

Towards Nonlinear-Motion-Aware and Occlusion-Robust Rolling Shutter Correction

Delin Qu^{1,2,*} Yizhen Lao^{4,*} Zhigang Wang^{2,*} Dong Wang² Bin Zhao^{2,3,†} Xuelong Li^{2,3,†}
¹Fudan University ²Shanghai AI Laboratory
³Northwestern Polytechnical University ⁴Hunan University

Abstract

This paper addresses the problem of rolling shutter correction in complex nonlinear and dynamic scenes with extreme occlusion. Existing methods suffer from two main drawbacks. Firstly, they face challenges in estimating the accurate correction field due to the uniform velocity assumption, leading to significant image correction errors under complex motion. Secondly, the drastic occlusion in dynamic scenes prevents current solutions from achieving better image quality because of the inherent difficulties in aligning and aggregating multiple frames. To tackle these challenges, we model the curvilinear trajectory of pixels analytically and propose a geometry-based **Quadratic Rolling Shutter (QRS)** motion solver, which precisely estimates the high-order correction field of individual pixels. Besides, to reconstruct high-quality occlusion frames in dynamic scenes, we present a 3D video architecture that effectively **Aligns and Aggregates** multi-frame context, namely, **RSA²-Net**. We evaluate our method across a broad range of cameras and video sequences, demonstrating its significant superiority. Specifically, our method surpasses the state-of-the-art by **+4.98**, **+0.77**, and **+4.33** of PSNR on *Carla-RS*, *Fastec-RS*, and *BS-RSC* datasets, respectively. Code is available at <https://github.com/DelinQu/qjsc>.

1. Introduction

The rolling shutter (RS) mechanism widely integrated in consumer video cameras continues to gather photos during the acquisition process, thus effectively increasing sensitivity [10]. The RS cameras donate CMOS sensors and scan the scene sequentially instead of instantly taking a snapshot of the entire scene, like the global shutter (GS) using the CCD sensor [19]. The time slot between consecutive scan lines causes motion artifacts called the RS effect, e.g., wob-

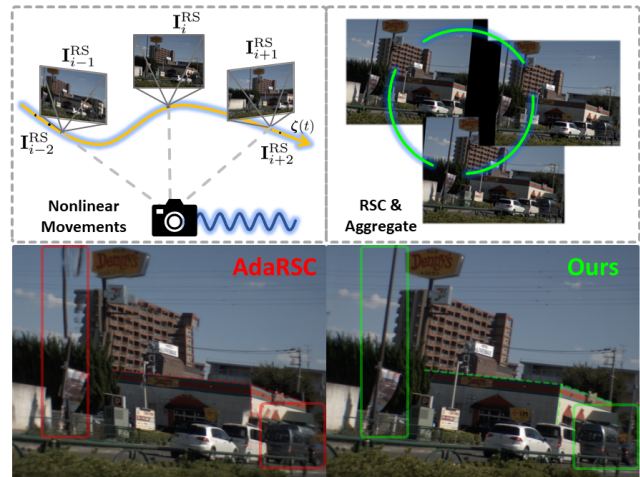


Figure 1: Illustration of the challenges in complex nonlinear motion and dynamic scenes with occlusion. The proposed method models the curvilinear trajectory and reconstructs the high-quality occlusion region from adjacent frames of the video stream. In contrast, the state-of-the-art fails to correct the pole and causes significant unaligned artifacts on the bottom right car.

ble and skew, under extreme motion conditions [1]. In addition to the detrimental visual artifacts, the RS effect damages numerous 3D vision algorithms based on GS assumptions, such as camera pose estimation [1, 4], structure-from-motion [34] and SLAM [20, 17]. Therefore, rolling shutter correction (RSC) is significant in photography and has attracted considerable research attention in the last decades.

Existing works on RS correction are generally categorized into single-frame and multi-frame methods. The single-frame methods are based on strict geometric assumptions [27, 24] or simplified camera motion [26, 15], thus obtaining unsatisfying results in complex scenes. In comparison, multi-frame methods are more sensible and achieve higher performance [36]. Typically, the correction field between two frames is estimated by a neural block [28] to recover the GS frame [18, 2, 7]. Nevertheless, existing RSC methods cannot produce satisfying results because of the

* Authors contributed equally: dlqu22@m.fudan.edu.cn

† Corresponding author

following limitations:

1) Complex higher-order motion: Current methods based on cost volume [18, 35, 30, 2] face challenges in estimating the accurate correction field since the field cannot be effectively supervised during training [2]. Besides, RSC solutions depending on optical flow use the uniform velocity assumption [5, 7, 25], ignoring the nonlinear movements. However, the motion in real scenes can be complex and variable, and the inaccuracy of correction fields will accumulate row by row and eventually lead to significant image correction errors *e.g.*, the distorted pole and house corrected by AdaRSC [2] in Fig. 1.

2) Scene occlusion: As shown in Fig. 1, object edge occlusion around the entity and image border occlusion prevent RSC solutions from better image synthetic quality. Despite the most recent multiple frames method [7, 2] trying to compensate for occluded pixels from consecutive frames, the visual performance cannot satisfy the regular application due to the inherent difficulties in aligning and aggregating multiple frames.

To address the challenges, we model the curvilinear trajectory of pixels analytically and propose a geometry-based **Quadratic Rolling Shutter (QRS)** motion solver, which precisely estimates the high-order correction field of individual pixel based on the forward and backward optical flows. Benefiting from the rigorous modeling of the RS mechanism, the QRS motion solver demonstrates a strong generalization performance across various datasets and handles RS temporal super-resolution tasks [5]. Besides, to reconstruct high-quality occlusion frames in extreme scenes, we present a 3D video architecture which effectively Aligns and Aggregates multi-frame context, namely, RSA²-Net. Tab. 1 exhibits the superiority of the proposed method, and our contributions can be summarized as follows:

- We analytically model the trajectory in complex nonlinear movements and present a novel geometry-based quadratic rolling shutter motion solver that precisely estimates the high-order correction field of individual pixels.
- We propose a self-alignment 3D video architecture for high-quality frame aggregation and synthesis against extreme scene occlusion.
- A broad range of evaluations demonstrates the significant superiority and generalization ability of our proposed method over state-of-the-art methods.

2. Related Work

Single-frame models. To simplify the RSC problem, many works apply different geometric assumptions, such as the straight lines kept straight in [27], vanishing direction restraint in [24, 23], and analytical 3D straight line RS

Table 1: Comparison of the proposed method vs. the state-of-the-art RSC solutions.

Method	DSfM [36]	DSUN [18]	JCD [35]	SUNet [6]	RSSR [5]	AdaRSC [2]	CVR [7]	Ours
Dynamic Scene		✓	✓	✓	✓	✓	✓	✓
Occlusion Scene							✓	✓
High-order Motion	✓							✓
Temporal Super-Resolution					✓		✓	✓

projection model in [15]. Besides, the simplified camera motion is also applied, for instance, the rotation-only model [26, 24, 15] and Ackerman model [24]. Moreover, the first learning-based model is proposed in [26] to remove RS from a single distortion image, and Zhuang *et al.* [38] further proposed Convolutional Neural Network (CNN)-based method to learning underlying geometry and recover GS image. Recently, Wang *et al.* [33] present an RS removal model with the global reset feature (RSGR). Nevertheless, single-frame models either rely on strong assumptions or depend on inconspicuous features, which causes an unsatisfactory performance.

Multi-frame models. Multi-frame methods can be categorized into classical and learning-based models. For the classical multi-frame methods, the works [9, 14] and [32, 37] modelled the motion between RS frames as a mixture of homography matrices for general unordered RS images and two consecutive frames, respectively. Qu *et al.* [25] proposed a pixel-wise pose-free correction solver that corrects a single image pixel parametrically by rescaling its corresponding optical flow vector. Besides, Zhuang *et al.* [36] developed a solution to estimate relative poses from two consecutive frames and recover GS images based on the differential epipolar constraint under constant velocity and acceleration. However, they either rely on simplified camera motion models or need prior lens calibration.

The learning model based on multiple frames can be divided into motion-field-based and flow-based methods. The former usually computes the cost volumes [28] by a correlation layer to obtain the motion field between two frames. For example, Liu *et al.* [18] designed a deep shutter unrolling network to predict the GS image from two consecutive RS Frames, and Fan *et al.* [6] present a pyramidal construction to recover the GS image. Besides, Zhong *et al.* [35] design an architecture with a deformable attention features fusion module handling both RS distortion and blur. Recently Cao *et al.* [2] propose an adaptive warping module to warp the RS features into the GS one with multiple displacement fields. Nevertheless, the motion-field-based methods indirectly estimate the correction field, needing adequate supervision[2]. As for flow-based methods, [5], and [7] formulate the RS undistortion flows under the constant velocity assumption indirectly and develop networks to recover RS undistortion from optical flow. However, current flow-based methods ignore the nonlinear movements, thus, fail in complex motion scenes.

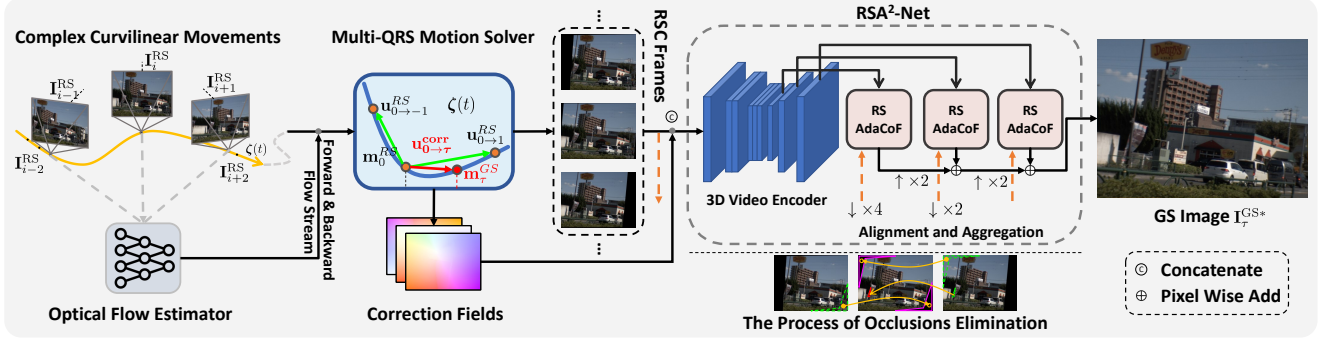


Figure 2: Overview of the proposed method. We aim to estimate precise correction fields in nonlinear motion with the QRS motion solver and synthesize high-quality frames against dynamic scenes with extreme occlusion by a self-alignment 3D video architecture RSA²-Net.

3. Methodology

3.1. Overview

Our method aims to estimate precise correction fields in complex nonlinear motion and synthesize high-quality frames against dynamic scenes with occlusion. As shown in Fig. 2, the proposed method receives a video stream, typically 5 frames, and precisely corrects the RS images with a geometry-based Multi-QRS motion solver. Then, a 3D video encoder-decoder architecture extracts the preliminary multi-scale features of corrected frames. After that, the hierarchical RSAdaCoF modules align and warp the features to produce a final synthetic high-quality GS frame.

3.2. RSC Formulation

As the rolling shutter mechanism shown in Fig. 3, the CMOS sensor scans the 3D scene sequentially, and every scanline holds a motion relevant to the row corresponding to the differential timestamp and readout time ratio γ . Previous work [18] proposes that the GS frame can be recovered by warping the RS features backwards with predicted displacement filed from GS to RS frame $U_{g \rightarrow r}$. However, every single timestamp τ between two consecutive frames corresponds to a GS frame, so it is more significant to model the RSC in the entire time series:

$$\mathbf{I}^{g(\tau)}(\mathbf{m}) = \mathbf{I}^r(\mathbf{m} + \mathbf{U}_{g(\tau) \rightarrow r}), \quad (1)$$

where $\mathbf{I}^{g(\tau)}$ denotes the potential GS frame at timestamp τ . \mathbf{I}^r is the source RS frame. \mathbf{m} is a specific pixel and $\mathbf{U}_{g(\tau) \rightarrow r}$ is the correction field from GS to RS frame. Relying on many established forward or backward warp techniques (e.g., bilinear interpolation, DFW [18], and softmax splatting [22]), existing methods achieve pleasing results for a given precise $\mathbf{U}_{g(\tau) \rightarrow r}$. Therefore, calculating the correction field from GS to RS frame is the critical factor of RSC.

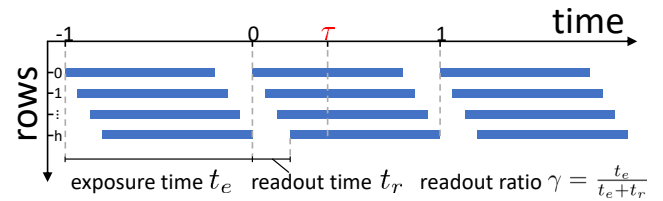


Figure 3: Illustration of the rolling shutter exposure mechanism among three consecutive frames. The interval time is normalized to $[0, 1]$, and our goal is to synthesize GS frames at any time $\tau \in [0, 1]$.

3.3. Quadratic Rolling Shutter Motion Solver

To address the correction field estimation problem under complex nonlinear motion, we focus on presenting a precise high-order motion solver for practical variable velocity and dynamic scenes. As the motion scheme shown in Fig 4, assuming a 3D point \mathbf{P} is projected to the consecutive three RS image planes \mathbf{I}_{-1}^{RS} , \mathbf{I}_0^{RS} and \mathbf{I}_1^{RS} as image points $\mathbf{m}_{-1}^{RS} = [x_{-1}^{RS}, y_{-1}^{RS}]^T$, $\mathbf{m}_0^{RS} = [x_0^{RS}, y_0^{RS}]^T$, and $\mathbf{m}_1^{RS} = [x_1^{RS}, y_1^{RS}]^T$, respectively. Despite the fact that the time slot between consecutive frames is extremely temporary, pixels may still follow complex curvilinear movements $\zeta(t)$. According to the derivative definition, we use the second-order Taylor expansion around t_0 to formulate as:

$$\zeta(t) \approx \zeta(t_0) + \dot{\zeta}(t_0)(t - t_0) + \frac{1}{2}\ddot{\zeta}(t_0)(t - t_0)^2. \quad (2)$$

Considering the trajectories from \mathbf{m}_0^{RS} to \mathbf{m}_{-1}^{RS} and \mathbf{m}_1^{RS} , respectively, we obtain:

$$\begin{bmatrix} (\zeta(t_{-1}) - \zeta(t_0))^T \\ (\zeta(t_1) - \zeta(t_0))^T \end{bmatrix} = \begin{bmatrix} (t_{-1} - t_0) & \frac{(t_{-1} - t_0)^2}{2} \\ (t_1 - t_0) & \frac{(t_1 - t_0)^2}{2} \end{bmatrix} \mathbf{M}, \quad (3)$$

$$\mathbf{M} = [\dot{\zeta}(t_0) \quad \ddot{\zeta}(t_0)]^T,$$

where t_{-1}, t_1 are the timestamp of \mathbf{m}_{-1}^{RS} and \mathbf{m}_1^{RS} . The matrix \mathbf{M} of shape 2×2 measures the quadratic motion of

pixels $\mathbf{m}^{\text{RS}}(t)$ at t . Note that the optical flow is the pattern of apparent motion of image objects between two consecutive frames. Thus, the differences of trajectory $\zeta(t)$ are equivalent to the flow vectors $\mathbf{u}_{0 \rightarrow -1}^{\text{RS}}$ from \mathbf{m}_0^{RS} to $\mathbf{m}_{-1}^{\text{RS}}$ and $\mathbf{u}_{0 \rightarrow 1}^{\text{RS}}$ from \mathbf{m}_0^{RS} to \mathbf{m}_1^{RS} .

$$\mathbf{u}_{0 \rightarrow -1}^{\text{RS}} = \zeta(t_{-1}) - \zeta(t_0), \mathbf{u}_{0 \rightarrow 1}^{\text{RS}} = \zeta(t_1) - \zeta(t_0). \quad (4)$$

According to the scanning mechanism illustrated in Fig. 3, the relative time can be expressed as:

$$\begin{aligned} t_{0 \rightarrow -1} &= t_{-1} - t_0 = -1 + \frac{\gamma}{h}(y_{-1}^{\text{RS}} - y_0^{\text{RS}}), \\ t_{0 \rightarrow 1} &= t_1 - t_0 = 1 + \frac{\gamma}{h}(y_1^{\text{RS}} - y_0^{\text{RS}}). \end{aligned} \quad (5)$$

Given the optical flow vectors $\mathbf{u}_{0 \rightarrow -1}^{\text{RS}}$, $\mathbf{u}_{0 \rightarrow 1}^{\text{RS}}$ and the relative times $t_{0 \rightarrow -1}$, $t_{0 \rightarrow 1}$, we can solve the square system of linear equations in Eq. (3) and obtain the quadratic motion matrix \mathbf{M} efficiently. Thus, we faithfully compute the correction vector $\mathbf{u}_{0 \rightarrow \tau}^{\text{corr}}$ from \mathbf{m}_0^{RS} to any timestamp τ by using:

$$\begin{aligned} \mathbf{u}_{0 \rightarrow \tau}^{\text{corr} \top} &= \begin{bmatrix} t_{0 \rightarrow \tau} & \frac{t_{0 \rightarrow \tau}^2}{2} \end{bmatrix} \mathbf{M}, \\ t_{0 \rightarrow \tau} &= \tau - t_0 = \tau - \frac{\gamma}{h} y_0^{\text{RS}}. \end{aligned} \quad (6)$$

Prime QRS motion solver: By given three consecutive RS images $\mathbf{I}_{-1}^{\text{RS}}$, \mathbf{I}_0^{RS} and \mathbf{I}_1^{RS} , the RS distribution point \mathbf{m}_0^{RS} can be corrected to corresponding GS image measurement $\mathbf{m}_\tau^{\text{GS}}$ at any timestamp τ by following transform:

$$\begin{aligned} \mathbf{m}_\tau^{\text{GS}} &= \mathbf{m}_0^{\text{RS}} + \mathbf{u}_{0 \rightarrow \tau}^{\text{corr}} \\ &= \mathbf{m}_0^{\text{RS}} + \mathbf{M}^\top \begin{bmatrix} t_{0 \rightarrow \tau} & \frac{t_{0 \rightarrow \tau}^2}{2} \end{bmatrix}^\top. \end{aligned} \quad (7)$$

With the precise correction field calculated by QRS motion solver, RS frame \mathbf{I}_0^{RS} can be effectively restored to corrected frame $\mathbf{I}_\tau^{\text{RSC}}$, by adapting a general bilinear interpolation technique mentioned in Sec. 3.2. As the sample $\mathbf{I}_{\tau,0}^{\text{RSC}}$ (green box) shown in Fig. 5, the proposed prime QRS motion solver is significantly superior to the state-of-the-art [7] according to the visual comparison result without considering image border occlusion.

3.4. Self-aligning Multi-QRS motion solver

Although the prime QRS motion solver accurately corrects RS pixels to GS pixels, the object edge occlusion and image border occlusion in Fig. 1 cannot be restored by the solver. Existing works attempt to compensate these occluded pixels from consecutive two frames [7] by a masked linear aggregation or three [2] frames via a self-attention-based warping module. Nevertheless, except for accurate corrected vector field estimation, they suffer from inherent difficulties in aligning and aggregating multiple frames.

Multi-QRS motion solver: In order to address the above problem, we design a self-aligning algorithm by extending

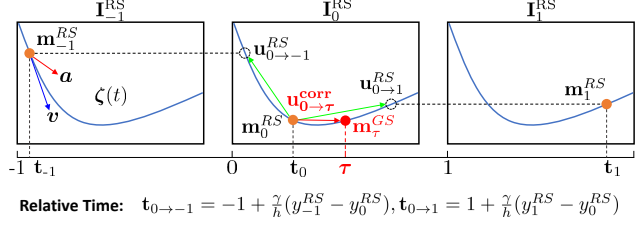


Figure 4: Illustration of the quadratic motion solver mechanism among three consecutive RS frames. The object moves at variable velocity and has a complex curvilinear trajectory. Note that the object has only 2 Dof in the image plane, and the time interval between frames is very small. Thus, we use a quadratic motion model to formalize the curvilinear trajectory of the pixel.

the QRS motion solver to a video sequence. Considering a consecutive RS video sequence $\mathbf{I}_{\{0,1,\dots,N-1\}}$ consisting of N frames, we aim to correct all the pixels \mathbf{m}_i^{RS} from the neighbour frames \mathbf{I}_i^{RS} , $\mathbf{I}_{i+1}^{\text{RS}} \in \mathbf{I}_{\{0,1,\dots,N-1\}}$ to time τ , usually in the intermediate, for occlusion elimination and obtain $\mathbf{I}_{\tau,i}^{\text{RSC}}$. Replacing -1 , 0 and 1 with $i-1$, i and $i+1$ at Eq. (2-6) in order we obtain:

$$\begin{aligned} \mathbf{m}_{\tau,i}^{\text{GS}} &= \mathbf{m}_i^{\text{RS}} + \mathbf{M}_i^\top \begin{bmatrix} t_{i \rightarrow \tau} & \frac{t_{i \rightarrow \tau}^2}{2} \end{bmatrix}^\top, \\ t_{i \rightarrow \tau} &= \tau - t_i = \tau - \frac{\gamma}{h} y_i^{\text{RS}}, i \in [1, N-2], \end{aligned} \quad (8)$$

where $t_{i \rightarrow \tau}$ is the relative time from time t_i to τ . The matrix \mathbf{M}_i denotes the quadratic motion of \mathbf{m}_i^{RS} . And the $\mathbf{m}_{\tau,i}^{\text{GS}}$ is the corrected GS pixel from $\mathbf{m}_i^{\text{RS}} = [x_i^{\text{RS}}, y_i^{\text{RS}}]^\top$. The Multi-QRS motion solver corrects pixels on different RS frames and naturally aligns pixels to time τ by Eq. (8), which accurately models the timeline of the consecutive frames. As shown in Fig. 5, RS frames $\mathbf{I}_{-1}^{\text{RS}}$, \mathbf{I}_0^{RS} and \mathbf{I}_1^{RS} are warped to time τ to obtain $\mathbf{I}_{\tau,-1}^{\text{RSC}}$, $\mathbf{I}_{\tau,0}^{\text{RSC}}$ and $\mathbf{I}_{\tau,1}^{\text{RSC}}$. By contributing a simple average fusion, we achieve a seamless and complete composite overlaid image.

3.5. 3D Video RSA²-Net

Given the corrected frames $\mathbf{I}_{\tau,1}^{\text{RSC}}$, $\mathbf{I}_{\tau,2}^{\text{RSC}}, \dots, \mathbf{I}_{\tau,N-2}^{\text{RSC}}$, we aim to eliminate extreme occlusions and synthesize a high-quality GS frames $\mathbf{I}_\tau^{\text{GS*}}$ through the multi-frame fusion function:

$$\mathbf{I}_\tau^{\text{GS*}} = f(\mathbf{I}_{\tau,1}^{\text{RSC}}, \mathbf{I}_{\tau,2}^{\text{RSC}}, \dots, \mathbf{I}_{\tau,N-2}^{\text{RSC}}; \boldsymbol{\theta}), \quad (9)$$

where $\boldsymbol{\theta}$ is the parameters of function f . We developed a 3D RS video encoder-decoder architecture RSA²-Net, which receives a video RSC stream and correction fields, to seek the fusion function f for extreme occlusions elimination and high-quality GS frames synthetic.

Model Architecture: As shown in Fig. 2, the RSA²-Net comprises a 3D video encoder (which is a 3D-transformer

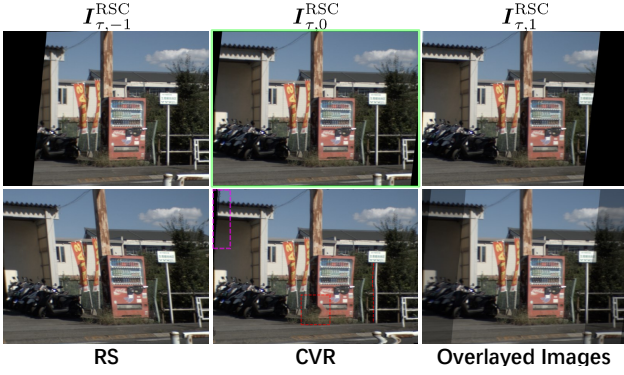


Figure 5: An example of the proposed Multi-QRS motion solver correcting from 5 frames. The overlapping image is directly average from three RSC image pixels.

in our model), the sequentially arranged Decode Layer, and iterative RSAdaCof modules. The encoder receives the sequence input of shape $B \times T \times C \times H \times W$, which is concatenated from consecutive frames and the corresponding correction field $\mathbf{u}_{i \rightarrow \tau}^{\text{corr}}$ obtained by the Multi-QRS motion solver. Then 3D video encoder encodes the input in four stages to obtain the multi-scale features, where B, C, T, H , and W respectively denote the batch size, channel, time, height, and width dimensions. The Encoded features are decoded with three times $2 \times$ upsampling, and then hierarchically warped by the RSAdaCof model at scale l , where $l \in \{1, 2, 3\}$, to produce a final synthetic high-quality GS Frame $\mathbf{I}_{\tau}^{\text{GS}*}$.

RSAdaCof Warping: Multi-frames are aligned to a specific time τ , as shown in Fig. 6. However, slight horizontal or vertical offsets might exist between the RSC pixel and GS pixel in extreme motion, which is not fully modeled by the QRS motion solver. Inspired by the AdaCof [16] and [29], we adopt three CNN-based offset Nets to obtain the per-pixel deformable kernels, which represent offsets $\alpha_i^l(k, x, y)$ and $\beta_i^l(k, x, y)$, and the weights $W_i^l(k, x, y)$, for frame $\mathbf{I}_{\tau, i, l}^{\text{RSC}}$ at scale l . Thus, the intermediate pixel $O_i^l(x, y)$ at position $[x, y]^T$ of frame $\mathbf{I}_{\tau, i, l}^{\text{RSC}}$ is:

$$O_i^l(x, y) = \sum_{k=1}^K W_i^l(k, x, y) \mathbf{I}_{\tau, i, l}^{\text{RSC}}(x + \alpha_i^l(k, x, y), y + \beta_i^l(k, x, y)), \quad (10)$$

where K is the number of sampling locations of each kernel. To align and aggregate the incomplete pixel region and the complete pixel region shown in Fig. 6, we use a CNN with softmax layer to predict the mask M_i^l corresponding to frame $\mathbf{I}_{\tau, i, l}^{\text{RSC}}$. However, considering the RS scanline mechanism, different pixels hold individual relative positions, and the quadratic model degrades gradually with increasing temporal distance (imagine using RS Frames at t to recover GS frames at infinity time τ). Thus, we propose to dilute the degradation using the nearest neighbor principle, which uses completely independent differential temporal distances

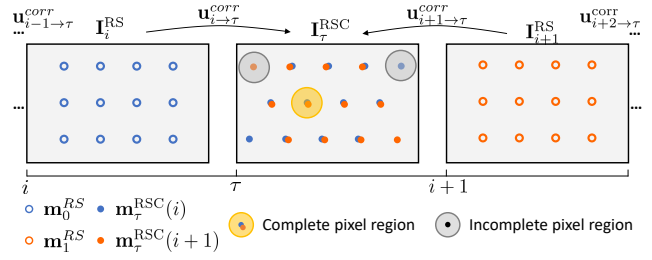


Figure 6: Illustration of the multi-frames alignment and fusion scheme. Slight offsets exist between the aligned pixels in extreme motion, and incomplete regions cover only pixels from the individual frame.

(time grid tg^l) to weigh the intensities of different pixels:

$$O^l = \sum_i \left(1 - \frac{tg^l}{S_{tg}^l}\right) M_i^l \cdot O_i^l, \quad tg^l = \left| \tau + \left\lfloor \frac{N-2}{2} \right\rfloor - i - \frac{\gamma}{h^l} y^l \right|, \quad (11)$$

$$S_{tg}^l = \sum_i^{N-2} tg^l, \quad i \in [1, N-2],$$

where h^l and y^l denote the image height and differential time grid at scale l . Note that the BS-RSC [2] dataset retains the $\gamma = 0.45$, meaning there are 55% blank rows between two consecutive frames. And this ultra-long time distance is a huge challenge for network training, so in general, we fix $\gamma = 1$ in RSAdaCof for rapid convergence.

3.6. Loss Functions

Following [35, 2], we use the Charbonnier loss \mathcal{L}_c and perceptual loss \mathcal{L}_p to ensure the synthetic visual quality, and the MSE \mathcal{L}_{mse} to avoid the extreme pixels. A simple but effective multi-loss balancing technique is adapted, *i.e.* scaling them to the same scale as the first. Thus total loss can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_c + \lambda_p \mathcal{L}_p + \lambda_{mse} \mathcal{L}_{mse},$$

$$\lambda_p = \frac{|\mathcal{L}_c|}{|\mathcal{L}_p|}, \quad \lambda_{mse} = \frac{|\mathcal{L}_c|}{|\mathcal{L}_{mse}|}. \quad (12)$$

4. Experiment

4.1. Experimental Setup

Datasets. We evaluate the proposed RSC method on the Carla-RS, Fastec-RS [18], BS-RSC [2], and ACC datasets. The synthetic Carla-RS contains general 6-DoF camera motions, and Fastec-RS holds multiple challenging dynamic scenes. Besides, the lately released real-world dataset BS-RSC [2] consists of nonlinear movements. Moreover, we derived the ACC dataset, which comprises more challenging variable movements, by excluding frames with constant general motions from the BS-RSC dataset (details can be found in the supplementary material). In addition, we provide more visual comparisons on other datasets in the supplementary material, including GPark [11], Seq77 [13], 3GS and House [8].

Table 2: Quantitative comparison against the state-of-the-art RSC methods on the Carla-RS dataset.

Method	PSNR \uparrow (dB)		SSIM \uparrow (dB)	LPIPS \downarrow
	CRM	CR	CR	CR
DSfM [36]	24.20	21.28	0.775	0.1322
DiffHomo [37]	19.60	18.94	0.606	0.1798
DSUN [18]	26.90	26.46	0.807	0.0703
SUNet [6]	29.28	29.18	0.850	0.0658
RSSR [5]	30.17	24.78	0.867	0.0695
VideoRS [21]	31.84	31.43	0.919	—
CVR [7]	<u>32.02</u>	<u>31.74</u>	<u>0.929</u>	<u>0.0368</u>
Ours	37.00	32.01	0.933	0.0253

Evaluation Metrics and Comparison methods. We use the PSNR, SSIM, and LPIPS to evaluate the quantitative results of RSC methods. In Sec. 4.2, we report the Carla-RS with the mask (CRM), Carla-RS without the mask (CR), Fastec-RS (FR), BS-RSC (BR), and ACC. Note that to force our method to fully integrate information among adjacent frames, the GS frames corresponding to the middle row are donated as supervision, and τ is set to 0.5 γ . We compare the proposed method with state-of-the-art RSC methods, including geometry-based methods **DSfM** [36], **DiffHomo** [37], and learning-based methods **DSUN** [18], **SUNet** [6], **JCD** [35], **RSSR** [5], **VideoRS** [21], **CVR** [7] and **AdaRSC** [2]. The models of most methods are unavailable on BS-RSC as it’s recently released, so we collected the models of DSUN and JCD provided by Cao *et al.* [2] and trained CVR [7] model on BS-RSC dataset using official released code. (Experiments of temporal super-resolution can be found in the supplementary material).

Implementation Details. We use RAFT [31] and GMA [12] of OpenMMLab Optical Flow Toolbox [3] to predict optical flow from the $N = 5$ consecutive RS frames, followed by Multi-QRS motion solver to predict the correction fields and obtain the latent occluded three RSC frames. Besides, we set the image readout ratio γ for Carla-RS, Fastec-RS, BS-RSC, and ACC datasets to 1.0, 1.0, 0.45, and 0.45, respectively, based on the intrinsic parameters of each dataset. The model is trained for 80 epochs using the Adam optimizer with learning rate $1e^{-4}$, batch size 4, and StepLR scheduler with gamma 0.1 and step size 25.

4.2. Quantitative Analysis

Results on Carla-RS and Fastec-RS. The results reported in Tab. 2 and Tab. 3 show that the proposed method outperforms the other eight RSC methods by large margins on both datasets, especially 37.00 (PSNR) compared to 32.02 on CRM and 0.0814 (LPIPS) against 0.1107 on FR achieved by CVR [7]. These superior performances significantly demonstrate the effectiveness of our model on general 6 Dof and highly dynamic scenes with occlusion.

Table 3: Quantitative comparison against the state-of-the-art RSC methods on the Fastec-RS dataset.

Method	PSNR \uparrow (dB)	SSIM \uparrow (dB)	LPIPS \downarrow
DSfM [36]	20.14	0.701	0.1789
DiffHomo [37]	18.68	0.609	0.2229
DSUN [18]	26.52	0.792	0.1222
SUNet [6]	28.34	0.837	0.1205
JCD [35]	24.84	0.778	0.1070
RSSR [5]	21.23	0.776	0.1659
VideoRS [21]	28.57	0.844	—
AdaRSC [2]	28.56	<u>0.855</u>	0.0793
CVR [7]	<u>28.72</u>	0.847	<u>0.1107</u>
Ours	29.49	0.872	0.0814

Table 4: Quantitative comparison against the state-of-the-art RSC methods on the BS-RSC and ACC datasets.

Method	PSNR \uparrow (dB)		SSIM \uparrow (dB)		LPIPS \downarrow	
	BR	ACC	BR	ACC	BR	ACC
DSfM [36]	19.80	15.74	0.698	0.551	0.2437	0.2544
DSUN [18]	23.60	22.39	0.808	0.780	0.1035	0.1145
JCD [35]	24.86	23.73	0.820	0.808	0.1897	0.1961
CVR [7]	24.58	24.07	0.823	0.816	0.0795	0.0822
AdaRSC [2]	<u>29.17</u>	<u>28.73</u>	<u>0.896</u>	<u>0.892</u>	<u>0.0617</u>	<u>0.0637</u>
Ours	33.50	33.36	0.946	0.945	0.0299	0.0303

Results on BS-RSC and ACC. The quantitative comparison on the real-world curvilinear movement BS-RSC and ACC datasets is shown in Tab. 4. The proposed method dramatically surpasses the existing five RSC methods with a PSNR score of 33.50 and an SSIM score of 0.946 compared to 29.17 and 0.896 achieved by the second-best entry AdaRSC on the BS-RSC dataset. Note that all the other methods produce a significant drop when testing on ACC, but our method remains stable. It demonstrates the effectiveness and robustness of our method in handling complex motion and acceleration trajectories.

4.3. Visual Comparisons

Dynamic and occlusion scenes: The comparisons of dynamic and highly occluded scenes are reported in Fig. 7 containing various moving objects with different depths and motions. Only the proposed method rectifies the poles back to the right position naturally, while the others either fail in correction (DSfM and DSUN) or produce artifacts on the occlusion areas (AdaRSC and CVR), although CVR was designed to handle occlusion.

Nonlinear Movements: As the results shown in Fig. 8, the proposed method precisely restored the GS images in curvilinear motion. Similarly, all existing RSC solutions fail in such a nonlinear scene because of the incorrectly estimated correction fields. For example, the tree on the first row and

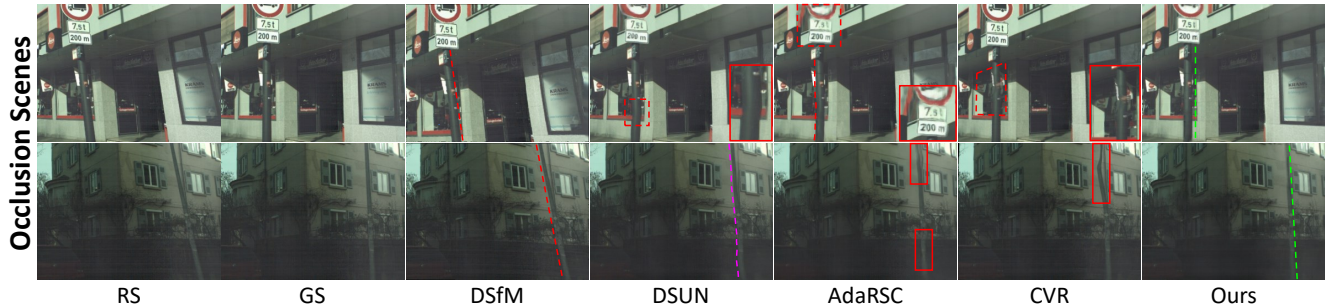


Figure 7: Visual comparison against the state-of-the-art RSC methods in dynamic and occlusion scenes on Fastec-RS datasets.

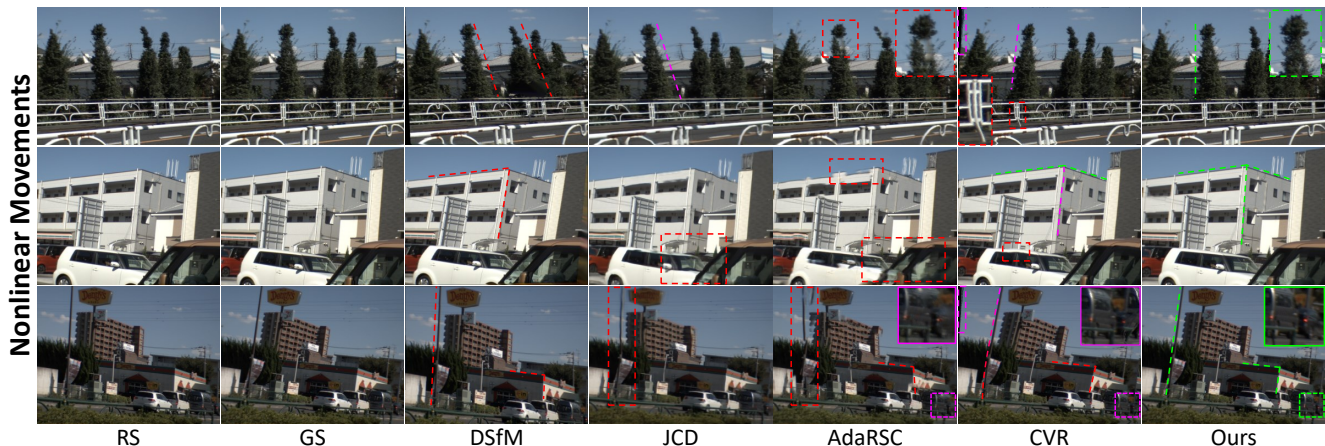


Figure 8: Visual comparison against the state-of-the-art RSC methods in nonlinear motion scenes of BS-RSC dataset.

pole on the third row are inaccurately corrected by DSfM, JCD and CVR, and AdaRSC produces noticeable artifacts. Besides, the sota methods CVR and AdaRSC cause significant unaligned shadow when handling occlusion in nonlinear scenes, *e.g.*, the bottom right car of the third row (more results can be found in the supplementary material).

4.4. Generalization Ability

To validate the generalization capability of the proposed method, we performed cross-tests on three datasets and donated a relative decline rate $rde(i, k)$ for evaluation:

$$rde_{i,k} = 1 - \frac{score_{i,k}}{score_{k,k}}, \quad (13)$$

where $rde_{i,k}$ and $score_{i,k}$ respectively denote the rde and metric score (PSNR, SSIM, and LPIPS), which is trained on dataset i and tested on dataset k . The confusion matrixes in Fig. 9 show that DSUN and CVR cannot accommodate numerous datasets, especially across the BS-RSC dataset. This is because the intrinsic readout ratio γ significantly changes from 1 (Carla-RS or Fastece-RS) to 0.45 (BS-RSC). In contrast, benefiting from the QRS motion solver, which analytical modeling of the RS mechanism, the

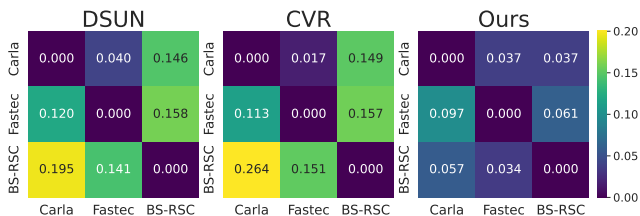


Figure 9: Generalization capability comparisons on SSIM of our method with existing RSC algorithms DSUN [18] and CVR [7] across Carla-RS, Fastec-RS and BS-RSC datasets.

proposed method demonstrates strong generalization performance with rde less than 0.1, despite the massive bias among three datasets.

5. Ablation Study

Linear Model vs Quadratic Model. This experiment compares the proposed quadratic and the linear model (the first-order form of QRS motion solver) with and without RSA²-Net f , respectively. Noting the solver generates RSC frames containing image occlusion, thus, it is more meaningful to focus on PSNR scores with Mask. As Fig. 10

Table 5: Ablation study results for RSAdaCof, correction field \mathbf{u} , 3D video encoder, number of input frames (NF).

Settings	PSNR↑ (dB)				SSIM↑ (dB)			LPIPS↓		
	CRM	CR	FR	BR	CR	FR	BR	CR	FR	BR
W/o correction field \mathbf{u}	36.00	30.92	28.72	32.36	0.922	0.872	0.938	0.0304	0.0878	0.0331
W/o RSAdaCof	35.90	30.84	28.70	32.75	0.923	0.863	0.943	0.0266	0.0881	0.0314
QRS motion solver + 3DUNet	36.15	31.06	28.51	33.41	0.929	0.865	0.946	0.0267	0.0827	0.0306
Full model	37.00	32.01	29.49	33.50	0.933	0.872	0.946	0.0253	0.0814	0.0299
3NF input	35.64	29.81	28.18	31.92	0.919	0.8530	0.942	0.0313	0.0912	0.0320
4NF input	35.95	30.98	28.26	32.40	0.925	0.8538	0.944	0.0282	0.0901	0.0303
5NF input	37.00	32.01	29.49	33.50	0.933	0.872	0.946	0.0253	0.0814	0.0299

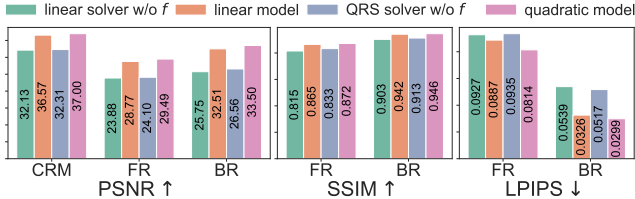


Figure 10: Ablation study of Linear and Quadratic Models.

shows, they all achieve high metric scores in the Carla-RS dataset. And the quadratic model outperforms the linear model, especially on the Fastec-RS and BS-RSC datasets containing complex variable velocity scenes, illustrating the validity of higher-order models under nonlinear motion.

QRS motion solver and RSA²-Net. We remove the RSA²-Net f or QRS motion solver from the pipeline to validate the capability of correction field estimation and occlusion elimination. As shown in Fig. 11, the QRS motion solver achieves ultra-high PSNR (32.31, CM), SSIM (0.913, BR), and lower LPIPS (0.0517, BR), and even outperform 32.02, 0.896 and 0.0617 achieved by the state-of-the-art. Besides, the single RSA²-Net performs fairly high metric scores on all three datasets, displaying the strong ability of content aggregation. In addition, the performance is significantly improved once the QRS motion solver and RSA²-Net are combined, especially on the Fastec-RS and BS-RSC datasets containing complex motion.

RSAdaCof and correction field. This experiment eliminated the RSAdaCof and correction field \mathbf{u} to validate their effectiveness separately. The results reported in Tab. 5 show significant performance degradation after removing module RSAdaCof due to the lack of alignment and aggregation capabilities. Similarly, our method achieves a lower score without correction field \mathbf{u} because it helps the model map RSC pixels back to RS planes.

3D-Transformer vs 3D-UNet. In this experiment, we further replace the 3D-Transformer in 3D Video RSA²-Net with a quite simple 3D-UNet to determine the effectiveness of the proposed QRS motion solver and RSAdaCof. The results in Tab. 5 show that even the simple 3D-UNet with QRS motion solver significantly outperforms the state-of-the-art



Figure 11: Ablation study of QRS solver and RSA²-Net.

methods and achieves similar scores with the Full model. It illustrates that our method does not rely on the ability of the Transformer architecture and confirms the effectiveness of the QRS motion solver and RSAdaCof module.

Number of Input Frames. We argue that exquisite alignment and fusion techniques based on the video sequence can solve the object edge and image border occlusion. According to the performance in Tab. 5, the evaluation score increases with the number of input frames, especially in extreme scenes. However, we did not test more than five frames due to computational limitations.

6. Conclusion and Limitation

This paper proposes a geometry-based quadratic rolling shutter motion solver and the 3D video stream-based structure RSA²-Net for RSC in complex nonlinear scenes with occlusion. A broad range of evaluations demonstrates the significant superiority of our proposed method over state-of-the-art methods. However, the proposed method requires dense matching between multiple consecutive frames, which can be expensive in some application scenes. This limitation is a common constraint among the most state-of-the-art RSC solutions. In future work, we aim to extend the QRS motion solver to sparse keypoint correction, serving for the 3D vision algorithm instead of the dense visual correction, which allows GS SfM/SLAM solutions to handle RS input in real time.

Acknowledgements. This work is supported by the Shanghai AI Laboratory, National Key R&D Program of China (2022ZD0160100) and the National Natural Science Foundation of China (62106183 and 62102145).

References

- [1] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. R6p-rolling shutter absolute pose problem. In *CVPR*, 2015. 1
- [2] Mingdeng Cao, Zhihang Zhong, Jiahao Wang, Yinqiang Zheng, and Yujiu Yang. Learning adaptive warping for real-world rolling shutter correction. In *CVPR*, 2022. 1, 2, 4, 5, 6
- [3] MMFlow Contributors. MMFlow: Openmmlab optical flow toolbox and benchmark. <https://github.com/open-mmlab/mmlflow>, 2021. 6
- [4] Yuchao Dai, Hongdong Li, and Laurent Kneip. Rolling shutter camera relative pose: generalized epipolar geometry. In *CVPR*, 2016. 1
- [5] Bin Fan and Yuchao Dai. Inverting a rolling shutter camera: Bring rolling shutter images to high framerate global shutter video. *ICCV*, 2021. 2, 6
- [6] Bin Fan, Yuchao Dai, and Mingyi He. Sunet: symmetric undistortion network for rolling shutter correction. In *ICCV*, 2021. 2, 6
- [7] Bin Fan, Yuchao Dai, Zhiyuan Zhang, Qi Liu, and Mingyi He. Context-aware video reconstruction for rolling shutter cameras. In *CVPR*, 2022. 1, 2, 4, 6, 7
- [8] Per-Erik Forssén and Erik Ringaby. Rectifying rolling shutter video from hand-held devices. In *CVPR*, 2010. 5
- [9] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *ICCP*, 2012. 2
- [10] James Janesick, Jeff Pinter, Robert Potter, Tom Elliott, James Andrews, John Tower, John Cheng, and Jeanne Bishop. Fundamental performance differences between CMOS and CCD imagers: part III. In *ASOS*, 2009. 1
- [11] Chao Jia and Brian L Evans. Probabilistic 3-d motion estimation for rolling shutter video rectification from visual and inertial measurements. In *MMSP*, 2012. 5
- [12] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021. 6
- [13] J. H. Kim, C. Cadena, and I. Reid. Direct semi-dense slam for rolling shutter cameras. In *ICRA*, 2016. 5
- [14] Yizhen Lao and Omar Ait-Aider. A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *CVPR*, 2018. 2
- [15] Yizhen Lao, Omar Ait-Aider, and Helder Araujo. Robustified structure from motion with rolling-shutter camera using straightness constraint. *Pattern Recognit Lett*, 2018. 1, 2
- [16] Hyeonmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *CVPR*, 2020. 5
- [17] Bangyan Liao, Delin Qu, Yifei Xue, Huiqing Zhang, and Yizhen Lao. Revisiting rolling shutter bundle adjustment: Toward accurate and fast solution. In *CVPR*, 2023. 1
- [18] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *CVPR*, 2020. 1, 2, 3, 5, 6, 7
- [19] M Meingast, C Geyer, and S Sastry. Geometric models of rolling-shutter cameras. In *OMNIVIS*, 2005. 1
- [20] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *T-RO*, 2015. 1
- [21] Eyal Naor, Itai Antebi, Shai Bagon, and Michal Irani. Combining internal and external constraints for unrolling shutter in videos. In *ECCV*, 2022. 6
- [22] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, 2020. 3
- [23] Pulak Purkait and Christopher Zach. Minimal solvers for monocular rolling shutter compensation under ackermann motion. In *WACV*, 2017. 2
- [24] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in manhattan world. In *ICCV*, 2017. 1, 2
- [25] Delin Qu, Bangyan Liao, Huiqing Zhang, Omar Ait-Aider, and Yizhen Lao. Fast rolling shutter correction in the wild. *TPAMI*, 2023. 2
- [26] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*, 2017. 1, 2
- [27] Vijay Rengarajan, Ambasadram N Rajagopalan, and Ranganarajan Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *CVPR*, 2016. 1, 2
- [28] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *CVPR*, 2011. 1, 2
- [29] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *CVPR*, 2022. 5
- [30] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 2
- [31] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 6
- [32] Subeesh Vasu, Mahesh Mohan MR, and AN Rajagopalan. Occlusion-aware rolling shutter rectification of 3d scenes. In *CVPR*, 2018. 2
- [33] Zhixiang Wang, Xiang Ji, Jia-Bin Huang, Shin'ichi Satoh, Xiaoping Zhou, and Yinqiang Zheng. Neural global shutter: Learn to restore video from a rolling shutter camera with global reset feature. *CVPR*, 2022. 2
- [34] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2011. 1
- [35] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *CVPR*, 2021. 2, 5, 6
- [36] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *ICCV*, 2017. 1, 2, 6
- [37] Bingbing Zhuang and Quoc-Huy Tran. Image stitching and rectification for hand-held cameras. In *ECCV*, 2020. 2, 6
- [38] Bingbing Zhuang, Quoc-Huy Tran, Pan Ji, Loong-Fah Cheong, and Manmohan Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *CVPR*, 2019. 2