Inverting a Rolling Shutter Camera: Bring Rolling Shutter Images to High Framerate Global Shutter Video

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

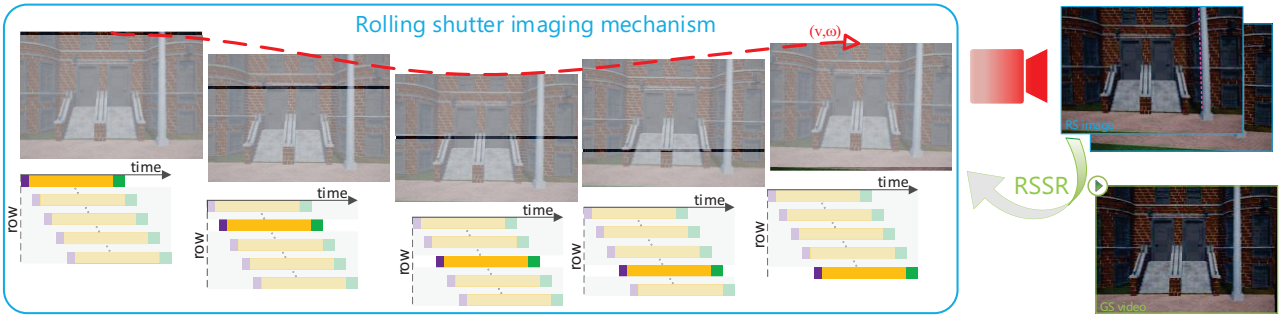

Figure 1: The RS image is generated by continuously synthesizing the GS image row by row, while our rolling shutter temporal super-resolution (RSSR) pipeline reverses this process, *i.e.*, extracting the latent GS image sequence from two consecutive RS images. *Please see the arXiv version to be able to view the GS video image (bottom right) as a video.*

Abstract

Rolling shutter (RS) images can be viewed as the result of the row-wise combination of global shutter (GS) images captured by a virtual moving GS camera over the period of camera readout time. The RS effect brings tremendous difficulties for the downstream applications. In this paper, we propose to invert the above RS imaging mechanism, i.e.*, recovering a high framerate GS video from consecutive RS images to achieve RS temporal super-resolution (RSSR). This extremely challenging problem,* e.g.*, recovering 1440 GS images from two 720-height RS images, is far from being solved end-to-end. To address this challenge, we exploit the geometric constraint in the RS camera model, thus achieving geometry-aware inversion. Specifically, we make three contributions in resolving the above difficulties: (i) formulating the bidirectional RS undistortion flows under the constant velocity motion model, (ii) building the connection between the RS undistortion flow and optical flow via a scaling operation, and (iii) developing a mutual conversion scheme between varying RS undistortion flows that correspond to different scanlines. Building upon these formulations, we propose the first RS temporal super-resolution network in a cascaded structure to extract high framerate global shutter video. Our method explores the underlying spatio-temporal geometric relationships within a deep learning framework, where no extra supervision besides the middle-scanline ground truth GS image is needed. Essentially, our method can be very efficient for explicit propagation to generate GS images under any scanline. Experimental results on both synthetic and real data show that our method can produce high-quality GS image sequences with rich details, outperforming state-of-the-art methods.*

*Corresponding author

1. Introduction

Many consumer cameras such as webcams or mobile phones are generally built upon CMOS sensors due to their low cost and simplicity in manufacturing, which commonly adopt the rolling shutter (RS) mechanism. Unlike its global shutter (GS) counterpart, an RS camera generates an image row-by-row sequentially, which gives rise to the so-called RS effect (*e.g.*, stretch, wobble) in images and videos captured by a moving RS camera. The RS effect is increasingly becoming a nuisance in photography. Simply ignoring the RS effect in computer vision applications leads to performance degradation or even failure [2, 6, 19]. However, the RS image combines information about both the scene geometry and the concealed motion [43], according to which the RS effect can be removed by single frame [18,28,29,43] or multiple frames [20, 30, 36, 41, 42]. As single-view RS

correction is inherently a highly ill-posed problem, following [20, 41, 42], using at least two consecutive frames can make it tractable. Intuitively, observing a pair of consecutive RS images, humans seem to be able to infer a plausible explanation for the underlying geometry (*i.e.*, camera motion and scene structure).

In this paper, we propose to invert the rolling shutter imaging mechanism, *i.e.*, recovering the high framerate global shutter video from consecutive rolling shutter images, as illustrated in Fig. 1. Essentially, this inversion process mimics the above human ability. This paper aims at recovering a temporal sequence of latent GS image frames from two consecutive RS images, *i.e.*, performing RS temporal super-resolution (RSSR). This task involves solving the underlying RS geometry, which is arduous for existing methods (*e.g.*, [41]) to achieve robust and accurate estimation due to subtle intra-frame motions, requiring non-trivial camera calibration and iterative optimizations.

To some extent, RSSR relates to two-view RS correction [20, 41, 42] and GS video interpolation [4, 5, 15, 24]. However, our task contains additional complexity, because beyond eliminating the geometric RS distortion, we also want to generate a set of high framerate GS images in chronological order. This is particularly challenging as we have to ensure the temporal smoothness of the output sequence. Unfortunately, the artifacts appear in the pure geometric methods [41, 42] due to inaccurate RS geometric estimation. Only one reliable GS image can be recovered by [20]. See supplementary materials for more analyses. Also, different from the slight and controllable pixel displacement in the GS video interpolation task, which is located inside its optical flow, the pixel displacement when correcting the RS image may exceed its local neighborhood defined by its optical flow, depending on the type of motion, the 3D structure and the scanline time. Specifically, we identify (for the first time) that the RS-aware pixel displacement can be obtained by scaling the corresponding optical flow vector under the constant velocity motion model, but the size and sign of the scaling factor are determined by the intrinsic RS geometry.

To address the above challenges, we formulate the bidirectional RS undistortion flows to characterize the pixel-wise RS-aware pixel displacement, and further advance a calculation method for the mutual conversion between varying RS undistortion flows corresponding to different scanlines. In particular, we prove that the scaling factor is in the interval of $(-1, 1)$ when correcting an RS image to its middle-scanline GS image. As a result of utilizing these parameterizations, we propose a data-driven solution for RSSR with good interpretability, which intrinsically encapsulates the complete underlying RS geometry that more sophisticated methods (*e.g.*, [41, 42]) struggle to learn away.

The proposed geometry-aware RSSR pipeline employs a cascaded architecture to extract a latent high framerate GS video sequence from two consecutive RS images. Firstly, we estimate the bidirectional optical flows by using the classic PWC-Net [34]. Secondly, we design a UNet-like network to learn the scaling factor of each pixel (*i.e.*, middle-scanline correlation map in Sec. 4) such that the middle-scanline RS undistortion flows can be inferred. At the same time, RS undistortion flows for any scanline can be associated and propagated explicitly. Finally, the softmax splatting [24] is used to produce the high framerate GS video frames at arbitrary scanlines. Our RSSR network can be trained end-to-end and only the middle-scanline GS images are needed for supervision. Since none of the learned network parameters are time-dependent, it can synthesize as many GS frames as needed. Extensive experimental results demonstrate that our approach is superior to the state-of-the-art methods in removing the RS artifacts, and it can generate a coherent video sequence as well.

Our main contributions are summarized as follows:

- We identify and establish a detailed proof of the scanline-dependent nature of the bidirectional RS undistortion flows, which is essential for understanding the intrinsic geometrical properties of RS correction problem [20, 41–43].
- From the theoretical perspective, we provide a sound motivation for our first learning-based RSSR solution for latent GS video sequence extraction from two consecutive RS images, which brings RS images alive.
- Our approach not only outperforms the state-of-the-art methods in both RS effect removal and inference efficiency, but also can produce a smooth and continuous video sequence far beyond the reach of [20].

2. Related Work

Geometric model based RS correction. Over the last decade, several works have revisited the RS geometric model to remove the RS effect [3, 9, 11, 14, 18, 19, 27, 29, 30, 32, 37]. Grundmann *et al.* [12] employed a homography mixture to achieve joint RS removal and video stabilization. The occlusion-aware undistortion method [36] removed the depth-dependent RS distortions from a specific setting of $\geq$ 3 RS images, assuming a piece-wise planar 3D scene. Zhuang *et al.* [41] proposed estimating the full camera motion by a differential formulation to remove RS distortions in two consecutive RS images. Such a model has achieved considerable success, but the further improvement has appeared challenging, due to the difficulties of making it robust and efficient to various situations (*e.g.*, relying too much on the initial optical flow estimation [42]). Subsequently, Zhuang and Tran [42] presented a differential RS homography to model the underlying scanline-varying RS camera poses, which can be used to perform RS-aware image stitching and correction. Albl *et al.* [1] explored a sim-

ple two-camera rig, mounted to have different shutter directions, to undistort the RS images acquired by a smartphone.

Deep learning based RS correction. Recently, convolutional neural networks (CNN) have been used to achieve more flexible and efficient RS correction. Rengarajan *et al*. [28] proposed the first CNN to correct a single RS image by assuming a simple affine model. Zhuang *et al*. [43] extended [28] to learn the RS geometry from a single frame through two independent networks, yielding geometrically more correct results. Zhong *et al*. [40] jointly handled RS correction and deblurring. Liu *et al*. [20] and Fan *et al*. [8] used two consecutive RS images as input and designed specialized CNNs to predict one GS image. To the best of our knowledge, our RSSR model is the first that is developed to learn the mapping from two consecutive RS frames to high framerate GS video frames corresponding to any scanline.

3. Differential Forward RS Geometry

GS-aware forward warping. Assuming that the GS camera experiences constant linear velocity $\mathbf{v} = [v_1, v_2, v_3]^{\mathrm{T}}$ and angular velocity $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^{\mathrm{T}}$ over two consecutive frames 1 and 2, a 3D point $\mathbf{X}$ with depth Z is observed by the camera to move with the 3D velocity as $-\mathbf{v} - \boldsymbol{\omega} \times \mathbf{X}$ [22]. Projecting this 3D velocity into the 2D image plane yields the image motion field $\mathbf{f}$, which is usually approximated by the optical flow vector $(\mathbf{f}_u, \mathbf{f}_v)^{\mathrm{T}}$ under the brightness constancy assumption, at pixel $\mathbf{x} = (x, y)$ as [22]:

$$\mathbf{f} = \frac{\mathbf{A}\mathbf{v}}{Z} + \mathbf{B}\boldsymbol{\omega} \triangleq \pi(\mathbf{v}, \boldsymbol{\omega}, \mathbf{x}, Z, f), \tag{1}$$

where

$$\mathbf{A} = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{xy}{f} & -\left(f + \frac{x^2}{f}\right) & y \\ \left(f + \frac{y^2}{f}\right) & -\frac{xy}{f} & -x \end{bmatrix}. \tag{2}$$

Here, (x, y) is the normalized image coordinate and f denotes the focal length.

RS forward motion parameterization. Since all RS scanlines are successively and instantaneously exposed one by one at different times, each scanline possesses a different optical center. To account for the scanline-varying camera poses, we make full use of the small motion hypothesis. Given the inter-frame camera velocities $(\mathbf{v}, \boldsymbol{\omega})$ between the two first scanlines of two consecutive RS images, the subtle intra-frame relative motion could be obtained by interpolation [26, 32, 41, 42]. Specifically, [41] derived a linear interpolation scheme under the constant velocity motion. Formally, the camera position and rotation $(\mathbf{p}_1^{s_1}, \mathbf{r}_1^{s_1})$ (resp. $(\mathbf{p}_2^{s_2}, \mathbf{r}_2^{s_2})$) of s_1-th (resp. s_2-th) scanline in frame 1 (resp. 2) *w.r.t.* the first scanline of frame 1 can be expressed as:

$$\begin{aligned} \mathbf{p}_1^{s_1} &= \lambda_1^{s_1}\mathbf{v}, \quad \mathbf{r}_1^{s_1} = \lambda_1^{s_1}\boldsymbol{\omega}, \\ \mathbf{p}_2^{s_2} &= \lambda_2^{s_2}\mathbf{v}, \quad \mathbf{r}_2^{s_2} = \lambda_2^{s_2}\boldsymbol{\omega}, \end{aligned} \tag{3}$$

where

$$\lambda_1^{s_1} = \tfrac{\gamma s_1}{h}, \quad \lambda_2^{s_2} = 1 + \tfrac{\gamma s_2}{h}. \tag{4}$$

Here, $\gamma = h\tau_r/\tau$ is the readout time ratio [41], *i.e.*, the ratio between the total readout time and the total time τ in an RS frame, which can be calibrated by [23, 25]. τ_r is the readout time of a single scanline and h is the total number of scanlines in an image. Consequently, the relative motion between s_1-th and s_2-th scanlines satisfies:

$$\begin{aligned} \mathbf{v}_{s_1 s_2} &= \mathbf{p}_2^{s_2} - \mathbf{p}_1^{s_1} = (\lambda_2^{s_2} - \lambda_1^{s_1})\,\mathbf{v}, \\ \boldsymbol{\omega}_{s_1 s_2} &= \mathbf{r}_2^{s_2} - \mathbf{r}_1^{s_1} = (\lambda_2^{s_2} - \lambda_1^{s_1})\,\boldsymbol{\omega}. \end{aligned} \tag{5}$$

Further details may be found in [41].

RS-aware forward warping. As $\mathbf{f}_v = s_2 - s_1$, plugging Eqs. (4) and (5) into Eq. (1), we can relate the forward optical flow of pixel $\mathbf{x}$ in frame 1 in terms of the RS camera velocity and the 3D point as:

$$\begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_v \end{bmatrix} = \alpha \begin{bmatrix} \pi_u(\mathbf{v}, \boldsymbol{\omega}, \mathbf{x}, Z, f) \\ \pi_v(\mathbf{v}, \boldsymbol{\omega}, \mathbf{x}, Z, f) \end{bmatrix}, \tag{6}$$

where

$$\alpha = 1 + \frac{\gamma \mathbf{f}_v}{h} \tag{7}$$

represents the RS-aware forward interpolation factor under the constant velocity motion model, which depends on the scanline involved in the optical flow. $\pi_u(\cdot)$ and $\pi_v(\cdot)$ denotes the first and second entries of $\pi(\cdot)$, respectively. Hence, the geometric inaccuracies induced by the RS effect could be compensated by simply scaling the optical flow vector of each pixel $\mathbf{x}$.

4. RS Undistortion Flow vs Optical Flow

RS imaging mechanism. As shown in Fig. 1, RS camera exposes each scanline in sequence, which results in a different local frame for each scanline. The RS image $\mathbf{I}_r$ can therefore be regarded as the result of a continuous row-wise combination of multiple virtual GS images over the period of camera readout time, *i.e.*, formulating the RS imaging model as:

$$\lfloor \mathbf{I}_r(\mathbf{x}) \rfloor_s = \lfloor \mathbf{I}_g^s(\mathbf{x}) \rfloor_s, \quad 0 \le s \le h-1, \tag{8}$$

where $\mathbf{I}_g^s$ is the virtual GS image captured at time $s\tau_r$. $\lfloor \cdot \rfloor_s$ indicates extracting the s-th scanline. On the contrary, the RSSR aims to reverse the above RS imaging formulation, *i.e.*, estimating the displacement vectors $\mathbf{u}_{r\to s}$ of pixel $\mathbf{x}$ from the RS image to the virtual GS image at s-th scanline such that

$$\mathbf{I}_r(\mathbf{x}) = \mathbf{I}_g^s(\mathbf{x} + \mathbf{u}_{r\to s}), \quad 0 \le s \le h-1. \tag{9}$$

Specifically, it is to estimate the dense *RS undistortion flow* $\mathbf{U}_{r\to s}$ that corresponds to scanline s by stacking $\mathbf{u}_{r\to s}$ for all pixels in matrix form. Note that similar to [20], $\mathbf{U}_{r\to s}$ is a forward warping operation, *e.g.*, softmax splatting [24].

Bidirectional RS undistortion flows. To deliver each RS pixel $\mathbf{x}$ on κ-th scanline in frame 1 to its GS canvas defined by the pose corresponding to s-th scanline of frame 1, the RS-aware forward warping displacement vector of pixel $\mathbf{x}$ can be formulated as:

$$\begin{bmatrix} \mathbf{u}_u \\ \mathbf{u}_v \end{bmatrix} = \beta \begin{bmatrix} \pi_u(\mathbf{v}, \boldsymbol{\omega}, \mathbf{x}, Z, f) \\ \pi_v(\mathbf{v}, \boldsymbol{\omega}, \mathbf{x}, Z, f) \end{bmatrix}, \tag{10}$$

where

$$\beta = \frac{\gamma(s-\kappa)}{h} \tag{11}$$

represents the RS-aware forward undistortion factor, which depends on the scanline offset between the target scanline s and the current scanline κ. Note that $0 \le s, \kappa \le h-1$ and the pixels in s-th scanline of the RS image remain unchanged. Stacking the forward warping displacement vectors of all pixels in frame 1, we can obtain the pixel-wise forward RS undistortion flow $\mathbf{U}_{1\to s}$ of frame 1, which could be used to restore the GS image corresponding to any scanline $s \in [0, h-1]$ of frame 1.

Unfortunately, Eqs. (6) and (10) describe merely the forward optical flow (*i.e.*, from frame 1 to frame 2) and the forward RS undistortion flow (*i.e.*, from frame 1 to scanline s), respectively. We cannot remove the geometric RS distortion in frame 2 based on two input consecutive RS images alone. Therefore, we further propose a **RS-aware backward warping model**[1] accounting for frame 2, which solely needs to *negative for the readout time ratio* γ. Accordingly, Z refers to the depth of each pixel in frame 2 and $(\mathbf{v}, \boldsymbol{\omega})$ need to be reversed. As a consequence, Eq. (6) indicates the backward optical flow from frame 2 to frame 1 and Eq. (10) models the backward RS undistortion flow $\mathbf{U}_{2\to s}$ in frame 2 (*i.e.*, from frame 2 to scanline s). Similarly, the GS image corresponding to any scanline $s \in [0, h-1]$ of frame 2 can be recovered.

Heretofore, we have obtained bidirectional RS undistortion flows $\mathbf{U}_{1\to s}$ and $\mathbf{U}_{2\to s}$, which can warp two consecutive RS images to GS images corresponding to their respective scanlines $s \in [0, h-1]$, *i.e.*, activating the rolling shutter temporal super-resolution.

Connection between RS undistortion flow and optical flow. Note that the optical flow in Eq. (6) exhibits the pixel displacement over two consecutive RS frames, while the RS undistortion flow in Eq. (10) models the pixel displacement between the RS frame 1 (or frame 2) and the GS frame at scanline s. Without loss of generality, taking the forward warping as an example (note that the backward warping is similar, except that γ is negative). First, eliminating $\mathbf{f}_v$ on the right hand side of Eq. (6), the optical flow of pixel $\mathbf{x}$ can be rewritten as:

$$\begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_v \end{bmatrix} = \frac{h}{h - \gamma\pi_v} \begin{bmatrix} \pi_u \\ \pi_v \end{bmatrix}, \tag{12}$$

where π_u and π_v are abbreviations, determined by camera parameters, camera motions and 3D depths. Then, through Eqs. (10) and (12), we establish the connection between the forward/backward RS undistortion flow $(\mathbf{u}_u, \mathbf{u}_v)^{\mathrm{T}}$ and the forward/backward optical flow $(\mathbf{f}_u, \mathbf{f}_v)^{\mathrm{T}}$ at pixel $\mathbf{x}$ as:

$$\begin{bmatrix} \mathbf{u}_u \\ \mathbf{u}_v \end{bmatrix} = c \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_v \end{bmatrix}, \tag{13}$$

where

$$c = \frac{\gamma(s-\kappa)(h-\gamma\pi_v)}{h^2} \tag{14}$$

represents the forward/backward correlation factor between bidirectional RS undistortion flow and bidirectional optical flow, which are distinguished by the sign of γ. Stacking c for all pixels yields the forward and backward correlation maps $\mathbf{C}_{1\to s}$ and $\mathbf{C}_{2\to s}$, respectively.

Consequently, after obtaining the bidirectional optical flows between two consecutive RS frames (*e.g.* via PWC-Net [34] or RAFT [35]), we can scale them to obtain the

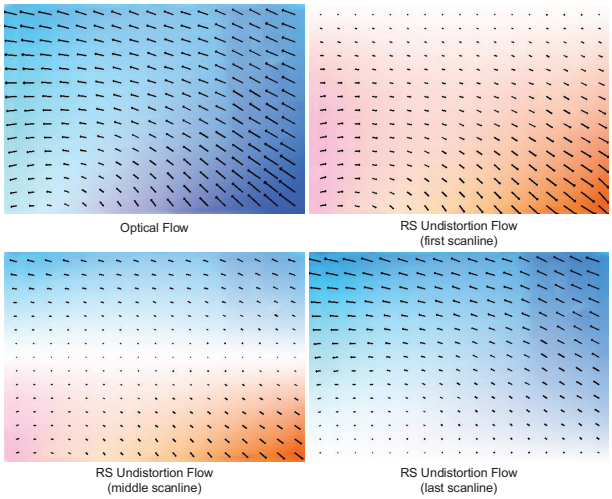

Figure 2: RS undistortion flow versus optical flow. Here, forward flows are visualized according to [33]. Compared to the isotropically smooth optical flow map, the RS undistortion flow map exhibits a more significant scanline dependence. On the one hand, the RS undistortion flows near the target scanline appear as lighter colors (*i.e.*, smaller warping displacement values). On the other hand, the RS undistortion flows corresponding to pixels smaller than and larger than the target scanline show different colors (*i.e.*, different warping displacement directions).

[1]The derivation is available in the supplementary material.

bidirectional RS undistortion flows to further recover the latent GS images corresponding to specific scanline $s \in [0, h-1]$. Note that the size and sign of the scaling factor (*i.e.*, correlation factor c) rely on the underlying RS geometry. As illustrated in Fig. 2, we can intuitively observe that, compared to the isotropically smooth optical flow map, the RS undistortion flow is more typically scanline-dependent. The closer the pixel to scanline s (assuming to correction to s-th scanline, *e.g.*, the first scanline), the smaller the value of the RS undistortion flow is generally. Moreover, we point out that Eq. (13) is essentially the core of RS correction in [41], but it requires sophisticated processing (such as RANSAC [10], non-convex optimization, etc.) to estimate dense depth as well as accurate camera motion based on the optical flow map. Nevertheless, potential gross errors in optical flow estimations could lead to severe artifacts [42].

Interconversion between varying RS undistortion flows. Eq. (10) defines the RS undistortion flow that warps the RS frame to its GS counterpart corresponding to scanline s. Assuming that two GS images corresponding to s_1-th scanline and s_2-th scanline are to be restored, it can be seen from Eqs. (13) and (14) that the difference in the warping displacement vector of the same pixel $\mathbf{x}$ is only $(s_1 - \kappa)$ and $(s_2 - \kappa)$. Note that $\mathbf{x}$ lies in scanline κ. Therefore, we can simply extend the RS undistortion flow of pixel $\mathbf{x}$ facing scanline s_1 to that for scanline s_2 by

$$\begin{bmatrix} \mathbf{u}_u^{s_2} \\ \mathbf{u}_v^{s_2} \end{bmatrix} = \frac{s_2 - \kappa}{s_1 - \kappa} \begin{bmatrix} \mathbf{u}_u^{s_1} \\ \mathbf{u}_v^{s_1} \end{bmatrix}. \tag{15}$$

In other words, the resulting RS undistortion flows corresponding to different scanlines can be converted to each other, which is also the key to achieve our RSSR solution. An example is shown in Fig. 2. One can find that the difference between varying RS undistortion flows is not only reflected in the distribution of the warping displacement size, but also the warping displacement direction is closely related to the scanline, as elucidated in Proposition 1.

Proposition 1 *Given bidirectional optical flows $\mathbf{F}_{1\to 2}$ and $\mathbf{F}_{2\to 1}$, to recover the GS image corresponding to the camera pose of the middle scanline, the forward correlation map $\mathbf{C}_{1\to m}$ (resp. backward correlation map $\mathbf{C}_{2\to m}$) for the forward RS undistortion flow $\mathbf{U}_{1\to m}$ (resp. backward RS undistortion flow $\mathbf{U}_{2\to m}$) in Eq. (13) satisfies:*

$$\lfloor \mathbf{C}_{1\to m} \rfloor_{i,j} \in \begin{cases} (0,1) & \text{if } i < \frac{h}{2} \\ 0 & \text{if } i = \frac{h}{2} \\ (-1,0) & \text{if } i > \frac{h}{2} \end{cases}, \tag{16}$$

resp.

$$\lfloor \mathbf{C}_{2\to m} \rfloor_{i,j} \in \begin{cases} (-1,0) & \text{if } i < \frac{h}{2} \\ 0 & \text{if } i = \frac{h}{2} \\ (0,1) & \text{if } i > \frac{h}{2} \end{cases}, \tag{17}$$

where $\lfloor \cdot \rfloor_{i,j}$ is the entry in i-th row and j-th column.

Proof 1 *We first decompose c_m defined in Eq. (14) into*

$$c_m = \frac{s-\kappa}{h} \cdot \frac{\gamma(h - \gamma\pi_v)}{h} \triangleq c_m^a \cdot c_m^b. \tag{18}$$

Taking the case of $i < \frac{h}{2}$ in Eq. (16) as an example, i.e., *$0 \le \kappa < \frac{h}{2}$. Since the GS image corresponding to the middle scanline is to be restored, $s = \frac{h}{2}$. Thus, $0 < c_m^a \le \frac{1}{2}$. In practice, π_v indicates the prediction of the inter-frame vertical optical flow value (see Eq. (1)), which is usually much smaller than the total number of image scanlines h. Also, the readout time ratio γ is less than or equal to 1 for a normal RS camera [13, 14, 41], so $\gamma \in (0, 1]$ when modeling the RS-aware forward warpings. We thus arrive at $0 < c_m^b < 2$. To sum up, $c_m \in (0, 1)$,* i.e., *$\lfloor \mathbf{C}_{1\to m} \rfloor_{i,j} \in (0, 1)$ holds when $i < \frac{h}{2}$. Other cases can be proved similarly. Note that $\gamma \in [-1, 0)$ when modeling the RS-aware backward warpings.*

5. RS Temporal Super-Resolution

Since DiffSfM [41] struggles to efficiently and accurately estimate the correlation factor defined in Eq. (14), requiring non-trivial readout calibration and cumbersome calculations, we propose to elaborate a concise and effective end-to-end CNN to mine the inherent regularity of data to account for Proposition 1. Inspired by the cascaded architecture for video interpolation in [4, 5, 15, 24], we carefully design an RSSR network to encapsulate the complete underlying RS geometry, as illustrated in Fig. 3. Our network takes two consecutive RS images $\mathbf{I}_r^1$ and $\mathbf{I}_r^2$ as input and predicts the GS image corresponding to any scanline (*i.e.*, generating a high framerate GS video). The proposed RSSR pipeline can be distilled down to two main submodules: the optical flow estimation network $\mathcal{F}$ and the middle-scanline RS undistortion flow estimation network $\mathcal{U}$. We first utilize $\mathcal{F}$ to obtain bidirectional optical flows, and then encode the relation between optical flows and middle-scanline RS undistortion flows by the middle-scanline correlation maps over $\mathcal{U}$. Finally, we compute the middle-scanline RS undistortion flows to produce two target middle-scanline GS frames by the softmax splatting [24]. Below we describe the details of each sub-network.

Optical flow estimator. We employ the classical PWC-Net [34] as our optical flow estimation network $\mathcal{F}$. As learning optical flow without ground truth supervision is extremely difficult, we fine-tune $\mathcal{F}$ on the RS benchmarks from the pre-trained model of PWC-Net in a self-supervised manner [21, 38].

Middle-scanline RS undistortion flow estimator. We adopt the hourglass architecture [15, 31] as our backbone network $\mathcal{U}$ to predict the correlation maps, modifying the last layer to output 6-channel predictions. To comply with Proposition 1, we fetch the first two channels in the output

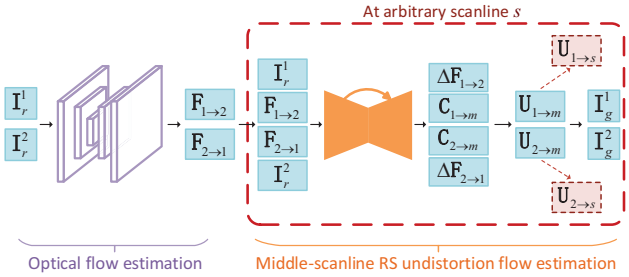

Figure 3: Overview of our RSSR network architecture. Given two input consecutive RS frames, we first estimate the bidirectional optical flows. Then, we use the UNet architecture associated with Eq. (19) to resolve the correlation maps. Next, the middle-scanline RS undistortion flows can be calculated explicitly by Eq. (20), while being certifiable. Finally, we adopt softmax splatting to generate the target middle-scanline GS frames. Note that our main network is designed to predict the latent GS images corresponding to the middle scanline during training. In particular, in the test phase, the RS undistortion flows for any scanline $s \in [0, h-1]$ can be propagated through Eq. (15) (dashed arrow), followed by the recovery of the GS image corresponding to scanline s.

of network $\mathcal{U}$, followed by a *Sigmoid* operation to map to the interval of $(0, 1)$, resulting in $\hat{\mathbf{C}}_{1\to m}$ and $\hat{\mathbf{C}}_{2\to m}$. Afterward, inspired by [20], we multiply them with their respective normalized scanline offsets (*i.e.*, $\mathbf{T}_{1\to m}$ and $\mathbf{T}_{2\to m}$, $\lfloor \mathbf{T} \rfloor_{i,j} \in [-1, 1]$), which is defined as the normalized scanline offset between the captured pixel and that of the middle scanline. Note that $\mathbf{T}_{1\to m} = -\mathbf{T}_{2\to m}$, and their values in middle-scanline are set to be small enough (not to be zero) for subsequent flow propagation. The final correlation maps $\mathbf{C}_{1\to m}$ and $\mathbf{C}_{2\to m}$ thus can be recovered as:

$$\begin{aligned} \mathbf{C}_{1\to m} &= \mathbf{T}_{1\to m} \odot \hat{\mathbf{C}}_{1\to m}, \\ \mathbf{C}_{2\to m} &= \mathbf{T}_{2\to m} \odot \hat{\mathbf{C}}_{2\to m}, \end{aligned} \tag{19}$$

where $\odot$ is an element-wise multiplier. It is easy to verify that they all naturally fit for the theoretical bounds of Proposition 1. Furthermore, we use the last four channels of network $\mathcal{U}$ to estimate optical flow residuals $\Delta\mathbf{F}_{1\to2}$ and $\Delta\mathbf{F}_{2\to1}$ to enhance the alignment of edge details and the generality of the proposed model. Finally, according to Eq. (13), the bidirectional RS undistortion flows $\mathbf{U}_{1\to m}$ and $\mathbf{U}_{2\to m}$ can be obtained as:

$$\begin{aligned} \mathbf{U}_{1\to m} &= \mathbf{C}_{1\to m} \odot (\mathbf{F}_{1\to2} + \Delta\mathbf{F}_{1\to2}), \\ \mathbf{U}_{2\to m} &= \mathbf{C}_{2\to m} \odot (\mathbf{F}_{2\to1} + \Delta\mathbf{F}_{2\to1}). \end{aligned} \tag{20}$$

Finally, the softmax splatting [24] is used to warp the RS image $\mathbf{I}^1_r$ (resp. $\mathbf{I}^2_r$) to the target middle-scanline GS image $\mathbf{I}^1_g$ (resp. $\mathbf{I}^2_g$) through $\mathbf{U}_{1\to m}$ (resp. $\mathbf{U}_{2\to m}$).

Extend to GS images under arbitrary scanlines. As the resulting RS undistortion flows corresponding to different scanlines can be converted to each other, based on Eq. (15), we can propagate the middle-scanline RS undistortion flows obtained above to arrive at the RS undistortion flows $\mathbf{U}_{1\to s}$ and $\mathbf{U}_{2\to s}$ for any scanline $s \in [0, h-1]$. Then, the corresponding high framerate GS images can be warped from the original two RS images, in the same vein, by using softmax splatting, *i.e.*, achieving the rolling shutter temporal super-resolution in a temporally coherent manner.

6. Experiments

6.1. Implementation Details

Loss function. Our loss function $\mathcal{L}$ is a linear combination of the reconstruction loss $\mathcal{L}_r$, perceptual loss $\mathcal{L}_p$ [16], warping loss $\mathcal{L}_w$ and smoothness loss $\mathcal{L}_s$, *i.e.*,

$$\mathcal{L} = \mu_r \mathcal{L}_r + \mu_p \mathcal{L}_p + \mu_w \mathcal{L}_w + \mu_s \mathcal{L}_s, \tag{21}$$

where we set $\mu_r = 10$, $\mu_p = 1$, $\mu_w = 10$ and $\mu_s = 0.1$. The details can be found in our supplementary material.

Training dataset. We evaluate our model on the Carla-RS and Fastec-RS datasets [20] that provide ground truth (GT) middle-scanline GS supervisory signals. 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 [20], 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 test. We train our network on both two benchmarks to predict the middle-scanline GS images (*i.e.*, $s = h/2$). At the test time, our model is able to be extended to generate arbitrary corrected GS frames for any scanline $s \in [0, h-1]$.

Training strategy. Our pipeline is implemented in PyTorch. We use the Adam optimizer [17] to optimize the proposed network. The learning rate is initially set to 10^{-4} and decreases by a factor of 0.8 every 50 epochs. The optical flow estimation network $\mathcal{F}$ is first fine-tuned for 100 epochs from the pre-trained model of [34], and then the entire model is jointly trained for another 200 epochs. The batch size is set as 6. 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 warrant the scanline dependence of RS camera.

6.2. Evaluation Protocols

We compute the average Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [39] between

Figure 4: Visual comparisons on Fastec-RS testing set. We zoom in the correction results according to the blue boxes. While other methods cause various artifacts, our method produces best effects.

Table 1: Quantitative comparisons on recovering GS images corresponding to the first scanline of the second RS frame. The numbers in red and blue represent the best and second-best performance. Note that we cannot benchmark the Fastec-RS dataset due to its lack of training ground truth. Regardless of the black edges of corrected images, our approach performs favorably against other methods.

Method	PSNR↑			SSIM↑		LPIPS↓	
	CRM	CR	FR	CR	FR	CR	FR
DeepUnrollNet [20]	26.90	26.46	26.52	0.81	0.79	0.0703	0.1222
DiffHomo [42]	19.60	18.94	18.68	0.61	0.61	0.1798	0.2229
DiffSfM-*PWCNet* [41]	19.53	18.62	18.59	0.69	0.63	0.2042	0.2416
DiffSfM-*RAFT* [41]	24.20	21.28	20.14	0.78	0.70	0.1322	0.1789
RSSR (Ours)	30.17	24.78	21.26	0.87	0.78	0.0695	0.1424

predictions and GT GS images. Higher PSNR/SSIM or lower LPIPS scores indicate better performance. Note that *unless otherwise stated, all competing methods refer to the GS image corresponding to the first scanline of the second RS frame* for consistent comparisons. See our supplementary material for more instructions on DeepUnrollNet [20]. Since the Carla-RS dataset provides the GT occlusion masks, for better evaluation, we conduct quantitative experiments including: the Carla-RS dataset with occlusion mask (*CRM*), the Carla-RS dataset without occlusion mask (*CR*), and the Fastec-RS dataset (*FR*).

6.3. Ablation Studies

To demonstrate the effectiveness of each component in our proposed network, we evaluate the controlled comparisons over network $\mathcal{F}$, network $\mathcal{U}$, and loss function $\mathcal{L}$, respectively. We train these variations using the same strategy as aforementioned in Subsec. 6.1. Please refer to the supplementary materials for an in-depth analysis.

6.4. Comparisons with Existing Methods

We evaluate the proposed RSSR method against the following RS correction algorithms:

- **SMARSC** [43]: The state-of-the-art single-view RS correction method. We follow the reimplementation by [20] as it is not open-sourced.
- **DeepUnrollNet** [20]: The state-of-the-art two-view RS correction method. However, it has no ability to produce high framerate GS video sequences.
- **DiffSfM** [41] and **DiffHomo** [42]: Pure geometric RS correction methods. They are difficult to robustly estimate the accurate RS geometry from two consecutive frames and rely on complicated handling. Also, we implement two versions of DiffSfM by using PWC-Net [34] and RAFT [35] as inputs respectively.

We report the quantitative results in Table 1. The better optical flow obtained by RAFT can improve the estima-

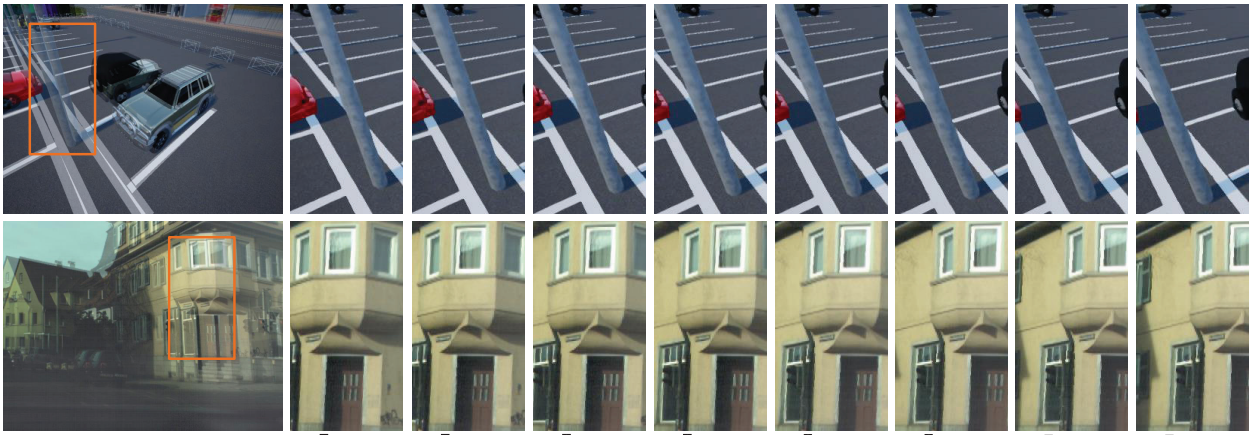

Figure 5: Interpolating GS video frames by applying our method. We zoom in the correction results according to the orange boxes. For example, "$\mathbf{I}_{1\to 0.3h}$" denotes the corrected GS image corresponding to $0.3h$-th scanline of frame 1. Our method not only preserves the temporal smoothness but also corrects the RS artifacts.

tion accuracy of RS geometry, thus promoting the performance of RS correction in DiffSfM, which also indicates the importance of mining RS geometry. Our model performs favorably against all the compared methods in the Carla-RS dataset and competes on par with DeepUnrollNet in the Fastec-RS dataset, since the Carla-RS dataset is more consistent with the constant motion assumption. Note that, compared with DeepUnrollNet, the black edges in our corrected GS images will reduce the PSNR score of our method without using masks. Very importantly, our method can generate high framerate and visually pleasing GS video frames. We also provide the visual results on images with noticeable RS distortions in Fig. 4, where DiffHomo and DiffSfM fail to correct the geometric RS distortions and DeepUnrollNet causes loss of local image details. More results are shown in our supplementary materials. Overall, our method can produce more correct and reliable results.

6.5. Generating Multiple GS Video Frames

We generate multiple GS video frames corresponding to different scanlines, as shown in Fig. 5. Our method not only produces smooth and continuous video sequences but also removes the RS artifacts successfully.

6.6. Inference Times

Our method can simultaneously predict two middle-scanline GS images with a resolution of 640×480 pixels in near real-time on an NVIDIA GeForce RTX 2080Ti GPU (average 0.12 seconds), which is faster than the average 0.34 seconds of [20] to restore a single GS image. Further, we can extend to generate a GS image corresponding to any certain scanline, using an average runtime of 1.75 milliseconds, because only simple explicit matrix operations are required. Therefore, our RSSR method can efficiently *produce 960 GS video frames in about 1.80 seconds*. DiffSfM [41], however, takes about 467.26 seconds to recover a single GS image on an Intel Core i7-7700K CPU, which is a disadvantage for time-constrained tasks, such as real-time robotic visual localization.

6.7. Limitations

The main limitation of our approach is that it is not robust to heavy occlusions and moving objects. This is a common challenge for RS correction algorithms [20,28,41–43]. In these challenging scenes, the RS undistortion flow of the corresponding region can not be well estimated, resulting in ambiguous object boundaries.

7. Conclusion

In this paper, we tackle the challenging task of RS image inversion, *i.e.*, converting consecutive RS images to high framerate GS images. To this end, we have confirmed and discussed the inherent connection between bidirectional RS undistortion flow and optical flow. We presented the first novel and intuitive RS temporal super-resolution framework that extracts a latent GS image sequence from two consecutive RS images, which is guided by the underlying geometrical properties of the problem itself. Our pipeline owns good interpretability and generalization ability due to RS geometry-aware learning. It does not require extra supervision other than the middle-scanline GT GS images. We have demonstrated that our approach can not only reconstruct geometrically and temporally consistent video sequences but also remove RS artifacts, achieving a favourable performance.

Acknowledgments

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

References

[1] 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. 2

[2] 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

[3] 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. 2

[4] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3703–3712. IEEE, 2019. 2, 5

[5] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: motion estimation and motion compensation driven neural network for video interpolation and enhancement. *Transactions on Pattern Analysis and Machine Intelligence*, 43(3):933–948, 2021. 2, 5

[6] 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

[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. 6

[8] Bin Fan, Yuchao Dai, and Mingyi He. Sunet: symmetric undistortion network for rolling shutter correction. In *Proceedings of the International Conference on Computer Vision*. IEEE, 2021. 3

[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. 2

[10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[11] 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. 2

[12] 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. 2

[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. 5

[14] Sunghoon Im, Hyowon Ha, Gyeongmin Choe, Hae-Gon Jeon, Kyungdon Joo, and In So Kweon. Accurate 3d reconstruction from small motion clip for rolling shutter cameras. *Transactions on Pattern Analysis and Machine Intelligence*, 41(4):775–787, 2018. 2, 5

[15] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: high quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 9000–9008. IEEE, 2018. 2, 5

[16] 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. 6

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

[18] 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, 2

[19] 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, 2

[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, 6, 7, 8

[21] Peidong Liu, Joel Janai, Marc Pollefeys, Torsten Sattler, and Andreas Geiger. Self-supervised linear motion deblurring. *Robotics and Automation Letters*, 5(2):2475–2482, 2020. 5

[22] Hugh Christopher Longuet-Higgins and Kvetoslav Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397, 1980. 3

[23] Marci Meingast, Christopher Geyer, and Shankar Sastry. Geometric models of rolling-shutter cameras. *arXiv preprint arXiv:cs/0503076*, 2005. 3

[24] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 5437–5446. IEEE, 2020. 2, 4, 5, 6

[25] Luc Oth, Paul Furgale, Laurent Kneip, and Roland Siegwart. Rolling shutter camera calibration. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1360–1367. IEEE, 2013. 3

[26] Alonso Patron-Perez, Steven Lovegrove, and Gabe Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *International Journal of Computer Vision*, 113(3):208–219, 2015. 3

[27] 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. 2

[28] 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, 8

[29] 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, 2

[30] Erik Ringaby and Per-Erik Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision*, 96(3):335–352, 2012. 1, 2

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015. 5

[32] Olivier Saurer, Kevin Koser, Jean-Yves Bouguet, and Marc Pollefeys. Rolling shutter stereo. In *Proceedings of the International Conference on Computer Vision*, pages 465–472. IEEE, 2013. 2, 3

[33] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2432–2439. IEEE, 2010. 4

[34] 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. 2, 4, 5, 6, 7

[35] Zachary Teed and Jia Deng. Raft: recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419. Springer, 2020. 4, 7

[36] 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. 1, 2

[37] 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. 2

[38] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 4884–4893. IEEE, 2018. 5

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 586–595. IEEE, 2018. 6

[40] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 9219–9228. IEEE, 2021. 3

[41] 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, 2, 3, 5, 7, 8

[42] 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, 2, 3, 5, 7, 8

[43] 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, 2, 3, 7, 8