Bringing Rolling Shutter Images Alive with Dual Reversed Distortion

Zhihang Zhong[1,4], Mingdeng Cao[2], Xiao Sun[3], Zhirong Wu[3], Zhongyi Zhou[1], Yinqiang Zheng[1], Stephen Lin[3], and Imari Sato[1,4]

[1] The University of Tokyo, zhong@is.s.u-tokyo.ac.jp
[2] Tsinghua University
[3] Microsoft Research Asia
[4] National Institute of Informatics

Abstract. Rolling shutter (RS) distortion can be interpreted as the result of picking a row of pixels from instant global shutter (GS) frames over time during the exposure of the RS camera. This means that the information of each instant GS frame is partially, yet sequentially, embedded into the row-dependent distortion. Inspired by this fact, we address the challenging task of reversing this process, *i.e.*, extracting undistorted GS frames from images suffering from RS distortion. However, since RS distortion is coupled with other factors such as readout settings and the relative velocity of scene elements to the camera, models that only exploit the geometric correlation between temporally adjacent images suffer from poor generality in processing data with different readout settings and dynamic scenes with both camera motion and object motion. In this paper, instead of two consecutive frames, we propose to exploit a pair of images captured by dual RS cameras with reversed RS directions for this highly challenging task. Grounded on the symmetric and complementary nature of dual reversed distortion, we develop a novel end-to-end model, IFED, to generate dual optical flow sequence through iterative learning of the velocity field during the RS time. Extensive experimental results demonstrate that IFED is superior to naive cascade schemes, as well as the state-of-the-art which utilizes adjacent RS images. Most importantly, although it is trained on a synthetic dataset, IFED is shown to be effective at retrieving GS frame sequences from real-world RS distorted images of dynamic scenes. Code is available at https://github.com/zzh-tech/Dual-Reversed-RS.

Keywords: Rolling shutter correction, frame interpolation, dual reversed rolling shutter, deep learning

1 Introduction

Rolling shutter (RS) cameras are used in many devices such as smartphones and self-driving vision systems due to their low cost and high data transfer rate [19]. Compared to global shutter (GS) cameras, which capture the whole scene at a single instant, RS cameras scan the scene row-by-row to produce an image.

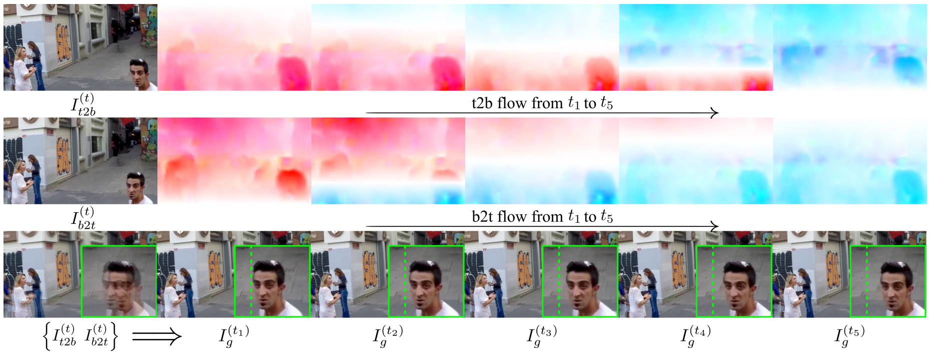

$I_{t2b}^{(t)}$
t2b flow from t_1 to t_5
$I_{b2t}^{(t)}$
b2t flow from t_1 to t_5
$\left\{I_{t2b}^{(t)}\ I_{b2t}^{(t)}\right\} \Longrightarrow$
$I_g^{(t_1)}$
$I_g^{(t_2)}$
$I_g^{(t_3)}$
$I_g^{(t_4)}$
$I_g^{(t_5)}$

Fig. 1: **Consecutive distortion-free frames extracted from a pair of images with reversed rolling shutter distortion.** The 1^{st} row presents the distorted image $I_{t2b}^{(t)}$ from top-to-bottom scanning at time t and the generated optical flows to the extracted frames. The 2^{nd} row presents the distorted image $I_{b2t}^{(t)}$ from bottom-to-top scanning at the same time and its corresponding optical flows. The 3^{rd} row presents the mixed input $\{I_{t2b}^{(t)}\ I_{b2t}^{(t)}\}$ and the extracted global shutter frames $\{I_g^{(t_i)}\}$ in chronological order.

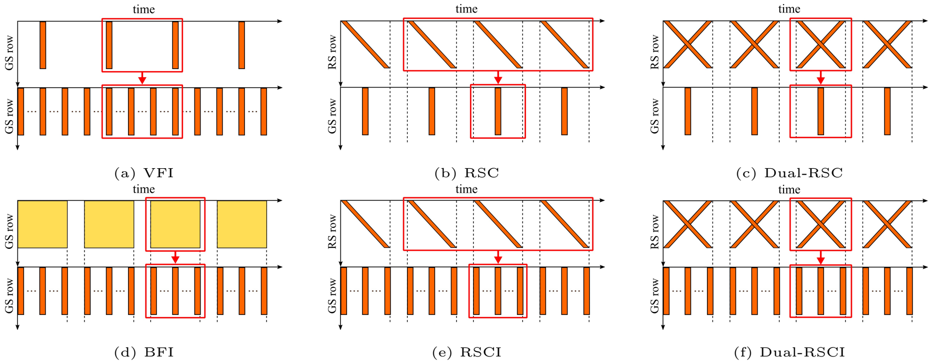
This scanning mechanism may be viewed as sub-optimal because it leads to RS distortion, also known as the jello effect, in the presence of camera and/or object motion. However, we argue that RS photography encodes rich temporal information through its push-broom scanning process. This property provides a critical cue for predicting a sequence of GS images, where distorted images are brought alive at a higher frame rate, which goes beyond the task of recovering a single snapshot as in the RS correction task [2,38,20], as shown in Fig. 1.

Fan and Dai [7] proposed a Rolling Shutter temporal Super-Resolution (RSSR) pipeline for this joint interpolation and correction task. Under the assumption of constant velocity of camera motion and a static scene, RSSR combines a neural network and a manual conversion scheme to estimate undistortion flow for a specific time instance based on the temporal correlation of two adjacent frames (See Fig. 2e for a variant using three consecutive frames). However, even without object motion, the undistortion flow learned in this way tends to overfit the training dataset, because of the intrinsic uncertainty of this setup especially the readout time for each row. As proved in [5], the relative motion of two adjacent RS frames is characterized by the generalized epipolar geometry, which requires at least 17 point matches to determine camera motion. Even worse, it suffers from non-trivial degeneracies, for example, when the camera translates along the baseline direction. In practice, both the relative motion velocity and readout setting will affect the magnitude of RS distortion, and the RSSR model and learning-based RS correction model [20] tend to fail on samples with different readout setups, especially on real-world data with complex camera and/or object motion (See details in Sec. 5 and the supplementary video).

To tackle this problem in dynamic scenes, modeling in the traditional way is particularly difficult, and the inconsistency in readout settings between training data and real test data is also challenging. Inspired by a novel dual-scanning setup [1] (bottom-to-top and top-to-bottom as shown in Fig. 2c) for rolling shutter correction, we argue that this dual setup is better constrained and bears more potential for dynamic scenes. Mathematically, it requires only 5 point matches to determine camera motion, which is much less than that required by the setup with two consecutive RS frames. The symmetric nature of the dual reversed distortion, *i.e.* the start exposure times of the same row in two images are symmetric about the center scan line, implicitly preserves the appearance of the latent undistorted images. Thus, this setup can also help to bypass the effects of inconsistent readout settings. Regarding the hardware complexity and cost, we note that synchronized dual RS camera systems can be easily realized on multi-camera smartphones [1,35] and self-driving cars. Interpolation of dual RS images into GS image sequences provides a promising solution to provide robust RS distortion-free high-fps GS images instead of directly employing expensive high-fps GS cameras. This can be further served as a high-quality image source for high-level tasks such as SfM [37], and 3D reconstruction [20].

Despite the strong geometric constraints arising from dual reversed distortion, it is still intractable to derive a video clip without prior knowledge from training data, as indicated in the large body of literature on video frame interpolation (VFI) from sparse GS frames (Fig. 2a). Therefore, grounded upon the symmetric feature of the dual-RS setup, we design a novel end-to-end **I**ntermediate **F**rames **E**xtractor using **D**ual RS images with reversed distortion (IFED) to realize joint correction and interpolation. Inspired by [20], we introduce the dual RS time cube to allow our model to learn the velocity cube iteratively, instead of regressing directly to an optical flow cube, so as to promote convergence. A mask cube and residual cube learned from an encoder-decoder network are used to merge the results of two reversely distorted images after backward warping. Taking our result in Fig. 1 as an example, the left image in the last row shows the mixed dual inputs $I_{t2b}^{(t)}$ (top-to-bottom scanning) and $I_{b2t}^{(t)}$ (bottom-to-top scanning) at time t. The rest of the row shows the extracted undistorted and smoothly moving frames by our method in chronological order.

To evaluate our method, we build a synthetic dataset with dual reversed distortion RS images and corresponding ground-truth sequences using high-fps videos from the publicly available dataset [22] and self-collected videos. Besides, we also construct a real-world test set with dual reversed distortion inputs captured by a custom-made co-axial imaging system. Although similar concept of dual-RS [1] (stereo) setup and time field [20] (2d) were proposed separately by previous works, we successfully combine and upgrade them to propose a simple yet robust architecture to solve the joint RS correction and interpolation (RS temporal super-resolution [7]) problem. The contributions of this work can be summarized as follows: 1) This is the first work that can extract video clips from distorted image in dynamic scenes. Besides, our solution can overcome the generalization problem caused by distinct readout settings. 2) We propose a novel



(a) VFI

(b) RSC

(c) Dual-RSC

(d) BFI

(e) RSCI

(f) Dual-RSCI

Fig. 2: **Comparison of different tasks.** The first row represents the input and the second row represents the output of each task. The x-axis and y-axis represent the time and the row location of the captured or generated image, respectively. (a) Video frame interpolation task (VFI). (b) RS correction task using neighboring frames (RSC). (c) RS correction task using dual frames with reversed RS distortion (Dual-RSC). (d) Blurry frame interpolation task (BFI). (e) Joint RS correction and interpolation task using neighboring frames (RSCI). (f) Joint RS correction and interpolation task using dual frames with reversed RS distortion (Dual-RSCI).

end-to-end network architecture (IFED) that can iteratively estimate the accurate dual optical flow cube using pre-defined time cube and efficiently merges the symmetric information of the dual RS inputs for latent GS frame extraction. 3) Extensive experimental results demonstrate the superior accuracy and robustness of IFED against the state-of-the-art both on synthetic dataset and real-world data.

2 Related Works

In this section, we briefly review the closely related research on video frame interpolation and rolling shutter correction.

2.1 Video Frame Interpolation

Most existing solutions to VFI utilize optical flows to predict intermediate frames of captured images. These methods warp the input frames in a forward or backward manner based on the flow estimated by off-the-shelf networks, such as PWCNet [32], FlowNet [6,12], and RAFT [33]. The warped frame is then refined by convolutional neural networks (CNNs) to obtain better visual quality. For example, SuperSlomo [13] uses a linear combination of two bi-directional flows from an off-the-shelf network for intermediate flow estimation and performs backward warping to infer latent frames. DAIN [3] further improves the intermediate flow estimation by employing a depth-aware flow projection layer. Recently, RIFE [11] achieves high-quality and real-time frame interpolation with an efficient flow network and a leakage distillation loss for direct flow estimation. In contrast to backward warping, Niklaus *et al.* [23] focuses on forward warping interpolation by proposing Softmax splatting to address the conflict of pixels mapped to the same target location. On the other hand, some recent works [4,17] achieve good results using flow-free methods. For example, CAIN [4] employs the PixelShuffle operation with channel attention to replace the flow computation module, while FLAVR [17] utilizes 3D space-time convolutions instead to improve efficiency and performance on non-linear motion and complex occlusions.

VFI includes a branch task, called blurry frame interpolation [15,28,14,31], which is analogous to our target problem. In this task, a blurry image is a temporal average of sharp frames at multiple instances. The goal is to deblur the video frame and conduct interpolation, as illustrated in Fig. 2d. Jin *et al.* [15] proposed a deep learning scheme to extract a video clip from a single motion-blurred image. For a better temporal smoothness in the output high-frame-rate video, Jin *et al.* [14] further proposed a two-step scheme consisting of a deblurring network and an interpolation network. Instead of using a pre-deblurring procedure, BIN [31] presents a multi-scale pyramid and recurrent architecture to reduce motion blur and upsample the frame rate simultaneously. Other works [25,18] utilize additional information from event cameras to bring a blurry frame alive with a high frame rate.

Existing VFI methods ignore the distortions in videos captured by RS cameras. In our work, instead of considering RS distortion as a nuisance, we leverage the information embedded in it to retrieve a sequence of GS frames.

2.2 Rolling Shutter Correction

RS correction itself is also a highly ill-posed and challenging problem. Classical approaches [9,2,24] work under some assumptions, such as a static scene and restricted camera motion (*e.g.*, pure rotations and in-plane translations). Consecutive frames are commonly used as inputs to estimate camera motion for distortion correction. Grundmann *et al.* [10] models the motion between two neighboring frames as a mixture of homography matrices. Zhuang *et al.* [37] develops a modified differential SfM algorithm for estimating the relative pose between consecutive RS frames, which in turn recovers a dense depth map for RS-aware warping image rectification. Vasu *et al.* [34] sequentially estimates both camera motion and the structure of the 3D scene that accounts for the RS distortion, and then infers the latent image by performing depth and occlusion-aware rectification. Rengarajan *et al.* [30] corrects the RS image according to the rule of "straight-lines-must-remain-straight". Purkait *et al.*[27] assumes that the captured 3D scene obeys the Manhattan world assumption and corrects the distortion by jointly aligning vanishing directions.

In recent years, learning-based approaches have been proposed to address RS correction in more complex cases. Rengarajan *et al.* [29] builds a CNN architecture with long rectangular convolutional kernels to estimate the camera motion

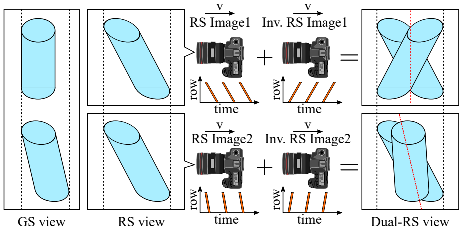


Fig. 3: **RS correction ambiguity.**

Table 1: **Details of RS-GOPRO**.

	train	validation	test
sequences	50	13	13
RS images	3554 (×2)	945 (×2)	966 (×2)
GS images	31986	8505	8694
resolution		960×540	
row exposure		1.0 ms	
row readout		87 μs	

from a single image for RS correction. Zhuang *et al.* [38] uses two independent networks to predict a dense depth map and camera motion from a single RS image, implementing RS correction as post-processing. Liu *et al.* [20] proposes a DeepUnrollNet to realize end-to-end RS correction with a differentiable forward warping block. SUNet [8] utilizes a symmetric consistency constraint of two consecutive frames to achieve state-of-the-art performance.

The most relevant research to ours are [7], [1] and the previously mentioned [20]. [7] proposed the first learning-based solution (RSSR) for latent GS video extraction from two consecutive RS images. On the other hand, [1] proposed a stereo dual-RS setup for RS correction task that infers an undistorted GS frame based on the geometric constraints among dual RS reversely distorted images. However, to the best of our knowledge, there are no methods able to achieve RS temporal super-resolution in dynamic scenes. Geometric constraints of [7] and [1] are limited to static scenes. Besides, current learning-based methods including [20,7] suffer from the inherent ambiguity of consecutive setup. We discover the merit of dual-RS to overcome distinct readout setups, which is not mentioned in [1], and we upgrade the velocity field from [20] to first time realize RS temporal SR in dynamic scenes.

3 Joint RS Correction and Interpolation

In this section, we first formulate the joint RS correction and interpolation problem. Then, we introduce the datasets for validation and comparison.

3.1 Problem Formulation

An RS camera encodes temporal visual information in an image similar to a high-frame-rate GS camera that samples the scene rapidly but only takes one row of the scene each time. In our case, we do not consider the presence of blur. Formally, given an RS video ($\{I_r^{(t)}\}$) and a GS video ($\{I_g^{(t)}\}$), we can express each row (i) in an RS image ($I_r^{(t)}[i]$) in terms of its corresponding GS image ($I_g^{(t)}[i]$) through the following equation:

$$I_r^{(t)}[i] = I_g^{(t+(i-M/2)t_r)}[i], \tag{1}$$

where t_r denotes the readout time for each RS row; M denotes the total number of rows in the image; $t + (i - M/2)t_r$ is the time instant of scanning the i^{th} row; and $I_g^{(t+(i-M/2)t_r)}[i]$ is the portion of the GS image that will appear in the RS image. Note that we define the time t of an RS image $I_r^{(t)}$ as the midpoint of its exposure period (*i.e.*, each RS image is captured from t_s to t_e, where $t_s = t - t_r M/2$ and $t_e = t + t_r M/2$).

The objective of the joint RS correction and interpolation is to extract a sequence of undistorted GS images ($\left\{I_g^{(t)}, t \in [t_s, t_e]\right\}$) from the RS images. Directly feeding an RS image ($I_r^{(t)}$) into a network $\mathcal{F}\left(I_r^{(t)}; \Theta\right)$, parameterized by the weight Θ, to extract a sequence of GS images is infeasible without strong restrictions such as static scenes and known camera motions. A straightforward approach is to use temporal constraints from neighboring frames, such that the input is a concatenation of neighboring frames as $I_{inp}^{(t)} = \left\{I_r^{(t-1/f)}, I_r^{(t)}\right\}$, where f denotes the video frame rate. This is the case of RSSR [7], which can easily overfit the readout setting of the training data. Theoretically, the generic RSC problem cannot be solved by using only consecutive frames. We show a common ambiguity of consecutive frames setup, using a toy example in Fig. 3. Suppose there are two similar cylinders, one of them is tilted, as shown in GS view. Then, two RS cameras moving horizontally at the same speed v but with different readout time setups can produce the same RS view, *i.e.*, a short readout time RS camera for the tilted cylinder and a long readout time RS camera for the vertical cylinder. Therefore, the models based on consecutive frames are biased to the training dataset. Although these models can correct RS images, they do not know how much correction is correct facing data beyond the dataset. Instead, we introduce another constraint setting that utilizes intra-frame spatial constraints of dual images taken simultaneously but with reversed distortion captured by top-to-bottom (t2b) and bottom-to-top (b2t) scanning. Formally, the optimization process is described as:

$$\widehat{\Theta} = \arg\min_{\Theta} \left| \left\{I_g^{(t)}, t \in [t_s, t_e]\right\} - \mathcal{F}\left(I_{t2b}^{(t)}, I_{b2t}^{(t)}; \Theta\right) \right|, \tag{2}$$

where $\widehat{\Theta}$ are optimized parameters for the joint task. $I_{t2b}^{(t)}$ denotes the t2b RS frame at time t, while $I_{b2t}^{(t)}$ denotes the b2t RS frame at the same time. We find that the dual-RS setup can avoid ambiguity because the correct correction pose can be estimated based on the symmetry, as shown in the dual-RS view.

3.2 Evaluation Datasets

Synthetic Dataset. For the pure RS correction task, the Fastec-RS [20] dataset uses a camera mounted on a ground vehicle to capture high-fps videos with only horizontal motion. Then, RS images are synthesized by sequentially copying a row of pixels from consecutive high-fps GS frames. We synthesized a dataset for the joint RS correction and interpolation task in a similar way, but with more motion patterns and multiple ground truths for one input. High-fps GS cameras with sufficient frame rate to synthesize RS-GS pairs are expensive and cumbersome to operate. Thus, we chose a GoPro (a specialized sports camera) as a trade-off. Empirically, the GoPro's tiny RS effect causes negligible impact on the learning process of our task. Specifically, we utilize the high-fps (240 fps) videos from the publicly available GOPRO [22] dataset and self-collected videos using a GoPro HERO9 to synthesize the dataset, which we refer to as RS-GOPRO. We first interpolated the original GS videos to 15 360 fps by using an off-the-shelf VFI method (RIFE [11]), and then followed the pipeline of [20] to synthesize RS videos. RS-GOPRO includes more complex urban scenes (*e.g.*, streets and building interiors) and more motion patterns, including object-only motion, camera-only motion, and joint motion. We created train/validation/test sets (50, 13, and 13 sequences) by randomly splitting the videos while avoiding images of a video from being assigned into different sets. Regarding input and target pairs, there are two kinds of input RS images which have reversed distortion, and nine consecutive GS frames are taken as ground truth for the extracted frame sequence. The image resolution is 960×540. The readout time for each row is fixed as 87 µs. Please see the details of RS-GOPRO in Table 1.

Real-world Test Set. Inspired by [36] and [1], we built a dual-RS image acquisition system using a beam-splitter and two RS cameras that are upside down from each other to collect real-world data for validation. The readout setting of the proposed dual-RS system can be changed by replacing the type of RS camera (*e.g.*, FL3-U3-13S2C, BFS-U3-63S4C). Please see details of our acquisition system in supplementary materials. We collect samples of various motion patterns, such as camera-only motion, object-only motion like moving cars and a rotating fan, and mixed motion. Each sample includes two RS distorted images with reversed distortion but without a corresponding ground truth sequence.

4 Methodology

We present the proposed architecture and implementation details in this section.

4.1 Pipeline of IFED

The proposed IFED model utilizes an architecture inherited from existing successful VFI methods [13,11], including a branch to estimate the optical flow for backward warping and an encoder-decoder branch to refine the output (see Fig. 4). However, directly estimating optical flow from the latent GS image to the input RS image is challenging due to the intra-frame temporal inconsistency of an RS image. The optical flow from GS to RS is dependent on two variables: the time difference and relative velocity of motion. As we already know the scanning mechanism of the RS camera, we are able to obtain the time difference between the input RS image and the target GS image. Thus, we propose a dual time cube as an RS prior to decouple this problem, and let the model regress the dual velocity cube to indirectly estimate the corresponding dual optical flow

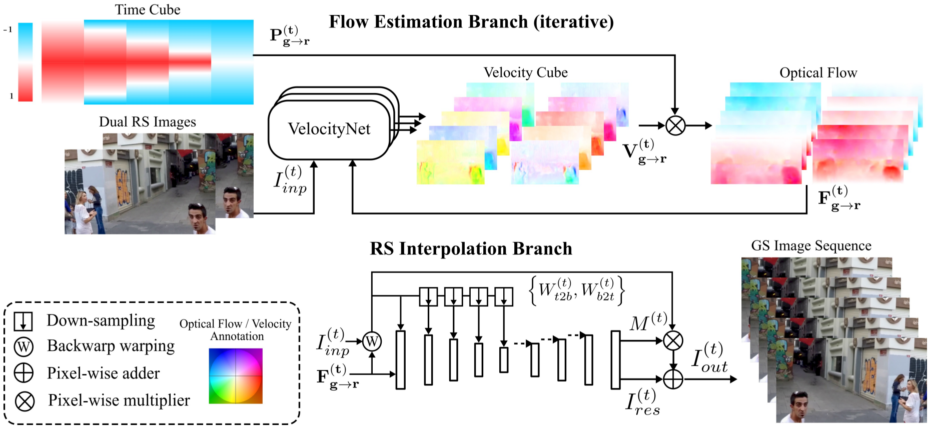

Time Cube
-1
1
$\mathbf{P}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}}$
Flow Estimation Branch (iterative)
Velocity Cube
Optical Flow
Dual RS Images
VelocityNet
$I^{(t)}_{inp}$
$\mathbf{V}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}}$
$\mathbf{F}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}}$
RS Interpolation Branch
GS Image Sequence
$\left\{W^{(t)}_{t2b}, W^{(t)}_{b2t}\right\}$
Down-sampling
Backwarp warping
Pixel-wise adder
Pixel-wise multiplier
Optical Flow / Velocity Annotation
$I^{(t)}_{inp}$
$\mathbf{F}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}}$
W
$M^{(t)}$
$I^{(t)}_{out}$
$I^{(t)}_{res}$

Fig. 4: **Network architecture of IFED.** Note that the color annotation of the dual RS time cube is different from optical flow and velocity. It represents the relative time gap between each row and the time instance of the latent GS image. This architecture first utilizes dual RS along with time cube to iteratively estimate velocity cube for better optical flow learning. Then, the warped dual frames are combined together as complementary information through the mask and residual cube learned from the encoder-decoder network to make inferences for the underlying GS image sequences.

cube. The number of time instances per dual cube is twice the number of extracted GS frames. These time instances are sampled uniformly from the entire RS exposure time (*e.g.*, Fig. 4 shows the extraction of 5 GS frames). There is an implicit assumption that the velocity field of each row of the extracted frame is constant. Considering the short exposure time of the actual RS image and the short percentage of time corresponding to the extracted GS frames, this assumption can be basically satisfied in most scenarios. Besides, the dual warped features can be further adjusted and merged by the interpolation branch, which enables our method to handle the challenging cases of the spinning fans and wheels with row-wise non-uniform velocity.

Specifically, assuming our target latent sequence has N images, the target optical flow cube for one RS image can be expressed as follows:

$$\mathbf{F}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}} = \left\{F^{(t_n)}_{g\to r}\right\}, n \in \{1, \cdots, N\}, \tag{3}$$

where $t_n = t - t_r M\left(\frac{1}{2} - \frac{n}{N}\right)$ and $F^{(t_n)}_{g\to r}$ denotes the optical flow from the GS image at time t_n to the distorted RS input $I^{(t)}_r$. Regarding the time cube, the values at row m of the time map $P^{(t)}_{g\to r}$ are given by:

$$P^{(t_n)}_{g\to r}[m] = \frac{m-1}{M-1} - \frac{n-1}{N-1}, m \in [1..M], n \in [1..N]. \tag{4}$$

Then, the RS time cube $\mathbf{P}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}} = \left\{P^{(t_n)}_{g\to r}\right\}$ can be expressed in the same format as $\mathbf{F}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}}$. To obtain the optical flow cube, we need the network to generate a velocity cube $\mathbf{V}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{r}} = \left\{V^{(t_n)}_{g\to r}\right\}$ and multiply it with the RS time cube as follows:

$$\left\{F^{(t_n)}_{g\to r}\right\} = \left\{P^{(t_n)}_{g\to r} V^{(t_n)}_{g\to r}\right\}. \tag{5}$$

Our flow branch uses several independent subnetworks (VelocityNet) to iteratively take dual RS images $I^{(t)}_{inp}$ and previously estimated dual optical flow cube as inputs for dual velocity cube $\mathbf{V}^{(\mathbf{t})}_{\mathbf{g}\to\mathbf{t2b}}$ estimation. The input scale (resolution) of the subnetwork are scaled sequentially in an iterative order following a coarse-to-fine manner (adjusted by bilinear interpolation). These sub-networks share the same structure, starting with a warping of the inputs, followed by a series of 2d convolutional layers. The initial scale velocity cube estimation is realized without the estimated optical flow cube. This branch is shown in the upper part of Fig. 4.

After obtaining the optical flow cube, we can generate a series of warped features and the warped dual RS images $W^{(t)} = \left\{W^{(t)}_{b2t}, W^{(t)}_{t2b}\right\}$ as multi-scale inputs to an encoder-encoder network with skip connections for merging results. Specifically, a residual cube $I^{(t)}_{res}$ and a dual mask cube $M^{(t)}$ are generated to produce the final frame sequence (See the bottom part of Fig. 4) as follows:

$$I^{(t)}_{out} = I^{(t)}_{res} + M^{(t)} W^{(t)}_{t2b} + \left(1 - M^{(t)}\right) W^{(t)}_{b2t}. \tag{6}$$

4.2 Implementation Details

We implement the method using PyTorch [26]. There are three 4 sub-networks in the flow network branch for velocity cube learning, each with eight 3 × 3 convolutional layers. The inputs scale is gradually adjusted from 1/8 to original size as the channel size is reduced. The network is trained in 500 epochs. The batch size and learning rate are equal to 8 and 1×10^{-4} separately. AdamW [21] is used to optimize the weights with a cosine annealing scheduler. The learning rate is gradually reduced to 1×10^{-8} throughout the whole process. 256 × 256 cropping is applied for both dual RS images and the time cube. Because the relative time difference between the same row of adjacent crops is constant, training with cropping does not affect the full frame inference. More details of the sub-networks and cropping are in supplementary materials. The loss function to train the model is given by:

$$\mathcal{L} = \mathcal{L}_{char} + \lambda_p \mathcal{L}_{perc} + \lambda_v \mathcal{L}_{var}, \tag{7}$$

where $\mathcal{L}_{char}$ and $\mathcal{L}_{perc}$ denote the Charbonnier loss and perceptual loss [16] for the extracted frame sequence; while $\mathcal{L}_{var}$ denotes the total variation loss for the estimated flows, to smooth the warping. λ_p and λ_v are both set to 0.1.

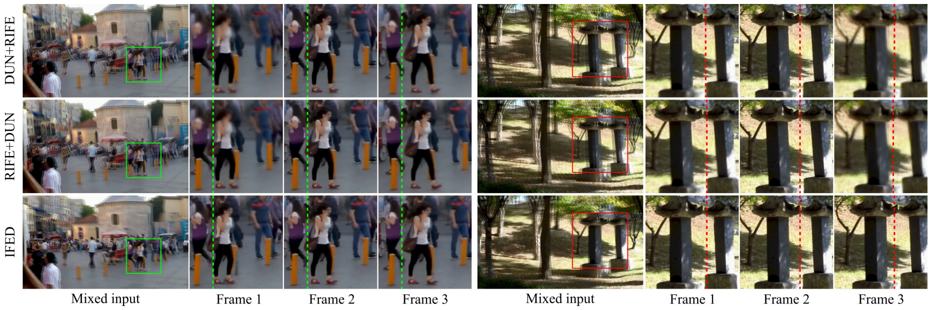


Fig. 5: **Visual results on RS-GOPRO.** Zoom-in results are shown chronologically on the right side of the mixed input. IFED restores the smooth moving sequence with clearer details while cascaded scheme introduced unclear artifacts.

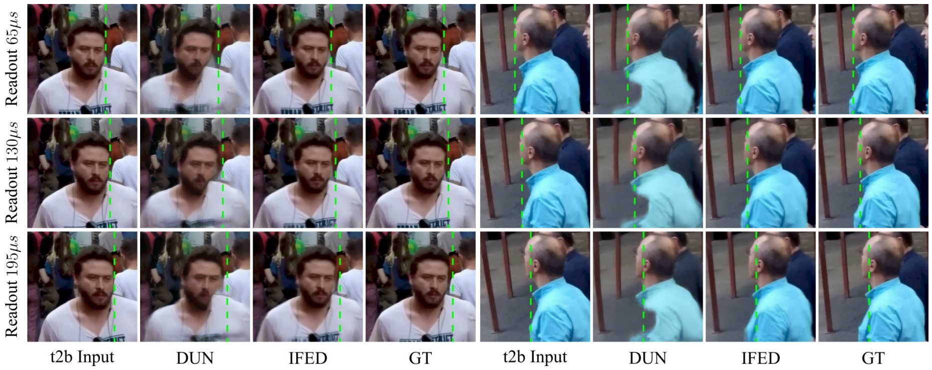


Fig. 6: **Generalization ability on distinct readout time settings.** Both our IFED and DUN [20] are trained on fixed readout setting, while IFED can successfully generalize to different readout settings from 65 μs to 195 μs.

5 Experimental Results

In this section, we first present comparison experiments on the synthesized dataset RS-GOPRO in Sec. 5.1. Next, we show the generality of our method on real-world data in Sec. 5.2. Finally, we present the ablation study in Sec. 5.3. Please see more additional experimental results in our appendix.

5.1 Results on Synthetic Dataset

We implemented cascade schemes with RSC model DUN (DeepUnrollNet [20]) and a VFI model RIFE [11] using adjacent frames as inputs. Both orderings were examined, *i.e.*, DUN+RIFE and RIFE+DUN ((b)+(a) and (a)+(b) in Fig. 2). We retrained DUN and RIFE on our dataset for extracting 1, 3, 5, and 9 frames for fair comparison. Quantitative results are shown in Table 2. Over the different

Table 2: **Quantitative results on RS-GOPRO.** f# denotes # of frames extracted from the input RS images.

	PSNR ↑	SSIM ↑	LPIPS ↓
DUN (f1)	26.37	0.836	0.058
DUN + RIFE (f3)	25.38	0.788	0.159
DUN + RIFE (f5)	25.45	0.798	0.111
DUN + RIFE (f9)	25.31	0.795	0.102
RIFE + DUN (f3)	23.05	0.719	0.124
RIFE + DUN (f5)	22.28	0.692	0.118
RIFE + DUN (f9)	21.88	0.677	0.113
IFED (f1)	32.07	0.934	0.028
IFED (f3)	28.48	0.872	0.058
IFED (f5)	29.79	0.897	0.049
IFED (f9)	30.34	0.910	0.046

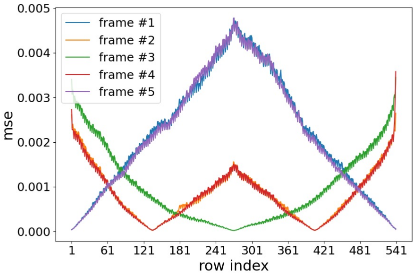


Fig. 7: **Image mean squared errors based on row number in the case of IFED (f5).**

extracted frame settings, IFED shows superiority over the cascade schemes. The average performance of IFED is worst when the number of extracted frames is 3. Our interpretation is that the task degrades to a relatively easy RS correction task when the number of extracted frames is 1, while the greater continuity between extracted frames is better for convergence when the number of extracted frames is greater than 3. Qualitative results are shown in Fig. 5. With the cascade schemes, the details are blurry, while ours are much clearer.

To verify the generalization on distinct readout settings, we synthesized RS images with distinct readout settings such as 65μs, 130μs, and 195μs. As illustrated in Fig. 6, both our IFED and DUN [20] are trained on fixed readout setting, while our IFED can successfully generalize to different readout settings without introducing artifacts and undesired distortions.

Besides, an row-wise image error analysis (f5) is shown in Fig. 7 in terms of MSE. It indicates that the performance of a given row index depends on the minimum time (the smaller the better) between the row of that extracted GS frame and the corresponding rows of dual RS frames.

5.2 Results on Real-world Data

We also compare our method to the only existing work on extracting a GS sequence from RS images RSSR [7] and the only work for dual reversed RS image correction [1]. Since the source codes of these two works are not publicly available, we sent our real-world samples from different type of cameras to the authors for testing. The comparison results with RSSR [7] are shown in Fig. 8. RSSR cannot generalize to either the case of camera-only motion (the left example) or the case of object-only motion (the right example), while IFED is robust to different motion patterns. The visual results of IFED and [1] are illustrated in Fig. 9. It demonstrates the ability of IFED to go beyond [1] by being able to extract a sequence of GS images in dynamic scenes, rather than just a single GS

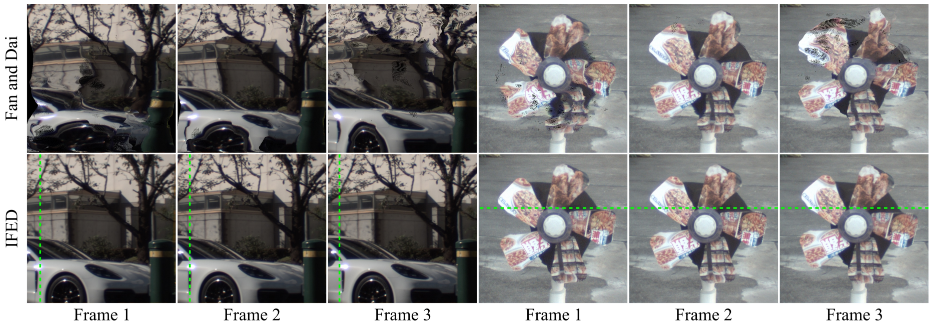


Fig. 8: **Comparison with Fan and Dai [7] on real data.** Our results (the 2^{nd} row) are significantly better than Fan and Dai's for objects under both horizontal and rotational movements. Please refer to our supplementary videos.

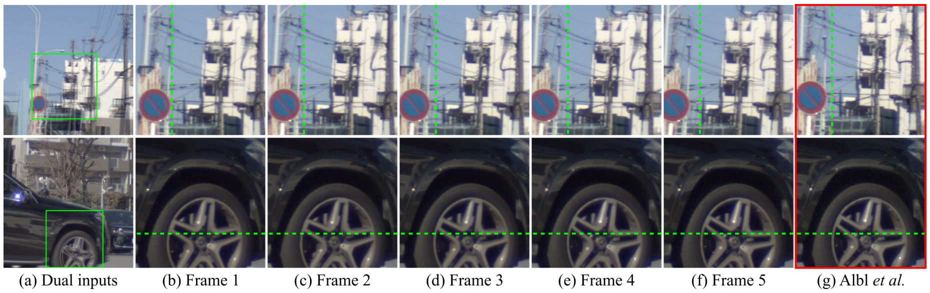
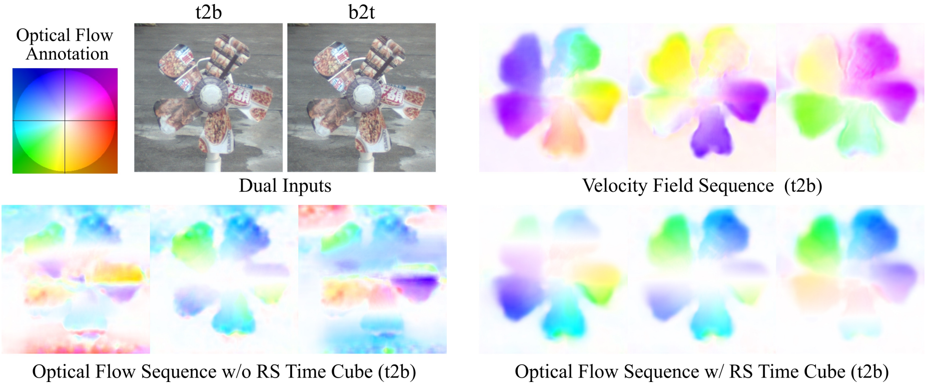


Fig. 9: **Comparison with Albl *et al.* [1] on real data.** Both our method and Albl *et al.*'s use the same dual inputs (the 1^{st} column). Our method brings the dual input alive by creating a sequence of images (Frame 1 $\sim$ 5), compared to one static image from Albl *et al.*'s.

image in static scenes, from a dual reversed RS image. More results of IFED on the real dataset can be found in the supplementary materials.

5.3 Ablation Study

Table 3 shows the results of our ablation study on the RS time cube prior. It shows that IFED without the prior generally leads to worse results, and the difference increases with a larger number of frames. Note that when the number of extracted frames equals 1, IFED *w/o pr* can achieve better performance. The reason is that the task simply becomes the RSC task in this case, and the model can directly learn a precise flow for the middle time instance using dual RS inputs. When the number of extracted frames increases, the model needs the time cube to serve as an "anchor" for each time instance to improve the temporal consistency of the learned flow. We show visualizations of the flow and velocity cube with RS time cube prior and the flow cube without the prior in



Dual Inputs

Velocity Field Sequence (t2b)

Optical Flow Sequence w/o RS Time Cube (t2b)

Optical Flow Sequence w/ RS Time Cube (t2b)

Fig. 10: **Visualization of optical flow and velocity cube.** Equally with dual RS frames as input, using RS time cube prior to learn velocity cube can reduce the difficulty of optical flow learning and ultimately improve the flow quality.

Fig. 10. The flow sequence estimated without the RS time cube prior exhibits poor quality and consistency in time.

6 Conclusions

In this paper, we addressed a challenging task of restoring consecutive distortion-free frames from RS distorted images in dynamic scenes. We designed an end-to-end deep neural network IFED for the dual-RS setup, which has the advantages of being able to model dynamic scenes and not being affected by distinct readout times. The proposed dual RS time cube for velocity cube learning improves performance by avoiding direct flow estimation from the GS image to the RS image. Compared to the cascade scheme with existing VFI and RSC models as well as RSSR which takes temporally adjacent frames as inputs to do the same task, our IFED shows more impressive accuracy and robustness for both synthetic data and real-world data with different motion patterns.

Table 3: **Ablation study for the prior.** *w/o pr* denotes "without RS time cube prior".

	PSNR ↑	SSIM ↑	LPIPS ↓
	Refer to IFED (f#)		
IFED *w/o pr* (f1)	+0.50	+0.006	-0.003
IFED *w/o pr* (f3)	-0.40	-0.008	0.000
IFED *w/o pr* (f5)	-0.50	-0.009	+0.001
IFED *w/o pr* (f9)	-0.70	-0.012	+0.001

Acknowledgement

This work was supported by D-CORE Grant from Microsoft Research Asia, JSPS KAKENHI Grant Numbers 22H00529, and 20H05951, and JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2108.

References

1. Albl, C., Kukelova, Z., Larsson, V., Polic, M., Pajdla, T., Schindler, K.: From two rolling shutters to one global shutter. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2505–2513 (2020)
2. Baker, S., Bennett, E., Kang, S.B., Szeliski, R.: Removing rolling shutter wobble. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2392–2399. IEEE (2010)
3. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3703–3712 (2019)
4. Choi, M., Kim, H., Han, B., Xu, N., Lee, K.M.: Channel attention is all you need for video frame interpolation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10663–10671 (2020)
5. Dai, Y., Li, H., Kneip, L.: Rolling shutter camera relative pose: Generalized epipolar geometry. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4132–4140 (2016)
6. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
7. Fan, B., Dai, Y.: Inverting a rolling shutter camera: bring rolling shutter images to high framerate global shutter video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4228–4237 (2021)
8. Fan, B., Dai, Y., He, M.: Sunet: Symmetric undistortion network for rolling shutter correction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4541–4550 (2021)
9. Forssén, P.E., Ringaby, E.: Rectifying rolling shutter video from hand-held devices. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 507–514. IEEE (2010)
10. Grundmann, M., Kwatra, V., Castro, D., Essa, I.: Calibration-free rolling shutter removal. In: 2012 IEEE international conference on computational photography (ICCP). pp. 1–8. IEEE (2012)
11. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Rife: Real-time intermediate flow estimation for video frame interpolation. arXiv preprint arXiv:2011.06294 (2020)
12. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017)
13. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9000–9008 (2018)
14. Jin, M., Hu, Z., Favaro, P.: Learning to extract flawless slow motion from blurry videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8112–8121 (2019)
15. Jin, M., Meishvili, G., Favaro, P.: Learning to extract a video sequence from a single motion-blurred image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6334–6342 (2018)

16. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
17. Kalluri, T., Pathak, D., Chandraker, M., Tran, D.: Flavr: Flow-agnostic video representations for fast frame interpolation. arXiv preprint arXiv:2012.08512 (2020)
18. Lin, S., Zhang, J., Pan, J., Jiang, Z., Zou, D., Wang, Y., Chen, J., Ren, J.: Learning event-driven video deblurring and interpolation. In: European Conference on Computer Vision. vol. 3 (2020)
19. Litwiller, D.: Ccd vs. cmos. Photonics spectra **35**(1), 154–158 (2001)
20. Liu, P., Cui, Z., Larsson, V., Pollefeys, M.: Deep shutter unrolling network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5941–5949 (2020)
21. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
22. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3883–3891 (2017)
23. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5437–5446 (2020)
24. Oth, L., Furgale, P., Kneip, L., Siegwart, R.: Rolling shutter camera calibration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1360–1367 (2013)
25. Pan, L., Scheerlinck, C., Yu, X., Hartley, R., Liu, M., Dai, Y.: Bringing a blurry frame alive at high frame-rate with an event camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6820–6829 (2019)
26. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 (2019)
27. Purkait, P., Zach, C., Leonardis, A.: Rolling shutter correction in manhattan world. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 882–890 (2017)
28. Purohit, K., Shah, A., Rajagopalan, A.: Bringing alive blurred moments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6830–6839 (2019)
29. Rengarajan, V., Balaji, Y., Rajagopalan, A.: Unrolling the shutter: Cnn to correct motion distortions. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition. pp. 2291–2299 (2017)
30. Rengarajan, V., Rajagopalan, A.N., Aravind, R.: From bows to arrows: Rolling shutter rectification of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2773–2781 (2016)
31. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z.: Blurry video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5114–5123 (2020)
32. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
33. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision. pp. 402–419. Springer (2020)

34. Vasu, S., Rajagopalan, A., et al.: Occlusion-aware rolling shutter rectification of 3d scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 636–645 (2018)
35. Yang, X., Xiang, W., Zeng, H., Zhang, L.: Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4781–4790 (2021)
36. Zhong, Z., Gao, Y., Zheng, Y., Zheng, B.: Efficient spatio-temporal recurrent neural network for video deblurring. In: European Conference on Computer Vision. pp. 191–207. Springer (2020)
37. Zhuang, B., Cheong, L.F., Hee Lee, G.: Rolling-shutter-aware differential sfm and image rectification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 948–956 (2017)
38. Zhuang, B., Tran, Q.H., Ji, P., Cheong, L.F., Chandraker, M.: Learning structure-and-motion-aware rolling shutter correction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4551–4560 (2019)