits (SC), we can record the light with two different exposure manners. **b**, We keep the first scanline's exposure time of GS and RSGR cameras to be equal. **c**, Given the RSGR video capture, our model outputs a corresponding GS video prediction in an end-to-end fashion. We optimize the algorithm using the backpropagation (BP) technique by minimizing the error between the prediction and the target GS video capture. As the target GS videos suffer from noises, we carefully chose the loss function to suppress the negative effect. **d**, Our model is based on encoder-decoder structure. We employ EE and SA to generate spatial-sensitive feature maps. Given the processed feature maps, we use dual temporal information aggregators to gather both long-term (**e**) and short-term temporal information (**f**). They create powerful feature maps for the decoder to render clean GS videos. To keep the details, we use the residual connection.

tention (SA) [7] for producing spatial-sensitive feature maps. **2)** a module with two recurrent neural networks (RNNs) [26] to propagate long-term information along the time axis bidirectionally. **3)** a module stacked with a few convolutional layers (Conv), a deformable convolutional network (DCN) [4], and a channel attention (CA) to fuse neighboring frames. We experimentally demonstrate that the widely ignored global reset feature enables us to recover clean GS videos from RS sensors with existing image-to-image translation algorithms. With specific designs, our algorithm achieves the *best* performance. Especially, it can work under significant camera motion where correspondences between two consecutive frames are hard to estimate for RS solutions. Compared to existing RS solutions, our solution without explicit motion estimation is *effective* and *efficient*. Considering it is easy to realize without changing the hardware, we believe our solution can be an alternative to the RS solution. To sum up, we make the following three contributions:

- **Problem:** we are the first to introduce RSGR, a widely ignored feature, to our community. This feature enables us to convert the old RS rectification problem into a deblur-like one.

- **Optic system and dataset:** we build an optic system that takes paired RSGR/GS videos and offer a new dataset captured under real scenes. The large-scale paired dataset enables developing and evaluating data-driven methods. We release the dataset to facilitate following researches[1].

- **Algorithm:** we propose a novel algorithm for RSGR video restoration. We experimentally demonstrate that it can render clean GS videos from distorted RSGR inputs and our integrated solution has the potential to replace the RS solution.

[1] https://github.com/lightChaserX/neural-global-shutter

2. Method

2.1. Paired video acquisition system

Global reset is a widely ignored feature, and investigating how to restore clean GS videos from RSGR videos is fresh. We are not aware of *any* dataset that contains RSGR videos and their corresponding GS counterparts. As the first attempt to crack this challenging nut, we build an optic system (Figure 2a) that captures synchronized RSGR and GS videos to facilitate developing and evaluating new algorithms. The system employs a beam splitter to divide the incoming light into two parts and feeds them into an RSGR and a GS camera. These two cameras are spatially calibrated with a calibration tool and synchronized through a synchronization circuit (Figure 2b). Thus, they can capture the same frames simultaneously. Besides, we also ensure the RSGR camera's exposure duration of the first scanline equals the GS camera's exposure duration. This system enables us to develop data-driven algorithms. Note that, due to the installation restrictions in practice, the actual scanning direction of all RSGR frames in this dataset is bottom-to-top. This will not affect the effectiveness of an algorithm trained on thus frames, as will be verified in the generalization evaluation.

2.2. Neural global shutter

Thanks to the global reset feature, we recover clean GS videos from RS sensors in a deblur-like way. Thus, following the common practice in image/video deblur, we employ the encoder-decoder structure (Figure 2d). Nevertheless, since the challenging RSGR distortion consists of spatial-varying blur and brightness, we make three specific designs, including the spatial-aware encoder, the long-term and short-term temporal information aggregator.

Spatial-aware encoder. We use an encoder E_θ to extract the low-dimensional representations $\{\boldsymbol{f}_t^i\}_{t=1}^S$ for a given video segment $\{\boldsymbol{x}_t\}_{t=1}^S$. Our encoder E_θ operates on each frame respectively. Two unique designs power its ability to address spatial-varying distortions. First, we encode each pixel's exposure duration (EE) and feed them along with the input frame into E_θ. The input $\boldsymbol{x}_t$, therefore, has four channels. This design comes from our observation that the RSGR distortion gradually changes and is relevant to the exposure duration. Second, we integrate the spatial attention (SA) mechanism [7] into the encoder E_θ for producing spatially selective feature maps, further enabling us to embed the position information. These two components enable the encoder to be adaptive to the exposure duration.

Long-term temporal aggregator. Similar to most of existing video deblur methods use temporal information as an essential cue, we leverage both the *long-term* and *short-term* temporal information from the input video segment $\{\boldsymbol{x}_t\}_{t=1}^S$. Specifically, we aggregate long-term temporal information $\{\boldsymbol{f}_t^i\}_{t=1}^S$ with two RNNs bidirectionally. We first perform forward information aggregation, with t increases from 1 to S, the output at time t is $\boldsymbol{f}_t^a = F_a([\boldsymbol{f}_t^i, \boldsymbol{h}_{t-1}^a])$, with $\boldsymbol{h}_t^a = H_a(\boldsymbol{f}_t^a)$, where $\boldsymbol{h}_t^a$ is the hidden state, $[\cdot,\cdot]$ denotes concatenation operation along the channel axis[2]. We instantiate F_a and H_a with residual blocks (RBs) [6] and residual dense blocks (RDBs) [39]. The initial hidden state $\boldsymbol{h}_0^a$ is set to 0. Given the outputs from the forward aggregator, we then perform backward information aggregation, with t decreases from S to 1, the output at time t is $\boldsymbol{f}_t^b = F_b([\boldsymbol{f}_t^a, \boldsymbol{h}_{t+1}^b])$, with $\boldsymbol{h}_t^b = H_b(\boldsymbol{f}_t^b)$. Likewise, F_b and H_b comprise RBs and RDBs. The initial hidden state $\boldsymbol{h}_{S+1}^b$ is set to 0. We also tried to initialize $\boldsymbol{h}_{S+1}^b$ with $\boldsymbol{h}_S^a$, but the results are unsatisfactory. We propagate bidirectionally as we find that using only one-directional temporal information will unbalance different frames. It is even worse than without using long-term temporal information.

Short-term temporal aggregator. We have incorporated the long-term temporal information, while the short-term (local) temporal information is also essential. We use a sliding window to traverse the video segment. In the window, we implicitly align the center frame feature $\boldsymbol{f}_t^b$ and its neighboring frame features $\{\boldsymbol{f}_{t\pm k}^b\}_{k=1}^K$ using a deformable convolutional network (DCN) [4]. Given the aligned neighboring features and the center feature, we concatenate them along the channel axis. We use the channel attention (CA) to learn to weight them and a convolutional layer (Conv) to learn to fuse them[3].

Decoder. Taking a refined feature map $\boldsymbol{f}_t^o$ as input, the decoder D_θ renders a clean GS image $\boldsymbol{y}_t$. To capture details, we let our network to learn the difference $\Delta\boldsymbol{x}_t$ between the generated image $\boldsymbol{y}_t$ and the original image $\boldsymbol{x}_t$ with a global residual connection [6]: $\boldsymbol{y}_t = \boldsymbol{x}_t + \Delta\boldsymbol{x}_t$.

Supervision. We train our network in a supervised fashion using the target GS videos as the direct supervisory signal. Specifically, we compute the differences between the rendered video frames $\{\boldsymbol{y}_t\}_{t=1}^T$ and its GS counterparts $\{\bar{\boldsymbol{y}}_t\}_{t=1}^T$. But since GS videos often suffer from noises, we use the perceptual loss [9] and SSIM loss [36] together to suppress the negative effects caused by noisy supervision:

$$L = \lambda \sum_l \|\psi_l(\bar{\boldsymbol{y}}_t) - \psi_l(\boldsymbol{y}_t)\|_1 + (1-\lambda)\phi(\boldsymbol{y}_t, \bar{\boldsymbol{y}}_t)\,, \quad (1)$$

where ψ_l is the feature extracted from l-th layer of a pretrained VGG-19 network [29]; $\phi(\cdot,\cdot) \in [0,1]$ is 1 minus a differentiable version of the SSIM [31]. The parameter λ

[2] We set the length of input video segment S to 8.

[3] We set the length of the window to 3, its stride to 1, and K to 1.

controls the balance between perceptual and SSIM components, which is dynamically set. All of the errors resulting from T frames are summarized together.

3. Experiments

3.1. Setup

Dataset. We use our contributed synthetic and real-world datasets for the following experiments.

- *Real-world dataset.* Using the proposed optic system, we collect the first RSGR dataset and the corresponding GS ground truth. Our dataset consists of 79 video sequences captured under real scenes. Each sequence consists of 300 consecutive frames with 640×640 spatial resolution. The exposure time of the GS camera is 1ms, which is the same as the exposure time of the first scanline of the RSGR camera. Since the exposure time is short, the GS videos are sometimes slightly noisy. We split the dataset into a training set with 27 sequences and testing sets with 52 sequences. The testing sets have two parts. The smaller one SET-I with 3 sequences share similar imaging conditions with the training set, while the larger one SET-II with 49 sequences have worse imaging conditions, where GS videos have noises. We use SET-I as validation set. Note that prevalent datasets used for RS correction and GS deblur either are synthesized [16,32], having significant gaps from real datasets or have no ground truth [43], which are hard for developing and evaluating data-driven algorithms. Unlike them, we capture real scenes with aligned ground truth (GT). We deliver them to the community to facilitate subsequent researches.
- *Synthetic dataset.* We synthesize 25 video sequences for each of the GS, RS, and RSGR exposures. Each video has 29 frames with 512×512 resolution. The scan directions are top-to-bottom ($\downarrow$). The first scanlines of a corresponding GS, RS, and RSGR frame are exposed simultaneously and with the same duration. During synthesizing RSGR videos, we use a parameter ξ to determine the ratio between readout time and the first scanline's exposure duration. We synthesize RSGR videos with 8 different ξ for both training and testing.

Evaluation metrics. Our optic system offers the convenience of using captured GS videos as GT to assess different algorithms. The Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) [36] are used as metrics. We calculate them with a single video frame. Considering the distortion is spatial-varying, we also divide the video frame into parts for evaluation.

3.2. External comparison

Comparison with other algorithms. Since this problem is unexplored, we compare our algorithm with several closely related algorithms from four different categories: **1)** unsupervised GS image-based deblur algorithm that uses adversarial training to get rid of the paired training data: deblurGANv2 [11]; **2)** supervised GS image-based deblur algorithm that integrates multi-scale reception fields: SRN [33]; **3)** supervised GS video-based deblur algorithms that leverages the temporal information by fusing neighboring frames or with an RNN: STRCNN [8], DBN [32], IFIRNN [20], and ESTRNN [40]; and **4)** supervised RS correction or deblur methods that end-to-end estimate motion and compensate distortion: DSUR [16] and JCD [41].

Table 1 reports the qualitative results. We observe that all algorithms except for deblurGANv2 improve the original videos. It demonstrates that supervised image-to-image translation algorithms can remove the RSGR distortion. But, as verified by Liu *et al.* [16], they cannot correct the RS distortions since a rectified pixel in the virtual GS image might lie far away from its corresponding pixel in the input RS image. The RS distortion correction requires accurately estimated motion. Moreover, our algorithm achieves the *best* performance. It consistently outperforms others with a large margin on all evaluation metrics. Especially, opposite to other algorithms that tend to improve only high-quality inputs captured without motion, textureless frames, *etc*., our algorithm improves not only the high-quality inputs but also the low-quality counterparts (see *supplemental material*). We suppose it is because the competing algorithms target RS or GS inputs with the same exposure duration for every pixel. Differently, our RSGR video has gradually increased exposure duration along the scan direction, leading to spatially-varying blur and brightness distortion. Our tailored algorithm incorporates a spatial-aware encoder and dual temporal information aggregators to achieve the best performance for correcting the RSGR distortion.

Figure 3 presents qualitative results. Similar to quantitative results, we find that our algorithm renders visually pleasing results that are better than other algorithms. It corrects mixed spatial-varying blur and brightness *without* introducing additional distortions. Remarkably, thanks to our carefully selected loss function, our rendered videos are noise-free even though there are noises in the supervisory signal. Unfortunately, other algorithms not for this problem have sub-optimal performance. They fail to remove the distortions perfectly like us and even bring some artifacts, *e.g*., geometric deformations, color distortions, and noises. For example, deblurGANv2 introduces noises due to adversarial training with noisy supervision. STRCNN and DBN introduce color distortions. IFIRNN and JCD yield geometric deformations owing to the large spatial-varying blur.

Comparison with using other knowledge. We have illustrated that our tailored architecture is superior in RSGR correction. But, it should be noted that besides the archi-

Table 1. **Quantitative comparison**. The performance is measured with mean PSNR/SSIM (higher is better). 'F' denotes evaluation using full-size frames, while 'U', 'M' and 'L' represent using only 200 rows of pixel from the top, middle, and bottom of each frame, respectively. [†]Training DSUR on our task from scratch is difficult. Thus we fine-tune it, unlike other algorithms trained from scratch here. [‡]Our model without using temporal information. **Bold** texts indicate the best method for each metric.

	SET-I				SET-II			
Method	F	U	M	L	F	U	M	L
Input	18.95 / 0.75	25.32 / 0.82	21.56 / 0.81	16.36 / 0.63	17.82 / 0.73	23.64 / 0.77	21.45 / 0.77	15.54 / 0.66
deblurGANv2 [11]	19.97 / 0.73	21.54 / 0.75	23.73 / 0.77	18.17 / 0.69	18.34 / 0.69	20.14 / 0.69	22.14 / 0.71	17.28 / 0.66
SRN [33]	26.87 / 0.86	26.12 / 0.83	27.08 / 0.85	29.59 / 0.89	25.05 / 0.81	24.32 / 0.79	25.65 / 0.81	27.02 / 0.83
STRCNN [8]	24.88 / 0.85	24.27 / 0.83	25.33 / 0.85	27.54 / 0.88	22.59 / 0.81	22.99 / 0.79	23.46 / 0.81	23.66 / 0.83
DBN [32]	26.49 / 0.87	26.50 / 0.85	26.66 / 0.87	28.47 / 0.89	22.57 / 0.81	23.24 / 0.80	23.81 / 0.81	23.24 / 0.82
IFIRNN [20]	28.01 / 0.89	27.20 / 0.88	28.35 / 0.89	29.21 / 0.90	25.17 / 0.82	24.77 / 0.80	25.62 / 0.81	26.94 / 0.84
ESTRNN [40]	25.85 / 0.89	26.67 / 0.88	30.16 / 0.90	25.19 / 0.89	22.72 / 0.83	23.42 / 0.81	26.03 / 0.83	22.86 / 0.83
DSUR[†] [16]	24.72 / 0.84	24.30 / 0.81	25.65 / 0.85	26.63 / 0.86	22.50 / 0.80	22.49 / 0.78	23.87 / 0.81	23.38 / 0.83
JCD [41]	28.15 / 0.85	27.50 / 0.84	28.73 / 0.85	30.44 / 0.87	25.33 / 0.80	24.77 / 0.78	25.71 / 0.80	27.43 / 0.83
Ours-noT[‡]	27.56 / 0.85	26.23 / 0.83	27.55 / 0.85	31.55 / 0.88	25.37 / 0.80	24.74 / 0.77	25.65 / 0.79	27.29 / 0.82
Ours	**32.72 / 0.92**	**31.83 / 0.92**	**33.01 / 0.92**	**34.65 / 0.92**	**27.29 / 0.85**	**26.96 / 0.84**	**27.57 / 0.85**	**28.35 / 0.86**

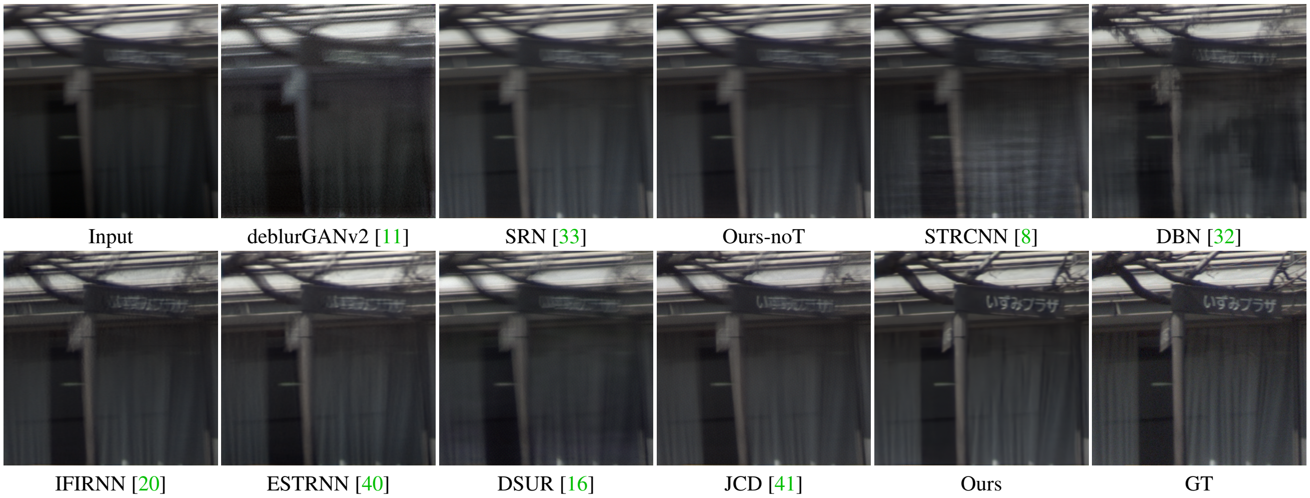
Input
deblurGANv2 [11]
SRN [33]
Ours-noT
STRCNN [8]
DBN [32]
IFIRNN [20]
ESTRNN [40]
DSUR [16]
JCD [41]
Ours
GT

Figure 3. **Qualitative results.** Note that the scan direction is bottom-to-top (↑).

tecture, there also exists another factor accounting for this success: the *paired* training data, which gives our model sufficient supervision to learn the *task-specific* knowledge. We seek to figure out if others can replace the knowledge. First, we compare RSGR correction using our end-to-end learned knowledge with that borrowed from other tasks (models with pre-trained weights) directly, including RS correction [16], RS deblur [41], GS motion deblur (MT-deblur) [11], and GS out-of-focus deblur (OF-deblur) [13]. From the results in Figure 4, we observe that although using pre-trained knowledge directly relaxes the requirement for a paired dataset, they cannot deal with the spatial-varying brightness and blur. Especially, the spatial-varying blurs lead to annoying geometric distortions. The results confirm our required knowledge is different from other tasks. Second, we use handcraft knowledge instead of learning-based knowledge to correct the spatial-varying brightness distortion. The result is frustrating. Third, we train two unsupervised methods, *i.e.*, CycleGAN [42] and deblurGANv2 [11] without paired data. The results verify our required knowledge cannot be replaced by unsupervised knowledge. Accordingly, we build the optic system and paired data with incredible efforts.

Comparison with RS solution. We argue that our solution is better than the RS solution. **1)** Formulation. With the global reset feature, we convert the RS *rectification* problem into a *deblur-like* one, which can be solved with image-to-image translation algorithms while the RS rectification cannot. **2)** Effectiveness. The results in Figure 5 show that our RSGR solution outperforms the RS solution. Especially, our solution is good at dealing with significant motion (lower neighboring PSNR/SSIM). We also find that our solution

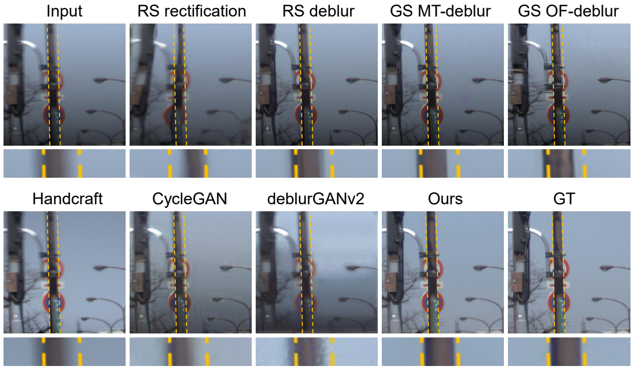

Figure 4. **Comparison with using different knowledge.** Note that the scan direction is bottom-to-top (↑).

goes ahead of the RS solution with a large margin when ξ is small, indicating a large exposure duration. Although our performance decays with the increase of ξ, we are still superior to the RS solution. Considering it is not always to require a large ratio ξ unless we need to capture high-speed videos, our solution is possible to replace RS solution in actual video capture. **3)** Efficiency. In addition to accuracy, our solution without explicit motion estimation is also more efficient than existing RS solutions. Classical two-frame-based RS correction methods often require a few minutes. Zhuang *et al*. [43] processing a 640×480 resolution frame requires 400 seconds, DSUR requires 0.43 seconds, JCD requires 0.83 seconds, but we only need 0.04 seconds[4]. Considering its *effectiveness* and *efficiency*, we believe our RSGR solution can replace RS solutions.

3.3. Internal comparison

Architecture ablation. We conduct ablation experiments on our network architecture by implementing 9 different variants to explore the effectiveness of each module. From the results in Table 2, we have three main findings. First, removing any of the components of our method will weaken the performance. The result verifies that all components are necessary. Second, using spatial-aware modules (T6) has better performance than using only the temporal counterparts (S3). We argue that this is because the power of the temporal information aggregators relies on clean feature maps produced by spatial-aware encoders. With the spatial-aware design, the aggregators would gather distorted feature maps and lose their effectiveness. Third, we surprisingly find that using only one path of the long-term aggregator (T1) is worse than using full (*ours*) and without using the long-term temporal aggregator (T2). We assume this is because our algorithm uses long-term and short-term temporal information together. When we gather the long-term information from one direction, the information imbalance in a video sequence will trouble the short-term aggregation. Consequently, our long-term and short-term aggregators gather information from forward and backward directions and yield the best performance.

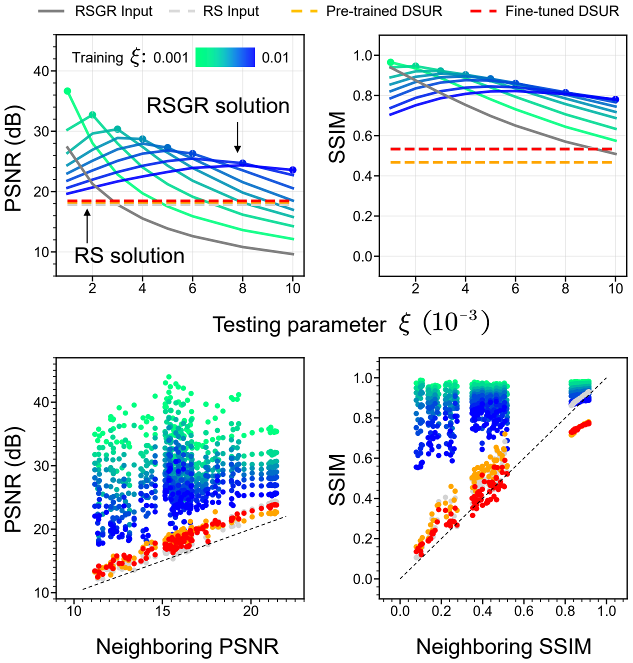

Figure 5. **Comparison with RS solution.** We train and evaluate DSUR [16] with synthesized RS videos as inputs and synthesized GS videos as supervision. Likewise, we train and evaluate our model with 8 different RSGR videos, synthesized with different ξ. We use neighboring metrics to represent the motion degree.

Loss ablation. We also verify the effectiveness of different loss functions in Figure 6, including the perceptual loss [9], the gradient loss [18], the Charbonnier loss [3] and the SSIM loss [36]. We find that all losses other than the SSIM loss result in different artifacts. It is because there exist noises in the supervisory signal. This phenomenon also appears in the adversarial loss. Besides, the perceptual loss has the best performance in structure restoration since it operates on the feature level. Therefore, we combine the SSIM loss and the perceptual loss, leading to the best results.

3.4. Practicality evaluation

Our RSGR solution works well on real applications. Firstly, the rendered results are not only visually pleasing but also applicable to downstream tasks. In Figure 7, we perform single image depth estimation [23] and edge detection [17]. The results reveal that although downstream tasks act worse on original RSGR videos captured undergoing motion, our produced virtual GS videos significantly improve them. The

[4] We conduct all run-time comparisons of DL-based methods, *i.e*. DSUR, JCD, and ours on the same machine with NVIDIA Tesla V100 GPU and Intel(R) CPU@3.80GHz. The run-time result of Zhuang *et al*. [43] borrowed from DSUR is based on an Intel Core i7-7700K CPU.

Table 2. **Ablation experiments of different architectures.** Notations: S1 is our model without (*w/o*) EE; S2 is *w/o* SA; S3 is *w/o* EE and SA; T1 is *w/o* the backward path of the long-term aggregator; T2 is *w/o* the long-term aggregator; T3 is replacing DCN of our model with Conv; T4 is combination of T2 and T3; T5 is T4 replacing CA with Conv; T6 is T5 *w/o* short-term temporal information.

Exp.	F	U	M	L
ours	32.72 / 0.92	31.83 / 0.92	33.01 / 0.92	34.65 / 0.92
S1	32.53 / 0.91	31.28 / 0.90	32.45 / 0.91	34.56 / 0.91
S2	26.51 / 0.90	28.58 / 0.90	30.67 / 0.92	25.66 / 0.89
S3	25.09 / 0.89	25.39 / 0.88	29.56 / 0.91	24.70 / 0.89
T1	31.76 / 0.90	30.60 / 0.89	32.15 / 0.90	34.10 / 0.91
T2	31.99 / 0.90	30.89 / 0.89	32.37 / 0.91	34.47 / 0.91
T3	31.70 / 0.90	30.68 / 0.89	31.99 / 0.90	33.83 / 0.91
T4	31.25 / 0.90	29.91 / 0.89	31.58 / 0.90	34.10 / 0.91
T5	29.31 / 0.89	28.12 / 0.88	29.75 / 0.89	32.48 / 0.90
T6	27.56 / 0.85	26.23 / 0.83	27.55 / 0.85	31.55 / 0.88
input	18.95 / 0.75	25.32 / 0.82	21.56 / 0.81	16.36 / 0.63

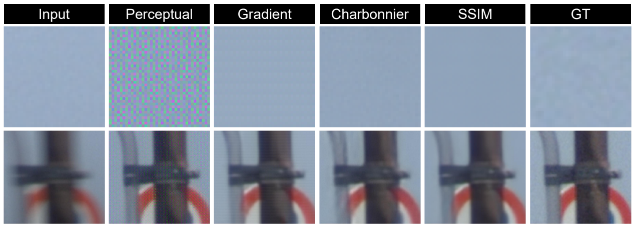

Figure 6. **Ablation experiments of different losses.** Top patches have 37×37 resolution and bottom patches have 140×129 resolution. They come from the same frame.

performance is even comparable to that of using GS videos. Secondly, our algorithm has a favorable generalization ability. The results in Figure 8 prove that our algorithm trained on a specific RSGR camera can be applied directly to a different RSGR camera. In addition, the results of SET-II (as Table 1 shows) can demonstrate the generalization ability of our algorithm across different imaging conditions. We still outperform other methods with a large margin when the testing set has different imaging conditions from the training set. Thirdly, as discussed in Section 3.2, our solution is efficient. Without explicit motion estimation, it is ×10 faster than the state-of-the-art RS solution DSUR [16].

4. Conclusion

This paper first attempts to restore clean GS videos from the RS sensor with the widely ignored global reset feature. We build a new optic system and capture a new dataset with it. Based on the dataset, we developed a data-driven algorithm. We experimentally demonstrate, with this feature, we can convert the RS rectification problem into a deblur-like one, getting rid of the non-trivial motion estimation steps. The results also verify that our tailored algorithm achieves the best performance, potentially applying to real scenarios. Compared to the RS solution, our solution is effective and efficient. Considering it is easy to realize without changing the hardware, we believe our RSGR solution can replace the RS solution.

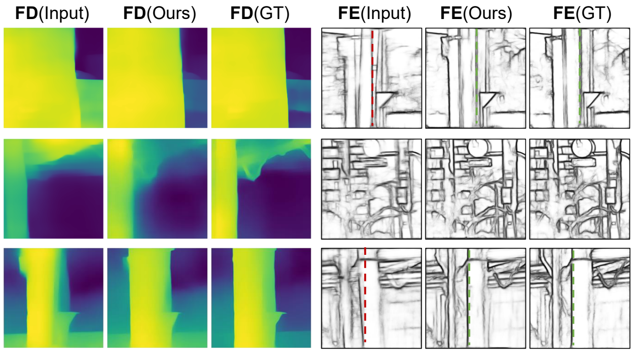

Figure 7. **Downstream applications.** We apply two tasks with pre-trained models including **FD**: the single image depth estimation [23] and **FE**: the edge detection [17] on the original RSGR, our corrected RSGR, and the GS video frames.

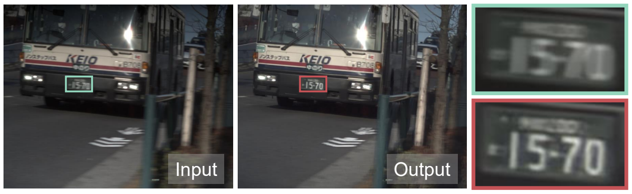

Figure 8. **Generalization evaluation.** We train our model on a certain RSGR camera and test it on another different RSGR camera, which has different exposure time, readout time, resolution, *etc*. Note that the scan direction is top-to-bottom (↓).

Limitations. Although we advocate RSGR over standard RS solution, it should be noted that RS videos do not suffer from any distortion in the completely static situation, yet RSGR still has intensity variations. Although our proposed algorithm can compensate it somehow, we suggest using our RSGR solution for dynamic scenarios. Relating to the point above, the last scanning lines of RSGR are more likely to be overexposed. We thus recommend adjusting the exposure time of RSGR properly. We also believe that the dynamic range might be improved by leveraging no-local exposure variations of RSGR. We leave it to our future work.

Acknowledgments. This research is supported in part by the JSPS KAKENHI Grant Numbers 20H05951, 20H04215, the Key Project of Natural Science Research of Universities in Anhui (KJ2017A934), and the Value Exchange Engineering, a joint research project between Mercari, Inc. and RIISE. ZW also thanks the MEXT Scholarship.

References

[1] Miika Aittala, Prafull Sharma, Lukas Murmann, Adam B Yedidia, Gregory W Wornell, William T Freeman, and Fredo Durand. Computational mirrors: Blind inverse light transport by deep matrix factorization. In *NeurIPS*, 2019. 1

[2] Derek Bradley, Bradley Atcheson, Ivo Ihrke, and Wolfgang Heidrich. Synchronization and rolling shutter compensation for consumer video camera arrays. In *CVPRW*, pages 1–8, 2009. 2

[3] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, pages 168–172, 1994. 7

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 3, 4

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[7] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. In *CVPR*, 2021. 3, 4

[8] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *ICCV*, 2017. 5, 6

[9] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 4, 7

[10] Tero Karras, Samuli Laine, and Timo Aila. A Style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1

[11] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019. 5, 6

[12] Yizhen Lao and Omar Ait-Aider. A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *CVPR*, 2018. 1, 2

[13] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *CVPR*, 2021. 6

[14] Hendrik PA Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM TOG*, 22(2):234–257, 2003. 1

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

[16] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *CVPR*, 2020. 1, 2, 5, 6, 7, 8

[17] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai, and Jinhui Tang. Richer convolutional features for edge detection. *TPAMI*, 41(8):1939 – 1946, 2019. 7, 8

[18] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *CVPR*, 2020. 7

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[20] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *CVPR*, 2019. 5, 6

[21] Junichi Nakamura. *Image sensors and signal processing for digital still cameras*. CRC press, 2017. 2

[22] Pulak Purkait and Christopher Zach. Minimal solvers for monocular rolling shutter compensation under ackermann motion. In *WACV*, 2018. 1, 2

[23] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 7, 8

[24] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*, 2017. 2

[25] Vijay Rengarajan, Ambasamudram N Rajagopalan, and Rangarajan Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *CVPR*, 2016. 1, 2

[26] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 3

[27] Charles Saunders, John Murray-Bruce, and Vivek K Goyal. Computational periscopy with an ordinary digital camera. *Nature*, 565(7740):472–475, 2019. 1

[28] Min-Woong Seo, Myunglae Chu, Hyun-Yong Jung, Suksan Kim, Jiyoun Song, Junan Lee, Sung-Yong Kim, Jongyeon Lee, Sung-Jae Byun, Daehee Bae, et al. A 2.6 e-rms low-random-noise, 116.2 mw low-power 2-mp global shutter cmos image sensor with pixel-level adc and in-pixel memory. In *IEEE Symposium on VLSI Circuits*, 2021. 1

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[30] Laurence Stark, Jeffrey M Raynor, Frederic Lalanne, and Robert K Henderson. A back-illuminated voltage-domain global shutter pixel with dual in-pixel storage. *IEEE Transactions on Electron Devices*, 65(10):4394–4400, 2018. 1

[31] Po-Hsun Su. Differentiable Structural Similarity (SSIM) index. https://github.com/Po-Hsun-Su/pytorch-ssim. 4

[32] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, 2017. 5, 6

[33] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018. 5, 6

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1

[35] Sergey Velichko, Jaroslav Jerry Hynecek, Richard Scott Johnson, Victor Lenchenkov, Hirofumi Komori, Hong-Wei Lee, and Frank YJ Chen. Cmos global shutter charge storage pixels with improved performance. *IEEE Transactions on Electron Devices*, 63(1):106–112, 2015. 1

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 4, 5, 7

[37] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010. 1

[38] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 1

[39] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 4

[40] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*, 2020. 5, 6

[41] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *CVPR*, 2021. 5, 6

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 6

[43] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *ICCV*, 2017. 1, 2, 5, 7

[44] Bingbing Zhuang and Quoc-Huy Tran. Image stitching and rectification for hand-held cameras. In *ECCV*, 2020. 2

[45] Bingbing Zhuang, Quoc-Huy Tran, Pan Ji, Loong-Fah Cheong, and Manmohan Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *CVPR*, 2019. 1, 2