SUNet: Symmetric Undistortion Network for Rolling Shutter Correction

Bin Fan Yuchao Dai* Mingyi He
School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China
binfan@mail.nwpu.edu.cn, daiyuchao@nwpu.edu.cn, myhe@nwpu.edu.cn

Abstract

The vast majority of modern consumer-grade cameras employ a rolling shutter mechanism, leading to image distortions if the camera moves during image acquisition. In this paper, we present a novel deep network to solve the generic rolling shutter correction problem with two consecutive frames. Our pipeline is symmetrically designed to predict the global shutter image corresponding to the intermediate time of these two frames, which is difficult for existing methods because it corresponds to a camera pose that differs most from the two frames. First, two time-symmetric dense undistortion flows are estimated by using well-established principles: pyramidal construction, warping, and cost volume processing. Then, both rolling shutter images are warped into a common global shutter one in the feature space, respectively. Finally, a symmetric consistency constraint is constructed in the image decoder to effectively aggregate the contextual cues of two rolling shutter images, thereby recovering the high-quality global shutter image. Extensive experiments with both synthetic and real data from public benchmarks demonstrate the superiority of our proposed approach over the state-of-the-art methods.

1. Introduction

Popular low-budget commercial cameras are generally built upon CMOS sensors due to their low cost and simplicity in design. Most CMOS cameras employ a rolling shutter (RS) mechanism. Different from the global shutter (GS) camera that exposes all pixels at the same time, the RS camera is exposed in a row-wise manner from top to bottom. Thus, the images and videos captured by a moving RS camera will have the RS effect (*e.g.*, stretch, wobble). We provide an illustration in Fig. 2 while assuming the camera has a frame time of τ. The high dynamic sampling characteristics of the RS mechanism are regarded as an advantage rather than a disadvantage [1], but simply ignoring the RS effects in 3D geometric vision applications may lead to erroneous, undesirable and distorted results [5, 18, 3, 9].

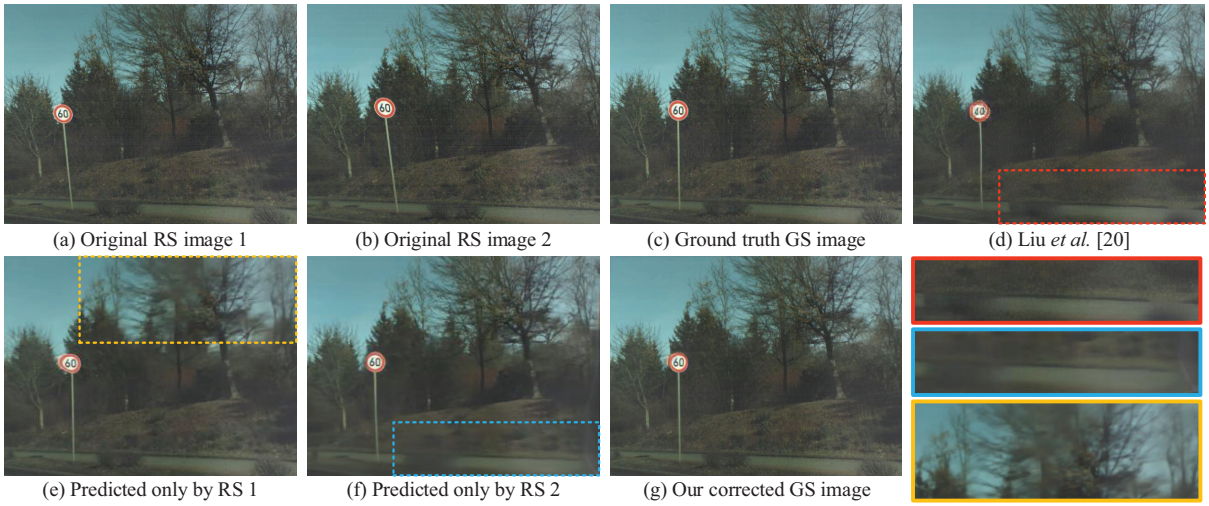
*Corresponding author

To mitigate or eliminate the RS effect which is increasingly becoming a nuisance in photography, a well-known RS correction problem has therefore attracted more and more attention [36, 38, 20]. This challenging problem aims at recovering the GS image corresponding to the camera pose at a specific exposure time (*e.g.*, 0 or τ in Fig. 2).

Existing works on RS correction generally fall into one of two categories: single-frame-based or multi-frame-based. It is an ill-posed problem in the case of single-frame-based RS correction, where additional prior assumptions need to be formulated explicitly [26, 17, 24, 23] or implicitly [25, 38]. Following the previous studies [36, 37, 20, 8] in reducing the ill-posedness, we also focus on the well-posed generic RS correction problem from two consecutive frames of a video. In particular, since RS cameras are usually time-synchronized with other sensors (*e.g.*, GS camera, IMU, etc.) in hardware by referring to the first scanline time [29, 13], we deal with a corresponding challenge of correcting the RS images to the GS image corresponding to the exposure time of the first scanline of the second frame (*i.e.*, the intermediate time τ of these two frames). This is of both theoretical interest and great practical importance.

To this end, classical two-frame-based methods [36, 37] heavily rely on specific RS motion models and require nontrivial iterative optimizations, which limits their use in time-constrained applications. Although the state-of-the-art learning-based method [20] has achieved promising results in recovering the GS image corresponding to the middle time $\frac{3\tau}{2}$ of the second frame, it is prone to fail to predict a plausible GS image at time τ, where many image texture details cannot be well restored (*c.f.* Fig. 1(d)). Since the GS image at time $\frac{3\tau}{2}$ and the second RS image have close content, their asymmetric network only exploits the limited information of the second RS image, but ignores the contextual aggregation of both RS frames.

Recently, various image-to-image translation problems have been tackled with the deep learning pipelines (*e.g.*, image correction [20], optical flow estimation [32], depth estimation [22], video deblurring [31], and image super-resolution [19], etc.). However, it still poses a challenge to learn to produce a satisfactory GS image corresponding to



(a) Original RS image 1 (b) Original RS image 2 (c) Ground truth GS image (d) Liu *et al.* [20]

(e) Predicted only by RS 1 (f) Predicted only by RS 2 (g) Our corrected GS image

Figure 1. An example of RS correction from two consecutive frames. (a-b) Input two consecutive RS images; (c) Ground truth GS image corresponding to the middle time of two consecutive frames; (d) GS image predicted by the state-of-the-art method [20]; (e-g) The forward, backward, and target GS images predicted from only the first RS image, only the second RS image, and the combined two RS images (our pipeline), respectively. The first and second RS images have different contributions to the upper and lower regions of the corresponding GS image which can be seen between yellow and blue boxes. Our proposed approach makes use of this property for RS correction.

the intermediate time τ of two consecutive frames. Its complexities mainly lie in the following facts: **1**) Different from the local neighbors that can provide sufficient description in general image-to-image translation problem, a pixel of the target GS image may not be in the neighboring pixel of its corresponding RS images, depending on the type of motion, the 3D structure, and the scanline time; **2**) We observe in Fig. 1(e)&(f) that the first and second RS images contribute greatly to the lower and upper parts of the corresponding time-centered GS image, respectively. This is expected because they are closer in time and thus share more similar camera poses. Consequently, the centered exposure time τ corresponds to the most unique camera pose that deviates greatly from both RS frames, such that only the cues from a single RS image may result in the lack of details of the corrected GS image. Therefore, motivated by these two insights, we propose a novel symmetric undistortion network architecture to rectify the geometric inaccuracies induced by the RS effect from two consecutive RS images. The network can benefit from the overall exploitation and aggregation of contextual information, where the time-centered GS image can be reconstructed by using an adaptive fusion scheme, as illustrated in Fig. 1(g).

Our network takes two consecutive RS images as input and predicts the corrected GS image corresponding to the intermediate time of these two frames. To essentially exploit and merge contextual information across both RS images, our pipeline is symmetrically constructed. It consists of two main processes: a PWC (pyramid, warping, and context-aware cost volume)-based undistortion flow estimator and a time-centered GS image decoder. The success of the classic PWC-Net framework [32] in aggregating multi-scale context information inspires the construction of our undistortion flow estimator, which is to estimate the pixel-wise undistortion flows of the first and second RS images. Note that the *context-aware cost volume* we construct can effectively promote contextual consistency at different scales. Then, the pixel-wise undistortion flows are used to warp the learned image features to their corresponding GS counterparts. Finally, we develop a time-centered GS image decoder to further align the contextual cues of their warped features and convert the aggregated feature representations to the target GS image in a robust way. Our symmetric undistortion network (SUNet) can be trained end-to-end and solely uses the ground truth GS image for supervision. Furthermore, it can also inpaint the occluded regions from the learned image priors, resulting in a high-quality GS image as shown in Fig. 1(g). Experimental results on two benchmark datasets demonstrate that our approach is superior to the state-of-the-art methods in removing the RS artifacts.

In summary, our main contributions are:

- We propose an efficient end-to-end symmetric RS undistortion network to solve the generic RS correction problem with two consecutive frames.
- Our context-aware cost volume together with the symmetric consistency constraint can aggregate the contextual cues of two input RS images effectively.
- Extensive experiments show that our approach performs favorably against the state-of-the-art methods in both GS image restoration and inference efficiency.

2. Related Work

We categorize the relevant works into classical model based and deep learning based RS correction methods.

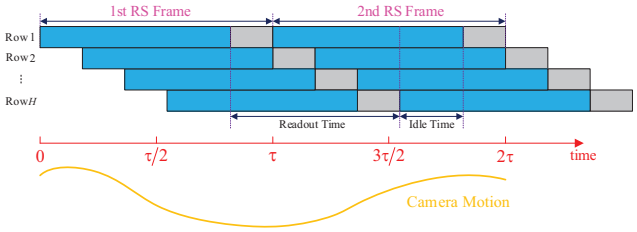


Figure 2. Illustration of the exposure mechanism of the RS camera over two consecutive frames. Assume the sensor exposure is instantaneous and each frame time is τ. The sensor is exposed and readout row by row, resulting in the scanline-varying camera poses. Our approach aims to restore the time-centered GS image that corresponds to the intermediate time τ of these two frames.

They can also be further subdivided into single-frame-based and multi-frame-based methods, respectively.

Classical model based RS correction methods. Many techniques have been implemented for RS image correction from two or more RS frames. The RS correction was posed as a temporal super-resolution problem in [4] to mitigate RS wobble in an RS video stream. [11] employed a homography mixture to achieve joint RS removal and video stabilization. [27, 10] assumed that the RS camera has either pure rotation or in-plane translational motion. An RS-aware warping [36] was proposed to rectify RS images based on a differential formulation, where linear solvers were developed to recover the relative pose of the RS camera that experienced a specific motion between two consecutive frames. They further refined both the camera motion and the depth map from dense correspondences to perform RS correction, which also determined that it relied too much on optical flow. The occlusion-aware undistortion method [33] removed the depth-dependent RS distortions from a specific setting of $\geq$ 3 RS images, assuming a piece-wise planar 3D scene. Such methods rely on computationally expensive non-linear optimizations by inputting a large number of correspondences. More recently, [37] proposed a differential RS homography model together with a minimal solver to account for the underlying scanline-varying poses of RS cameras, which can be used to perform RS-aware image stitching and rectification at one stroke. [2] explored a simple two-camera rig, mounted to have different RS directions, to undistort the RS images acquired by a smartphone.

Removing RS artifacts based on a single image frame is inherently a highly ill-posed task. To make it tractable, some external constraints about camera motion or scene structure need to be enforced. [26] attempted to exploit the presence of straight lines in urban scenes. [23] modeled the Ackermann motion based on a known vertical direction [35], while [24] leveraged a prior to the Manhattan world. [17] relaxed the Manhattan assumption and required merely the (curved) images of straight 3D lines to undistort a single RS image. [26, 17, 24] also assumed that the camera underwent the pure rotational motion. Hence, they cannot work well if these underlying assumptions on scene structures and camera motions do not hold.

Deep learning based RS correction methods. The success of deep learning in high-level vision tasks has been gradually extended to the RS geometry estimation problem (camera motion and scene structure), where a convolutional neural network (CNN) was trained to warp the RS images to their perspective GS counterparts. Rengarajan et al. [25] proposed the first CNN to correct a single RS image by assuming a simple affine motion model. Afterward, Zhuang et al. [38] extended [25] to learn the underlying scene structure and camera motion from a single RS image, followed by a post-processing step to produce a geometrically consistent GS image. Note that they often follow a general rule popular in classical methods (*e.g.*, [36, 37, 17, 24, 25, 34]) that returns the GS image under the first scanline time (0 or τ). Very recently, Liu et al. [20] used two consecutive RS images as input and designed a deep shutter unrolling network to predict the GS image corresponding to the middle time of the second frame ($\frac{3\tau}{2}$). To the best of our knowledge, our symmetric RS undistortion network is the first that is developed to learn the mapping from two input consecutive RS frames to the target GS frame corresponding to the intermediate time of these two frames (τ).

3. Approach

Our network accepts two consecutive RS images and outputs a corrected GS image that corresponds to the intermediate time of these two frames. To effectively exploit and aggregate contextual information of two consecutive RS images, our pipeline is symmetrically constructed and consists of two main parts, *i.e.*, a PWC-based undistortion flow estimator network and a time-centered GS image decoder network, as shown in Fig. 3. Henceforth, let $\boldsymbol{I}$ denote the image, $\boldsymbol{c}$ the feature representation, and $\boldsymbol{F}$ the undistortion flow. Meanwhile, in the subscript, $t \in \{1, 2\}$ indicates the t-th RS image, $t \rightarrow g$ shows the corresponding GS counterpart warped from the t-th RS content, and g corresponds to the time-centered corrected GS instance. Particularly, the superscript l represents the products of $\frac{1}{2^{l-1}}$ resolution corresponding to the l-th pyramid level. Since our network is symmetric, next we will describe only the network architecture associated with the first RS image.

We first build a weight-sharing feature pyramid for two input RS images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. At the top L-th pyramid level, we construct a cost volume by comparing features of a pixel in the first RS image with corresponding features in the second RS image. As the top level is of small spatial resolution, we can construct the cost volume using a small search range. The cost volume and features $\boldsymbol{c}_1^L$ of the first image are then fed to a CNN to estimate the undistortion flow

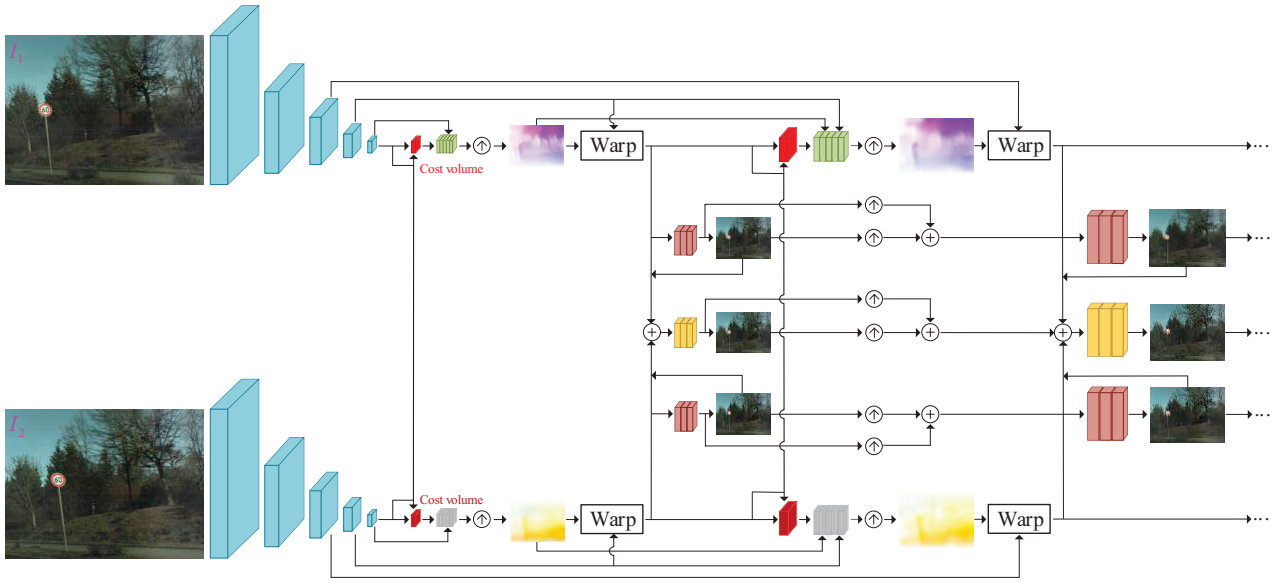


Figure 3. Overall network architecture. It mainly consists of two sub-networks: a PWC-based undistortion flow estimator and a time-centered GS image decoder. We only show the RS correction modules at the top two levels. For the rest of the pyramidal levels (excluding the first two layers), the overall RS correction modules have a similar structure as the second to the top level. Note that only the second to fifth pyramid features are warped, following a tailored correlation GS image decoder. Our network is designed symmetrically to aggregate two consecutive RS images in a coarse-to-fine manner. The symmetric convolutional layers of the same color share the same weights.

$\boldsymbol{F}_{1\to g}^{L-1}$ followed by an upsampling operation. The upsampled undistortion flow $\boldsymbol{F}_{1\to g}^{L-1}$ is then delivered to the next pyramid level. At the second to the top level, we warp features $\boldsymbol{c}_1^{L-1}$ of the first RS image to its GS counterpart $\boldsymbol{c}_{1\to g}^{L-1}$ using this upsampled undistortion flow. Meanwhile, $\boldsymbol{c}_{1\to g}^{L-1}$ is fed to a CNN to predict the forward GS image $\boldsymbol{I}_{1\to g}^{L-1}$. The second RS image is similarly processed. Further, $\boldsymbol{I}_{1\to g}^{L-1}$, $\boldsymbol{I}_{2\to g}^{L-1}$, $\boldsymbol{c}_{1\to g}^{L-1}$, and $\boldsymbol{c}_{2\to g}^{L-1}$ are concatenated into $\boldsymbol{c}_g^{L-1}$, which is then decoded by CNN layers to recover the time-centered corrected GS image $\boldsymbol{I}_g^{L-1}$. Next, we construct a context-aware cost volume using features $\boldsymbol{c}_{1\to g}^{L-1}$ and $\boldsymbol{c}_{2\to g}^{L-1}$ warped from the first and the second RS images respectively. As warping compensates for the large motion, we can still use a small search range to construct the cost volume. This cost volume, features $\boldsymbol{c}_1^{L-1}$ of the first RS image, and the upsampled undistortion flow $\boldsymbol{F}_{1\to g}^{L-1}$ are fed to a CNN to estimate a square-scale undistortion flow $\boldsymbol{F}_{1\to g}^{L-2}$ at the $(L-1)$-th level. These processes repeat until the desired level.

In the following, we introduce the key components of each module, including pyramid feature extractor, undistortion flow estimator, and time-centered GS image decoder networks. The details of the architecture are provided in the *supplementary material*.

Feature pyramid extractor. For two consecutive RS images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, we encode the L-level pyramids of feature representations to explore richer multi-scale information. Note that the bottom (0-th) level is the input images, *i.e.*, $\boldsymbol{c}_t^0 = \boldsymbol{I}_t$. We utilize a single 2D convolution with a stride of 2 to downsample the features $\boldsymbol{c}_t^{l-1}$ at the $(l-1)$-th pyramid level, and obtain the feature representation $\boldsymbol{c}_t^l$ at the l-th layer. We do not downsample the 0-th pyramid, *i.e.*, $\boldsymbol{c}_t^0$ and $\boldsymbol{c}_t^1$ have the same size. Note that the 1-th pyramid is used as a transitional layer. Specifically, three ResNet blocks [12] are performed after each downsampling operation. In our implementation, we use a 6-level pyramid, *i.e.*, $L = 5$, consisting of 5 levels of CNN features and the input RS images as the bottom level. Wherein, the number of feature channels is set to 16, 32, 64, 96, and 128, respectively.

Context-aware cost volume layer. After obtaining the upsampled undistortion flow $\boldsymbol{F}_{t\to g}^{l-1}$ at the l-th layer, we warp the feature $\boldsymbol{c}_t^{l-1}$ of the t-th RS image to its GS counterpart $\boldsymbol{c}_{t\to g}^{l-1}$ using the forward warping block [20], which can compensate for RS distortions and put the corrected GS image patches at the right scale. The warped features are the same as the pyramid features at the top level, *i.e.*, $\boldsymbol{c}_{t\to g}^L = \boldsymbol{c}_t^L$. We further construct a cost volume $\boldsymbol{cv}_{1\to 2}^{l-1}$ as the correlation [6, 32] between $\boldsymbol{c}_{1\to g}^{l-1}$ and $\boldsymbol{c}_{2\to g}^{l-1}$ by taking $\boldsymbol{c}_{2\to g}^{l-1}$ as a reference. Similarly, with $\boldsymbol{c}_{1\to g}^{l-1}$ as a reference, $\boldsymbol{cv}_{2\to 1}^{l-1}$ can also be obtained accordingly. A search range of d pixels is used to compute the cost volume at each level. Note that this is more reasonable than the ill-aligned embedding of cost volume in [20]. Building these multi-resolution matching costs can promote mutual consistency between the contextual warped feature representations $\boldsymbol{c}_{1\to g}^{l-1}$ and $\boldsymbol{c}_{2\to g}^{l-1}$, which is conducive to improving the fidelity of the subsequent corrected GS images decoded by the aggregated features.

Undistortion flow estimator. Next, we feed features $\boldsymbol{c}_t^{l-1}$ of the t-th RS image, the undistortion flow $\boldsymbol{F}_{t\to g}^{l-1}$, and the corrsponding cost volume into five DenseNet blocks [14]. The undistortion flow $\boldsymbol{F}_{t\to g}^{l-2}$ is then estimated followed by an upsampling operation at the $(l-1)$-th level. To connect the subsequent time-centered GS image decoder modules, we set the desired level to be l_0, *i.e.*, our model outputs multi-scale undistortion flows $\boldsymbol{F}_{t\to g}^{l}$, $L-1 \geq l \geq l_0-1$, which can be used to estimate the multi-scale GS images with different resolutions. Note that the number of feature channels for these five DenseNet blocks is respectively 128, 128, 96, 64, and 32 at the top pyramid level. Decreasingly, we have 64, 64, 48, 32, 16 for the $(L-1)$-th pyramid level and 32, 32, 24, 16, 8 for the $(L-2)$-th pyramid level.

Time-centered GS image decoder. Inspired by the image decoder proposed in [20], we employ three ResNet blocks [12], whose structure is consistent with that in the feature pyramid extractor, followed by a GS image prediction layer and a deconvolutional layer. The warped features $\boldsymbol{c}_{1\to g}^{l-1}$ of the first RS image, the warped features $\boldsymbol{c}_{2\to g}^{l-1}$ of the second RS image, and the concatenated features $\boldsymbol{c}_g^{l-1}$ are embedded to generate forward and backward GS images $\boldsymbol{I}_{1\to g}^{l-1}$ and $\boldsymbol{I}_{2\to g}^{l-1}$, and a target GS image $\boldsymbol{I}_g^{l-1}$ between l-th and $(l-1)$-th pyramid levels, respectively. Note that the concatenated features $\boldsymbol{c}_g^{l-1}$ fuse $\boldsymbol{c}_{t\to g}^{l-1}$, $\boldsymbol{I}_{t\to g}^{l-1}$, and features of the previous level (if it exists) in an adaptive selection manner. Our network finally outputs a half-resolution GS image and we use bilinear upsampling followed by a 2D convolution to obtain the full-resolution time-centered GS image.

Training loss. Given a pair of consecutive RS images $\{\boldsymbol{I}_t\}_1^2$, our SUNet predicts the undistortion flows $\boldsymbol{F}_{t\to g}^{l-1}$, the forward/backward GS images $\boldsymbol{I}_{t\to g}^{l-1}$, and the corrected GS image $\boldsymbol{I}_g^{l-1}$ at the l-th pyramid level ($l \geq l_0$). Let $\boldsymbol{I}_{GT}^{l-1}$ denote the corresponding ground truth (GT) GS image in the intermediate time of two consecutive RS frames. Note that the superscript $^{l-1}$ indicates $\frac{1}{2^{l-2}}$ resolution images. Our loss function can be formulated as:

$$\mathcal{L} = \lambda_r \mathcal{L}_r + \lambda_p \mathcal{L}_p + \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s. \tag{1}$$

Reconstruction loss $\mathcal{L}_r$. We model the pixel-wise reconstruction quality of the corrected GS image on multiple scales as:

$$\mathcal{L}_r = \sum_{l=l_0-1}^{L} \left\| \boldsymbol{I}_{GT}^{l-1} - \boldsymbol{I}_g^{l-1} \right\|_1, \tag{2}$$

where all images are defined in the RGB space.

Perceptual loss $\mathcal{L}_p$. To mitigate the blurry effect in the corrected GS image, similar to [20], we employ a perceptual loss $\mathcal{L}_p$ [15] to preserve details of the predictions and make estimated GS image sharper. $\mathcal{L}_p$ is defined as:

$$\mathcal{L}_p = \sum_{l=l_0-1}^{L} \left\| \phi\left(\boldsymbol{I}_{GT}^{l-1}\right) - \phi\left(\boldsymbol{I}_g^{l-1}\right) \right\|_1, \tag{3}$$

where ϕ represents the $conv3\_3$ feature extractor of the VGG19 model [30].

Consistency loss $\mathcal{L}_c$. To combine cues from $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, we enforce their respective warped feature representations $\boldsymbol{c}_{1\to g}^{l-1}$ and $\boldsymbol{c}_{2\to g}^{l-1}$ to be as close to each other as possible. This operation also facilitates subsequent alignment in the concatenated feature representation $\boldsymbol{c}_g^{l-1}$, thereby alleviating artifacts in the corrected GS image $\boldsymbol{I}_g^{l-1}$ predicted by $\boldsymbol{c}_g^{l-1}$. Specifically, we introduce the pixel-wise consistency loss $\mathcal{L}_c$ to supervise the network to align the forward and backward images $\boldsymbol{I}_{1\to g}^{l-1}$ and $\boldsymbol{I}_{2\to g}^{l-1}$ predicted by the first and the second RS images respectively across different levels, so that it can recover more details. $\mathcal{L}_c$ is defined as:

$$\mathcal{L}_c = \sum_{t=1}^{2} \sum_{l=l_0}^{L} \left\| \boldsymbol{I}_{GT}^{l-1} - \boldsymbol{I}_{t\to g}^{l-1} \right\|_1. \tag{4}$$

Smoothness loss $\mathcal{L}_s$. Finally, we add a smoothness term [21] to encourage piecewise smoothness in the estimated undistortion flows as:

$$\mathcal{L}_s = \sum_{t=1}^{2} \sum_{l=l_0}^{L} \left\| \nabla \boldsymbol{F}_{t\to g}^{l-1} \right\|_2. \tag{5}$$

Note that the context-aware cost volume and warping layers have no learnable parameters. Our network can be end-to-end trained as each of its modules is differentiable.

4. Experimental Results

Datasets. As far as we know, the two RS datasets Carla-RS and Fastec-RS published in [20] are the only RS datasets that provide ground truth GS supervisory signals and are therefore suitable for our task. The Carla-RS dataset is generated from a virtual 3D environment using the Carla simulator [7], involving general six degrees of freedom motions. The Fastec-RS dataset contains real-world RS images synthesized by a professional high-speed GS camera. Following the evaluation in [20], we also test our algorithm on these two benchmarks. Similarly, the Carla-RS dataset is divided into a training set of 210 sequences and a test set of 40 sequences, and the Fastec-RS dataset has 56 sequences for training and 20 sequences for the test. There are no overlapping scenarios between the training set and the test set. Note that the Carla-RS dataset provides the ground truth occlusion masks. Following [20], we set quantitative evaluation experiments as the Carla-RS dataset with occlusion mask (CRM), the Carla-RS dataset without occlusion mask (CR), and the Fastec-RS dataset (FR), respectively.

Implementation details. The network is implemented in PyTorch. The weights have been determined empirically as $\lambda_r = 10$, $\lambda_p = 1$, $\lambda_c = 5$, and $\lambda_s = 0.1$. We downsample the ground truth GS images to obtain the supervision signals at different levels. The desired level l_0 is set to 3 and

Table 1. Ablation study on the context-aware cost volume. Removing the cost volume (0) leads to consistently worse performances. Our network can handle large RS distortions using a small search range to compute the cost volume.

Max. Disp.	PSNR↑			SSIM↑	
	CRM	CR	FR	CR	FR
0	21.87	21.84	25.21	0.66	0.76
2	28.76	28.60	28.07	0.84	0.83
4 (Ours)	**29.28**	**29.18**	**28.34**	**0.85**	**0.84**
6	29.22	29.11	28.14	**0.85**	0.83

Table 2. Ablation study on the size of the convolving kernel at the 1-th pyramid level. Note that "0×0" means that we remove the first layer of the current feature pyramid extractor, *i.e.*, $L = 4$ and the new $\boldsymbol{c}_t^1$ is half the size of $\boldsymbol{c}_t^0$. The transitional feature representation across the 1-th pyramid level can promote the better perception of large RS distortions, thereby producing substantially better results.

Kernel Size	PSNR↑			SSIM↑	
	CRM	CR	FR	CR	FR
0×0	29.02	28.91	27.83	0.84	0.82
3×3	29.08	28.93	28.13	**0.85**	0.83
5×5	29.26	29.16	28.17	**0.85**	0.83
7×7 (Ours)	**29.28**	**29.18**	**28.34**	**0.85**	**0.84**

the top pyramid layer L is set to 5, *i.e.*, we only work with the feature representations captured by the second to fifth pyramid layers to estimate the undistortion flow and correct the RS images. We set a search range of 4 pixels to compute the cost volume at each level, *i.e.*, $d = 4$. We adopt the Adam optimizer [16] with a learning rate of 10^{-4}. The chosen batch size is 6 and the network is trained for 400 epochs. We use a uniform random crop at a horizontal resolution of 256 pixels for data augmentation. Note that we do not change the longitudinal resolution to maintain the inherent characteristics of consecutive RS images, which can help to better learn to accumulate contextual information of two consecutive RS images.

Competing methods. To evaluate the performance of the proposed approach, we compare with two representative two-frame-based RS correction methods [36, 20] that are the most relevant baselines to our approach. In addition, we also compare with the state-of-the-art single-frame-based RS correction method [38]. Since [38] does not release their code, we follow the reimplementation by [20]. Note that the Fastec-RS dataset does not provide ground truth depth and motion, so we are unable to complete the RS correction using [38]. A constant velocity model is assumed and solved to achieve RS correction in [36]. Because the deep shutter unrolling network (DSUN) [20] recover the GS image corresponding to the time of the middle scanline of the second RS frame, we thus adjust and retrain it to predict the GS image corresponding to the intermediate time of two consecutive RS frames for consistent and fair comparisons.

Evaluation metrics. Following previous works, we use PSNR and SSIM metrics to report the quantitative results of our method. The larger the PSNR/SSIM score, the higher the quality of the corrected GS image.

Table 3. Ablation study on whether to multiply the time offsets. Compared with [20], our symmetric network can implicitly learn the dependence of RS undistortion flow on scanline time.

Time Offset	PSNR↑			SSIM↑	
	CRM	CR	FR	CR	FR
Yes	28.56	28.45	28.13	0.83	0.83
No (Ours)	**29.28**	**29.18**	**28.34**	**0.85**	**0.84**

Table 4. Ablation study on the consistency loss. $\lambda_c = 0$ means no consistency loss is used. The self-supervised consistency loss is defined as measuring only the difference between forward and backward GS images. Our loss function is effective to align contextual cues, especially the Fastec-RS dataset.

Consist. Loss	PSNR↑			SSIM↑	
	CRM	CR	FR	CR	FR
$\lambda_c = 0$	29.05	28.94	27.89	0.84	0.82
Self-supervised	29.15	28.99	28.02	**0.85**	0.83
Ours	**29.28**	**29.18**	**28.34**	**0.85**	**0.84**

Ablation studies. We study the role of each proposed component in SUNet, for further insight into the design choices.

1) Context-aware cost volume. We analyze the effect of different search range in the cost volume on the performance of our approach, shown in Table 1. One can see that removing the cost volume is consistently bad for GS image restoration, which indicates that the cost volume can effectively promote the perception and alignment of the warped contexts of two consecutive RS images. Thus the contextual cost volume can be regarded as a core unit of our network to deal with the non-local operations of the RS correction. Furthermore, a 4-pixel search range is enough to undistort the RS images successfully, which is due to the fact that a 4-pixel search range is capable of handling up to 100-pixel displacements at the input resolution. Specifically, a larger range leads to similar performance, so a larger search range may bring better benefits when the image resolution increases or the RS distortion is severe.

2) Kernel size. Although we do not directly utilize the feature representation of the first pyramid level, we investigate its influence as a transitional layer on the final performance, shown in Table 2. First, we directly remove the first layer of the current feature pyramid extractor, *i.e.*, a 5-level pyramid is used. Then we test different convolution kernel sizes at the first pyramid level. All these settings have similar performance in terms of PSNR and SSIM, but we empirically observe that a 7×7 kernel size has relatively better correction capabilities on the foreground, resulting in better visual quality. One possible explanation is that the foreground objects exist more serious RS distortion, and a larger convolution kernel can increase the receptive field, which therefore contributes to recovering more geometrically consistent GS images.

3) Time offset. The undistortion flow depends not only on the camera motion and the 3D scene geometry but also

(a) Original RS image 2 (b) Ground truth GS image (c) Zhuang *et al.* [36] (d) Liu *et al.* [20] (e) Ours

Figure 4. Qualitative results against baseline methods on the Fastec-RS dataset. Even rows: absolute difference between the corresponding image and the ground truth GS image. (c-e) GS images predicted by Zhuang *et al.* [36], Liu *et al.* [20], and our approach, respectively.

on the scanline time of a particular pixel. Thus, [20] explicitly multiplies the velocity field with the time offset to model the adapted undistortion flow. Similarly, we also explicitly multiply our upsampled undistortion flows by the corresponding normalized time offsets between the exposure time of the captured pixel and the intermediate time of the two frames. Interestingly, the results in Table 3 demonstrate that our SUNet can directly return the suitable RS-aware undistortion flows to transform the learned feature representations to their corresponding GS counterparts. We reckon this is because explicit modeling of time offset at multi-resolution may confuse the network, while our PWC-based symmetric architecture can successfully learn the inherent scanline-dependent characteristics of the undistortion flow. See *supplementary materials* for more analyses.

4) Loss function. We also perform two experiments to prove the effectiveness of the proposed consistency loss $\mathcal{L}_c$. One is to remove $\mathcal{L}_c$ from the loss function $\mathcal{L}$ in Eq. 1, *i.e.*, $\lambda_c = 0$. The other is to change the consistency metric in Eq. 4 to the direct difference between forward/backward GS images $\boldsymbol{I}_{1\to g}^{l-1}$ and $\boldsymbol{I}_{2\to g}^{l-1}$ without introducing the ground truth GS images $\boldsymbol{I}_{GT}^{l-1}$, where λ_c is still fixed at 5, which constitutes a self-supervised metric. As can be seen from Table 4, removing the consistency loss consistently weakens the performance of GS image restoration, and the self-supervised metric is not effective enough, especially in the Fastec-RS dataset. In addition, they will cause visible seams in the foreground, such as the signboard in Fig. 1. We also find that they all degrade the estimates of both $\boldsymbol{I}_{1\to g}^{l-1}$ and $\boldsymbol{I}_{2\to g}^{l-1}$ to black, which sacrifices the interpretability of the network in the geometric sense. One possible reason is that the occlusion between $\boldsymbol{I}_{1\to g}^{l-1}$ and $\boldsymbol{I}_{2\to g}^{l-1}$ is more serious than that between them and $\boldsymbol{I}_{GT}^{l-1}$, which degrades the predictions of both $\boldsymbol{I}_{1\to g}^{l-1}$ and $\boldsymbol{I}_{2\to g}^{l-1}$ and thus contributes little to the alignment of the contextual cues. Overall, it proves our loss function in Eq. 1 can fully supervise the RS correction.

Table 5. Quantitative comparisons of the performance between our approach and the state-of-the-art baseline methods. Note that we cannot use [38] to benchmark the Fastec-RS dataset due to its lack of training ground truth. Our approach again performs the best.

Methods	PSNR↑ (dB)			SSIM↑	
	CRM	CR	FR	CR	FR
Single-frame [38]	18.70	18.47	-	0.58	-
Model-based [36]	25.93	22.88	21.44	0.77	0.71
DSUN [20]	26.90	26.46	26.52	0.81	0.79
SUNet (Ours)	**29.28**	**29.18**	**28.34**	**0.85**	**0.84**

Comparison with baseline methods. We report the quantitative and qualitative comparisons against the baseline methods in Table 5 and Fig. 4, respectively. See our *supplementary materials* for more results. The experimental results demonstrate that our approach outperforms the three state-of-the-art methods by a significant margin. Note that the deep shutter unrolling network (DSUN) [20] is most relevant to our approach. However, DSUN is difficult to restore the GS image details that have not been seen by the second RS image, which also validates our preceding analysis in Section 1. The visually unpleasing areas, *i.e.*, the lower parts of the corrected GS image, are mainly concentrated in the scanlines with a large time offset from the intermediate time of two input frames. On the contrary, our approach can overcome these obstacles quite well, as shown in Fig. 4. Also, the forward and backward GS images rectified from the first frame and the second frame respectively are depicted in Fig. 1(e)&(f), which indicates the importance of contextual cues. More analyses are shown in the *supplementary materials*.

Furthermore, [36] is a classical model based RS correction method with two consecutive frames by assuming a constant velocity motion model. However, its performance in poorly textured regions is not satisfactory. [38] uses a single RS image as input to achieve RS correction, but it shows limited generalization performance on the Carla-RS dataset. A possible reason is that the estimation of cam-

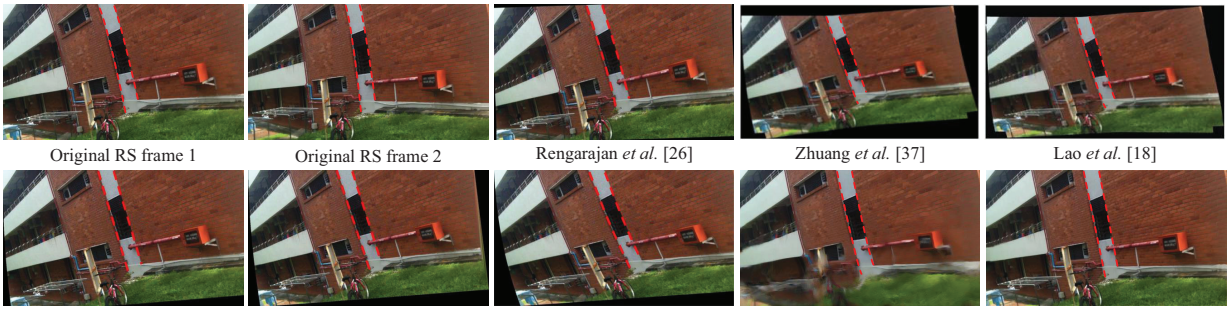


Figure 5. Qualitative comparisons against relative methods using data provided by [36]. Our pipeline estimates a plausible GS image and complements the occluded regions based on the learned image prior.

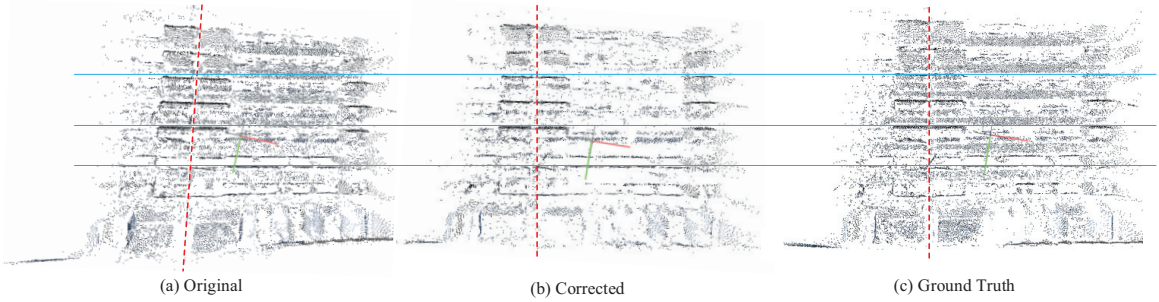


Figure 6. SfM results by Colmap [28] for a building. (a) Reconstructed 3D model with original RS images. (b) Reconstructed 3D model with corrected GS images. (c) Reconstructed 3D model with ground truth GS images. It demonstrates that our pipeline removes the undesired RS distortion and generates a more accurate 3D model as the ground truth 3D model.

era motion and scene depth is degraded due to unseen scene contents in the test data. Also, the specific RS camera model assumptions in post-processing may have poor adaptability to this dataset. In contrast, our data-driven approach solves a general two-frame-based RS correction problem without relying on specific RS camera model assumptions, thereby achieving better performances. It is also worth mentioning that, benefit from the powerful expressive power of CNNs, our approach can effectively fill the occluded regions, which is unable to be reconstructed by [36] and [38]. An intuitive analysis is that on the Carla-RS dataset, the difference of PSNR in [36] with or without the mask is 3.05 dB, while ours is only 0.1 dB.

Inference time. Our method can correct RS images with a resolution of 640×480 pixels using an average of 0.21 seconds on an NVIDIA GeForce RTX 2080Ti GPU, which is faster than the average 0.34 seconds of DSUN [20]. Note that the classical two-frame-based RS correction method [36] takes several minutes on an Intel Core i7-7700K CPU.

Generalization performance and 3D reconstruction. We use the real data provided by [36] to carry out the generalization experiment, and make a qualitative comparison with several relevant RS correction methods. The results are summarized in Fig. 5. We can see that our approach owns good generalization ability and recovers a visually compelling GS image. Moreover, we run an SfM pipeline (*i.e.*, Colmap [28]) to process the original RS image sequences, the corrected GS image sequences, and the ground truth GS image sequences, respectively. Fig. 6 demonstrates that our approach can correct the geometric inaccuracies and generate a more accurate 3D structure that complies with the underlying ground truth 3D scene geometry. Note that one can see the obvious RS distortions (*e.g.*, skew and shrink) in the 3D model obtained by the original RS images.

5. Conclusion

In this paper, we have proposed an end-to-end symmetric undistortion network for generic RS correction. Given two consecutive RS images, it can effectively estimate the GS image corresponding to the intermediate time of these two frames. Our context-aware undistortion flow estimator and the symmetric consistency enforcement can efficiently reduce the misalignment between the contexts warped from two consecutive RS images, thus achieving state-of-the-art RS correction performances. Currently, we concern ourselves with the RS correction problem corresponding to a particular time. In the future, we will explore more challenging tasks, *e.g.*, a typical exposure time manipulated by the user to complete the corresponding RS correction.

Acknowledgments

This research was supported in part by National Key Research and Development Program of China (2018AAA0102803) and National Natural Science Foundation of China (61871325, 61901387). We would like to thank the anonymous reviewers for their useful feedback.

References

[1] Omar Ait-Aider, Nicolas Andreff, Jean Marc Lavest, and Philippe Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *Proceedings of the European Conference on Computer Vision*, pages 56–68. Springer, 2006. 1

[2] Cenek Albl, Zuzana Kukelova, Viktor Larsson, Michal Polic, Tomas Pajdla, and Konrad Schindler. From two rolling shutters to one global shutter. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2505–2513. IEEE, 2020. 3

[3] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. R6p-rolling shutter absolute camera pose. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2292–2300. IEEE, 2015. 1

[4] Simon Baker, Eric Bennett, Sing Bing Kang, and Richard Szeliski. Removing rolling shutter wobble. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2392–2399. IEEE, 2010. 3

[5] Yuchao Dai, Hongdong Li, and Laurent Kneip. Rolling shutter camera relative pose: generalized epipolar geometry. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4132–4140. IEEE, 2016. 1

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: learning optical flow with convolutional networks. In *Proceedings of the International Conference on Computer Vision*, pages 2758–2766. IEEE, 2015. 4

[7] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: an open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 5

[8] Bin Fan and Yuchao Dai. Inverting a rolling shutter camera: bring rolling shutter images to high framerate global shutter video. In *Proceedings of the International Conference on Computer Vision*. IEEE, 2021. 1

[9] Bin Fan, Ke Wang, Yuchao Dai, and Mingyi He. Rs-dpsnet: deep plane sweep network for rolling shutter stereo images. *IEEE Signal Processing Letters*, 28:1550–1554, 2021. doi: 10.1109/LSP.2021.3099350. 1

[10] Per-Erik Forssén and Erik Ringaby. Rectifying rolling shutter video from hand-held devices. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 507–514. IEEE, 2010. 3

[11] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *Proceedings of the International Conference on Computational Photography*, pages 1–8. IEEE, 2012. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016. 4, 5

[13] Johan Hedborg, Per-Erik Forssén, Michael Felsberg, and Erik Ringaby. Rolling shutter bundle adjustment. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1434–1441. IEEE, 2012. 1

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4700–4708. IEEE, 2017. 5

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, pages 694–711. Springer, 2016. 5

[16] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. 6

[17] Yizhen Lao and Omar Ait-Aider. A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4795–4803. IEEE, 2018. 1, 3

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

[19] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144. IEEE, 2017. 1

[20] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 5941–5949. IEEE, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[21] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the International Conference on Computer Vision*, pages 4463–4471. IEEE, 2017. 5

[22] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4040–4048. IEEE, 2016. 1

[23] Pulak Purkait and Christopher Zach. Minimal solvers for monocular rolling shutter compensation under ackermann motion. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 903–911. IEEE, 2018. 1, 3

[24] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in Manhattan world. In *Proceedings of the International Conference on Computer Vision*, pages 882–890. IEEE, 2017. 1, 3

[25] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: cnn to correct motion distortions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2291–2299. IEEE, 2017. 1, 3

[26] Vijay Rengarajan, Ambasamudram N Rajagopalan, and Rangarajan Aravind. From bows to arrows: rolling shutter rectification of urban scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2773–2781. IEEE, 2016. 1, 3

[27] Erik Ringaby and Per-Erik Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision*, 96(3):335–352, 2012. 3

[28] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4104–4113. IEEE, 2016. 8

[29] David Schubert, Nikolaus Demmel, Lukas von Stumberg, Vladyslav Usenko, and Daniel Cremers. Rolling-shutter modelling for direct visual-inertial odometry. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 2462–2469. IEEE, 2019. 1

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015. 5

[31] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1279–1288. IEEE, 2017. 1

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 8934–8943. IEEE, 2018. 1, 2, 4

[33] Subeesh Vasu, Mahesh MR Mohan, and AN Rajagopalan. Occlusion-aware rolling shutter rectification of 3d scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 636–645. IEEE, 2018. 3

[34] Ke Wang, Bin Fan, and Yuchao Dai. Relative pose estimation for stereo rolling shutter cameras. In *Proceedings of the International Conference on Image Processing*, pages 463–467. IEEE, 2020. 3

[35] Chunhui Zhao, Bin Fan, Jinwen Hu, Quan Pan, and Zhao Xu. Homography-based camera pose estimation with known gravity direction for uav navigation. *Science China Information Sciences*, 64(1):1–13, 2021. 3

[36] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *Proceedings of the International Conference on Computer Vision*, pages 948–956. IEEE, 2017. 1, 3, 6, 7, 8

[37] Bingbing Zhuang and Quoc-Huy Tran. Image stitching and rectification for hand-held cameras. In *Proceedings of the European Conference on Computer Vision*, pages 243–260. Springer, 2020. 1, 3

[38] Bingbing Zhuang, Quoc-Huy Tran, Pan Ji, Loong-Fah Cheong, and Manmohan Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4551–4560. IEEE, 2019. 1, 3, 6, 7, 8