Joint Appearance and Motion Learning for Efficient Rolling Shutter Correction

Bin Fan* Yuxin Mao* Yuchao Dai† Zhexiong Wan Qi Liu
School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China

Abstract

Rolling shutter correction (RSC) is becoming increasingly popular for RS cameras that are widely used in commercial and industrial applications. Despite the promising performance, existing RSC methods typically employ a two-stage *network structure that ignores intrinsic information interactions and hinders fast inference. In this paper, we propose a* single-stage *encoder-decoder-based network, named JAMNet, for efficient RSC. It first extracts pyramid features from consecutive RS inputs, and then simultaneously refines the two complementary information (i.e., global shutter appearance and undistortion motion field) to achieve mutual promotion in a joint learning decoder. To inject sufficient motion cues for guiding joint learning, we introduce a transformer-based motion embedding module and propose to pass hidden states across pyramid levels. Moreover, we present a new data augmentation strategy "vertical flip + inverse order" to release the potential of the RSC datasets. Experiments on various benchmarks show that our approach surpasses the state-of-the-art methods by a large margin, especially with a 4.7 dB PSNR leap on real-world RSC. Code is available at* `https://github.com/GitCVfb/JAMNet`.

1. Introduction

As commonly used image sensors in the automotive sector and motion picture industry, CMOS sensors offer particular benefits, including low cost and simplicity in design [15, 20, 42, 48]. The row-wise readout mechanism from top to bottom of electronic CMOS sensors, however, results in undesirable image distortions called the rolling shutter (RS) effect (also known as the jelly effect, *e.g*., wobble, skew) when a moving camera or object is in progress. Often, even a small camera motion causes visible geometric distortions in the captured RS image or video. Because of this, the RS effect inevitably becomes a hindrance to scene understanding and a nuisance in photography. As such, RS correction (RSC), as a way to make up for such deficiencies, is gradually gaining more and more attention [5, 11, 27, 29, 58].

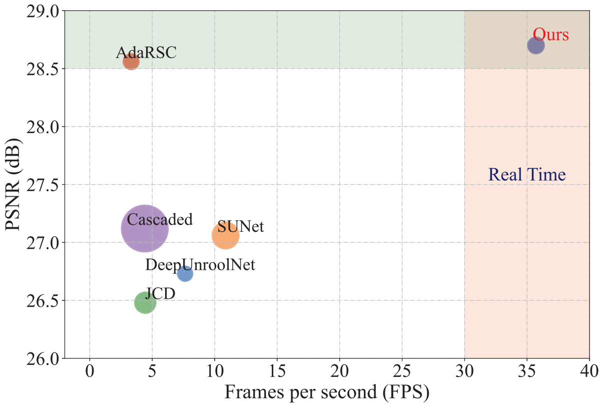


Figure 1. **Performance *vs*. Speed.** Each circle represents the performance of a model in terms of FPS and PSNR on the Fastec-RS testing set [29] with 640 × 480 images using a 3090 GPU. The radius of each circle denotes the model's number of parameters. Our method achieves state-of-the-art performance with real-time inferences and smaller parameters compared with prior RSC methods, including DeepUnrollNet [29], SUNet [10], JCD [56], AdaRSC [5], and Cascaded method (*i.e*., SUNet + DAIN [4]).

The RSC task aims to recover a latent distortion-free global shutter (GS) image corresponding to a specific exposure time between consecutive RS frames. The resulting RSC methods can be divided into traditional and deep learning-based ones. The traditional RSC methods [1, 13, 26, 38, 40, 48, 50] usually rely on hand-designed prior assumptions, geometric constraints, and complex optimization frameworks. Consequently, such processes are typically time-consuming and require complex parameter-tuning strategies for different scenarios, which restricts their real-world applications. In contrast, convolutional neural networks have also been used to remove RS artifacts in recent years due to the considerable success in many computer vision tasks, such as [5, 9, 14, 29, 51, 60]. Particularly, RSC methods based on multiple consecutive RS images have been heavily investigated [5, 10, 29, 56].

In general, these multi-image-based RSC approaches often consist of a *two-stage* network design with two key elements: a motion estimation module and a GS frame synthesis module, as illustrated in Fig. 2 (a). The former is dedicated to estimating a pixel-wise undistortion field, which is utilized to warp the RS appearance content to

[*] Equal contribution. † Corresponding author (`daiyuchao@gmail.com`).

the corresponding GS instance. The latter aims to fuse the contextual information in a coarse-to-fine manner, ultimately decoding the desired GS image. Although this two-stage idea sounds relatively straightforward, it suffers from several drawbacks. **First**, the two-stage RSC faces a classic "chicken-and-egg" problem: motion estimation and GS frame synthesis are inextricably linked; a high-quality undistortion field improves GS frame synthesis, and vice versa. Therefore, this step-by-step combination is not conducive to information interaction and joint optimization, resulting in a bottleneck for high-quality RSC. **Second**, the two modules are implemented by two independent encoder-decoders, ignoring the mutual promotion of these two key elements for RSC. **Third**, the two-stage network design inevitably increases the model size and inference time, which greatly limits their efficient deployment in practice.

To address these issues, we propose a novel *single-stage* solution for RS correction through Joint Appearance and Motion Learning (JAMNet). Our approach is a single encoder-decoder structure with coarse-to-fine refinements, as depicted in Fig. 2 (b), allowing the simultaneous learning of complementary GS appearance and undistortion motion information. After extracting hierarchical pyramid features, we design an efficient decoder for simultaneous occlusion inference and context aggregation. It leverages a warping branch to estimate the undistortion field to compensate for RS geometric distortions, while a synthetic branch is used to progressively refine the GS appearance, which forms a mutual promotion of complementary information. Among them, a hidden state is maintained to transmit additional cues across pyramid levels. Further, we propose to inject sufficient motion priors into the network at the coarsest level via a transformer-based motion embedding module. Moreover, inspired by the imaging principle of RS data, we also develop a new data augmentation strategy, *i.e. vertical flip + inverse order*, in the training process, to enhance the robustness of RSC models. Extensive experimental results demonstrate that our JAMNet significantly outperforms state-of-the-art (SOTA) RSC methods, especially achieving a real-time inference speed, as shown in Fig. 1. It is worth mentioning that our pipeline achieves a *4.7 dB PSNR improvement* on real-world RSC applications.

In a nutshell, our main contributions are summarized:

1) We propose a tractable single-stage architecture to jointly perform GS appearance refinement and undistortion motion estimation for efficient RS correction.
2) We develop a general data augmentation strategy, *i.e.*, vertical flip and inverse order, to maximize the exploration of the RS correction datasets.
3) Experiments show that our approach not only achieves SOTA RSC accuracy, but also enjoys a fast inference speed and a flexible and compact network structure.

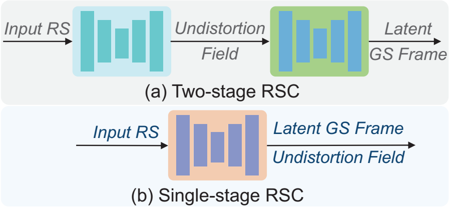


Figure 2. **Different RSC paradigms**. (a) The currently popular two-stage structure first estimates the undistortion field, and then completes GS recovery accordingly. (b) We propose a single-stage RSC framework with a joint learning mechanism to estimate the undistortion field and the latent GS frame at the same time.

2. Related Work

Geometric model based RS correction. The RS geometric model was first proposed in [33]. Subsequently, Dai *et al.* [7] derived a discrete RS epipolar constraint, while Zhuang *et al.* [58] presented a differential one. Very recently, Bai *et al.* [2] and Lao *et al.* [27] developed a scanline-homography and an RS-homography to perform plane-based RS correction, respectively. Notably, some works have been devoted to joint RS effect removal and other image processing tasks, such as super-resolution [3, 36], motion deblurring [32, 44], image stitching [59], and video stabilization [17, 52]. Furthermore, some additional assumptions are often applied to simplify the problem of RSC. For instance, the scene structure obeys the Manhattan world [38] or satisfies the straightness of straight lines [40], and the camera motion is purely rotational [16,26,37,40,41] or purely translational [3, 28]. Therefore, they cannot work well if these underlying assumptions on scene structures and camera motions do not hold. Also, these traditional methods are usually time inefficient, greatly limiting their practical applications.

Learning-based RS correction. In recent years, several appealing deep learning-based RSC methods [5, 34, 57, 60] have been developed, where a convolutional neural network (CNN) is trained to warp the RS frame to its GS counterpart. To reduce the ill-posedness of single-image RSC [39, 60], more attention has been paid to multi-image RSC. Liu *et al.* [29] took two adjacent RS frames as input and designed a deep shutter unrolling network to recover the latent GS frame. Afterward, Fan *et al.* [10] presented a symmetric undistortion network to aggregate contextual cues. Given three consecutive RS images, Zhong *et al.* [56] proposed to deal with RS effects and blurring in real-world distorted RS images simultaneously, and Cao *et al.* [5] put forward an adaptive warping strategy for coarse-to-fine refinement. Overall, these methods uniformly adopt a *two-stage* step-by-step learning framework, as shown in Fig. 2 (b), which is not conducive to information interaction and efficient inference. In contrast, we propose to learn complementary appearance and motion information simultaneously in a *single-stage* architecture, thus achieving *light-weight*, *real-*

time, and *high-accuracy* performance. To the best of our knowledge, our method is the first real-time RSC pipeline.

More generally, the recently popular RS temporal super-resolution (RSSR) [9, 14] can reconstruct a high frame-rate GS video from two consecutive RS frames. In theory, the latent GS image at any timestamp in the exposure period can be generated. However, RSSR requires supervision of GS ground-truth at multiple moments during training, which is more complicated than the RSC task since RSC only needs that at a single moment. Therefore, to be fair we will not compare with the RSSR method in this paper. Alternatively, the video frame interpolation (VFI) method can produce a GS video from two GS frames, such as [4,22,35]. However, they are tailored for GS cameras and cannot be applied immediately to RS images due to network defects [12].

Pixel-wise motion modeling. Correlation is designed to model the matching cost volume between data pairs and is often used to measure pixel-wise similarity in optical flow estimation tasks [8, 45, 47]. FlowNet [8] and PWC-Net [45] perform correlation on the local range and decode optical flow in a coarse-to-fine manner. While RAFT [47] uses all-pairs correlation followed by a correlation lookup to estimate optical flow within an iterative structure. Meanwhile, such a practice often migrates to RS correction for the undistortion field estimation [5, 10, 29, 56]. As the improvement of vision transformer, there are many works using a transformer, especially cross attention [23, 46, 53] for cross-view modeling. This process can help enhance feature representation through frame-wise correspondence modeling. Inspired by this, we also utilize the transformer with self- and cross-attentions as a motion embedding module to inject sufficient motion information into the network.

3. Method

Given two input RS frames R_0 and R_1 at adjacent time instances, our method aims to output a latent GS image corresponding to the exposure time of the middle scanline of the second RS image, consistent with [5, 29, 56]. For this purpose, we combine appearance and motion modeling in a single-stage architecture to achieve effective and efficient GS recovery. As shown in Fig. 3, our approach first extracts the pyramid features of the RS image pair by a weight-sharing encoder. Then, we employ a transformer-based motion embedding module to enhance the discrimination of the coarsest features for motion embedding. Finally, we develop an efficient decoder to simultaneously estimate the GS appearance and the motion field within a coarse-to-fine refinement framework.

3.1. Feature pyramid encoder

We construct an encoder to extract L-level feature maps: $\{F_0^l\}_{l=1}^L$, $\{F_1^l\}_{l=1}^L$. The feature pyramid does not include the input RS image because the bottom-level feature representations F_0^1 and F_1^1 have the same resolution as the input. Consistent with [10, 29, 56], a 7×7 convolutional layer is employed firstly, followed by a residual block [19] to extract full-resolution image features at the bottom level. The rest of the pyramid levels have a 3×3 convolutional layer with a stride of 2 for downsampling, also followed by a residual block. In particular, we attach a PReLU activation [18] after each 2D convolution. The parameters of the feature pyramid encoder are shared for R_0 and R_1. Ultimately, we produce a hierarchical feature representation, which facilitates subsequent coarse-to-fine joint appearance and motion decoding.

3.2. Transformer-based motion embedding module

Since the latent GS image is unknown, it is unreasonable to construct an explicit motion embedding by the cost volume between the RS features. Considering that the top-level decoder lacks guidance from prior motion, we adopt a standard yet trivial swin-transformer [30] to enhance the cross-correlation between the coarsest features F_0^L and F_1^L, aiming to implicitly inject motion information into the subsequent decoder. To this end, we first impose spatial information on the features by 2D position encoding P to mask the features as position-dependent. Then, we input the position encoded features $F_0^L + P$ and $F_1^L + P$ into the transformer layer to get the enhanced features $\bar{F}_0^L$ and $\bar{F}_1^L$. The transformer layer consists of self-attention, cross-attention, and a feed-forward network [49]. For self-attention, the query, key, and value are projections of the same feature. While for cross-attention, the query comes from another feature in the feature pair, which allows the module to focus on capturing mutual dependencies from features. This whole process is symmetrically performed for both F_0^L and F_1^L as follows:

$$\bar{F}_0^L = \mathcal{T}(F_0^L + P, F_1^L + P),\ \bar{F}_1^L = \mathcal{T}(F_1^L + P, F_0^L + P), \tag{1}$$

where $\mathcal{T}$ denotes a Transformer with the first input as query and the second as key and value.

3.3. Joint appearance and motion decoder

The core component in our proposed JAMNet is a multi-level decoder that jointly learns appearance and motion in a coarse-to-fine manner. It mainly contains two synchronized branches: warping-based and synthesis-based. The former gradually refines the motion fields to remove RS effects through a warping operation, facilitating higher-quality GS appearance restoration. The latter continuously synthesizes latent GS appearances, promoting more accurate motion estimation in turn. The joint action of the two branches enables the network to focus on context aggregation and occlusion reasoning, thereby achieving better RS effect removal. Additionally, we also exploit a hidden state to pass the rich appearance information as well as motion information across pyramid levels.

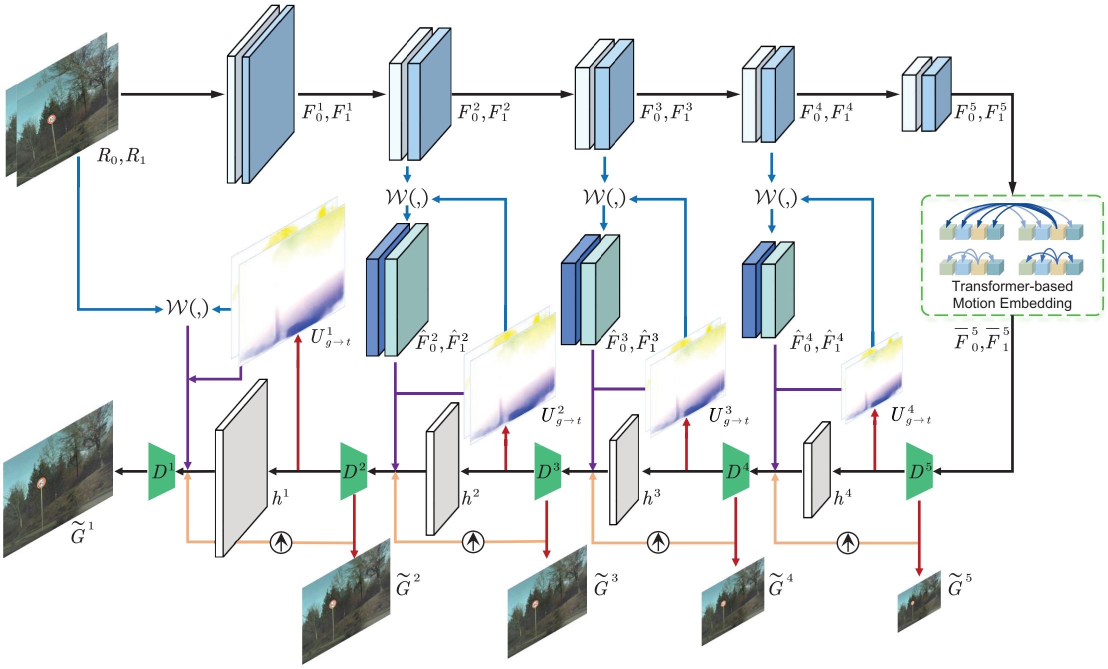


Figure 3. **Overall architecture of our JAMNet.** It has three main processes: a feature pyramid encoder, a transformer-based motion embedding module, and a joint appearance and motion decoder. After extracting the hierarchical pyramid features, the transformer is used for motion embedding to inject motion cues, followed by a coarse-to-fine decoder that gradually refines the GS appearance and motion fields at the same time (*cf*. the red line), until synthesizing the final full-resolution GS image. A hidden state h^j is also passed sequentially.

Specifically, in each pyramid level j, $1 \leq j \leq L-1$, we first upsample the bilateral undistortion field $\tilde{U}^{j+1}_{g\to t}$ and the hidden state $\tilde{h}^{j+1}$ of the previous level by a deconvolution layer, resulting in $U^j_{g\to t}$ and h^j. Here, $t \in \{0,1\}$. At the same time, the previously synthesized GS candidate $\tilde{G}^{j+1}$ is also bilinearly upsampled to yield G^j. Inspired by [5, 12, 29], we update $U^j_{g\to t}$ by multiplying the time offset to better explore the scanline-dependent properties of RS images. Then, based on the RS-aware motion fields $U^j_{g\to t}$ and the encoded pyramidal features F^j_t, a warping operation $\mathcal{W}$ is applied to generate two warping-based GS candidates $\hat{F}^j_t = \mathcal{W}(F^j_t, U^j_{g\to t})$ for intermediate feature reconstruction at current level. Note that at the bottom level, we warp the original RS image instead of the feature map to produce $\hat{G}^1_t$ for efficiency. Subsequently, the warping-based and synthesis-based GS appearances can work together to decode complementary information for efficient RSC.

Similar to [5, 10, 29, 56], the warping-based branch helps to compensate for RS edge distortions and place image patches in the correct position. Synchronously, in the synthesis-based branch, the latent GS appearance is capable of enhancing the quality of the bilateral motion fields. Moreover, we maintain a hidden state h^j to promote more adequate information transfer. Immediately after, $\hat{F}^j_t$, $U^j_{g\to 0}$, $U^j_{g\to 1}$, G^j, and h^j are cascaded and fed into three residual blocks, and its output is decoded by a simple 3×3 convolution layer to predict both the bilateral undistortion field $\tilde{U}^j_{g\to t}$ and the synthesized GS candidate $\tilde{G}^j$. Notably, the residual connection from $U^j_{g\to t}$ is used to update $\tilde{U}^j_{g\to t}$. Note that, as the initialization of coarse-to-fine refinement, we perform feature aggregation on $\bar{F}^L_0$ and $\bar{F}^L_1$ directly at the top level. More importantly, the GS candidate $\tilde{G}^1$ synthesized at the bottom level is our final desired GS image. Overall, the decoder process can be formulated as follows:

$$\begin{aligned}
[U^{L-1}_{g\to t}, G^{L-1}, h^{L-1}] &= Up(\mathcal{D}^L([\bar{F}^L_0, \bar{F}^L_1])), \\
[U^{j-1}_{g\to t}, G^{j-1}, h^{j-1}] &= Up(\mathcal{D}^j([\hat{F}^j_0, \hat{F}^j_1, U^j_{g\to t}, G^j, h^j])), \\
\tilde{G}^1 &= \mathcal{D}^1([\hat{G}^1_0, \hat{G}^1_1, U^1_{g\to t}, G^1, h^1]),
\end{aligned} \tag{2}$$

where $\mathcal{D}^j$ $(j = 2, ..., L-1)$ are middle-level decoders, Up denotes an upsampling operation, $[\cdot]$ indicates a concatenation operation, t refers to both 0 and 1 due to spatial limits. Note that we do not show all intermediate variables in Eq. 2 and Fig. 3 for clarity; see *suppl. materials* for more details.

3.4. Loss function

Our model can be end-to-end trained. Given a pair of consecutive RS images R_0 and R_1, our JAMNet jointly estimates multi-scale synthesis-based GS candidates $\hat{G}^i$ $(1 \leq i \leq L)$, and bilateral undistortion fields $\hat{U}^j_{g\to 0}$, $\hat{U}^j_{g\to 1}$ $(1 \leq j \leq L-1)$ from coarse to fine. Here, $\hat{G}^1$ is the final full-resolution GS image we desire to obtain. We use G^i to denote the corresponding ground-truth (GT) GS image.

Note that the superscript i indicates $1/2^{i-1}$ resolution maps at level i, and G^1 represents the GS GT with the same resolution as the input RS image. Our total loss function is a linear combination of four terms:

$$\mathcal{L} = \lambda_{\text{r}}\mathcal{L}_{\text{r}} + \mathcal{L}_{\text{p}} + \lambda_{\text{mc}}\mathcal{L}_{\text{mc}} + \lambda_{\text{tv}}\mathcal{L}_{\text{tv}}, \tag{3}$$

where λ_{r}, λ_{mc} and λ_{tv} are trade-off hyper-parameters. The pixel intensities of images lie in the range $[0, 1]$.

Reconstruction loss $\mathcal{L}_{\text{r}}$. We measure the pixel-wise reconstruction quality of the final synthesized GS image at the bottom level as:

$$\mathcal{L}_{\text{r}} = \rho(G^1 - \hat{G}^1), \tag{4}$$

where $\rho(x) = \sqrt{x^2 + \varepsilon^2}$ is the Charbonnier penalty function [6]. We set the constant ε to 0.001.

Perceptual loss $\mathcal{L}_{\text{p}}$. Since using only the $\mathcal{L}_{\text{r}}$ loss may cause blur in the final frame prediction [10, 22, 29], we additionally employ a widely used $\mathcal{L}_{\text{p}}$ loss [24] to preserve fine details and improve the perceptual quality of final synthesized GS images. Specifically, we define the perceptual loss $\mathcal{L}_{\text{p}}$ as:

$$\mathcal{L}_{\text{p}} = \|\phi(G^1) - \phi(\hat{G}^1)\|_1, \tag{5}$$

where ϕ indicates the conv4_3 feature extractor of the pre-trained VGG16 network [43].

Multi-scale consistency loss $\mathcal{L}_{\text{mc}}$. To better guide the final GS frame synthesis, we force the GS candidates (including warping-based and synthesis-based ones) to be consistent with the GT across different pyramid levels. Specifically, from level 1 to $L-1$, we combine the bilateral undistortion fields and the warping operator $\mathcal{W}$ to obtain the warping-based GS candidates $\hat{G}_t^j = \mathcal{W}(R_t, \hat{U}_{g\to t}^j)$, where $t \in \{0, 1\}$, $1 \le j \le L-1$. Meanwhile, from level 2 to L, our JAMNet produces the multi-scale synthesis-based GS candidates $\hat{G}^i$, where $2 \le i \le L$. Formally, the $\mathcal{L}_{\text{mc}}$ loss consists of a warping loss $\mathcal{L}_{\text{warp}}$ and a synthetic loss $\mathcal{L}_{\text{syn}}$, *i.e.*,

$$\begin{aligned}
&\mathcal{L}_{\text{mc}} = \mathcal{L}_{\text{warp}} + \mathcal{L}_{\text{syn}},\\
&\mathcal{L}_{\text{syn}} = \tfrac{1}{L-1}\textstyle\sum_{i=2}^{L} \alpha_i \cdot \rho(G^i - \hat{G}^i),\\
&\mathcal{L}_{\text{warp}} = \tfrac{1}{2(L-1)}\textstyle\sum_{t=0}^{1}\sum_{j=1}^{L-1} \alpha_j \beta \cdot \rho(G_t^j - \hat{G}_t^j),
\end{aligned} \tag{6}$$

where α. and β depict the importance at multiple scales.

Total variation loss $\mathcal{L}_{tv}$. Finally, to enforce the estimated flow to be smooth [29, 31, 55], we add a smooth regularization on the bilateral undistortion fields as:

$$\mathcal{L}_{\text{tv}} = \frac{1}{2(L-1)} \sum_{t=0}^{1} \sum_{j=1}^{L-1} \|\hat{U}_{g\to t}^j\|_2. \tag{7}$$

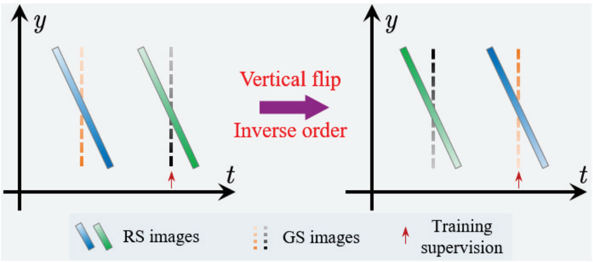


Figure 4. **Illustration of our proposed data augmentation strategy.** Gradient colors are used for ease of understanding.

3.5. A new data augmentation strategy

Given two consecutive RS images, we predict the GS image corresponding to the middle scanline of the second RS frame. Nevertheless, the prevailing RSC datasets [5, 29] are often constructed based on the RS video, *i.e.*, the GT GS image corresponding to the middle scanline of the first RS frame is available at the same time. To maximize the utilization of the current RSC dataset, we propose a novel data augmentation strategy: *vertical flip + inverse order*. In this way, the last scanline of the original second RS frame will become the first scanline of the new first RS frame; the first scanline of the original first RS frame will become the last scanline of the new second RS frame, as shown in Fig. 4. This can effectively improve the performance of the resulting RSC model, as demonstrated in Section 4.3.2.

4. Experiments

4.1. Dataset and implementation details

Datasets. We adopt the standard RSC benchmark datasets [29] including Carla-RS and Fastec-RS. The Carla-RS dataset is synthesized from a virtual 3D environment, involving general six degrees of freedom camera motions. For the Fastec-RS dataset, RS images are generated by row-by-row stitching of high-frame-rate GS videos captured by a high-speed GS camera mounted on a ground vehicle. Note that the Carla-RS dataset provides the GT occlusion mask. Following [10, 29], we perform quantitative evaluations as follows: the Carla-RS dataset with occlusion mask (*CRM*), the Carla-RS dataset without occlusion mask (*CR*), and the Fastec-RS dataset (*FR*). Moreover, we utilize the recently released real-world BS-RSC dataset [5], in which RS-GS image pairs are acquired by a well-designed beam-splitter acquisition system in the dynamic urban environment. In particular, various camera and object motions (*e.g.*, vehicles and pedestrians) are covered in the BS-RSC dataset.

Training details. Our JAMNet is trained end-to-end using the Adam optimizer [25] for 600 epochs with a learning rate of 10^{-4} and a batch size of 8. Similar to [10], we construct a 5-level pyramid, *i.e.*, $L = 5$. The number of feature channels is $\{16, 28, 40, 64, 96\}$ to balance accuracy and

Table 1. Quantitative comparison against the state-of-the-art RSC methods on the Carla-RS and Fastec-RS datasets [29]. The numbers in **red** and blue represent the best and second-best performance. Our approach consistently achieves the highest RSC accuracy and fastest inference time, while maintaining a compact and lightweight network structure. Note that our method is capable of real-time RSC for the first time, thanks to the design concept of learning both appearance and motion in a single-stage architecture.

Method	# Parameters	Runtime	PSNR↑ (dB)			SSIM↑		LPIPS↓	
	(Million)	(ms)	CRM	CR	FR	CR	FR	CR	FR
SUNet [10] + BMBC [35]	23.0	938	28.51	28.69	25.49	0.848	0.796	0.1033	0.2118
SUNet [10] + DAIN [4]	36.0	227	28.63	28.93	27.12	0.851	0.823	0.0919	0.1642
DiffSfM [58]	-	$4.7e^5$	25.93	22.88	21.44	0.770	0.710	0.1201	0.2180
AdaRSC [5]	4.25	302	-	-	28.56	-	0.855	-	0.0796
JCD [56]	7.51	225	28.12	27.75	26.48	0.836	0.821	0.0595	0.0943
SUNet [10]	12.0	92	28.44	28.17	27.06	0.838	0.825	0.0702	0.1030
DeepUnrollNet [29]	3.91	131	27.86	27.54	26.73	0.829	0.819	0.0555	0.0995
JAMNet (Ours)	4.73	**28**	**31.00**	**30.70**	**28.70**	**0.905**	**0.865**	**0.0371**	**0.0691**

Table 2. Quantitative comparison against the state-of-the-art RSC methods on the BS-RSC dataset [5]. Our JAMNet is far superior to baseline methods, demonstrating the significant advantages of our approach in real-world RSC applications.

RSC Method	PSNR↑ (dB)	SSIM↑
DiffSfM [58]	19.80	0.698
DeepUnrollNet [29]	25.21	0.833
SUNet [10]	27.76	0.875
JCD [56]	25.59	0.841
AdaRSC [5]	28.23	0.882
JAMNet (Ours)	**32.93**	**0.941**

efficiency. We set the hyper-parameters $\{\lambda_{\rm r}, \lambda_{\rm mc}, \lambda_{\rm tv}\}$ as $\{100, 100, 0.1\}$. Inspired by the training process of multi-scale networks (*e.g.* [8, 45]), the weights of the multi-scale consistency loss in Eq. (6) are empirically set to $\alpha_5 = \alpha_4 = \alpha_3 = 0.25$, $\alpha_2 = \alpha_1 = 0.5$, and $\beta = 0.5$. We uniformly transfer hidden states with 16 channels between the pyramid levels. The GT GS images are downsampled to yield multi-scale supervision signals. Following [9, 10, 14], we keep the vertical resolution constant and leverage a uniform random crop with a horizontal resolution of 256 pixels during training. Meanwhile, we also augment the training data with random "horizontal flips" and our newly proposed random "vertical flip + reverse order" strategies (*cf*., Section 3.5). All experiments were implemented with PyTorch and executed on a single NVIDIA RTX 3090 GPU.

Evaluation metrics. Following previous works, we apply standard PSNR and SSIM metrics, and learned perceptual metric LPIPS [54] to compute the quantitative result.

4.2. Comparison with SOTA methods

We perform comparisons with the following baselines. (i) **DiffSfM** [58] is a traditional two-image based RSC method that needs sophisticated differential RS optimization. (ii) **DeepUnrollNet** [29] and **SUNet** [10] develop specialized CNNs to remove RS artifacts from two consecutive RS frames. (iii) **JCD** [56] and **AdaRSC** [5] are the deep learning solutions of three-image based RSC, recovering a GS image of the intermediate moment. (iv) **Cascaded method** generates two first-scanline GS images sequentially from three adjacent RS inputs using SUNet, and then interpolates an in-between GS image using BMBC [35] or DAIN [4], called "SUNet+BMBC" or "SUNet+DAIN".

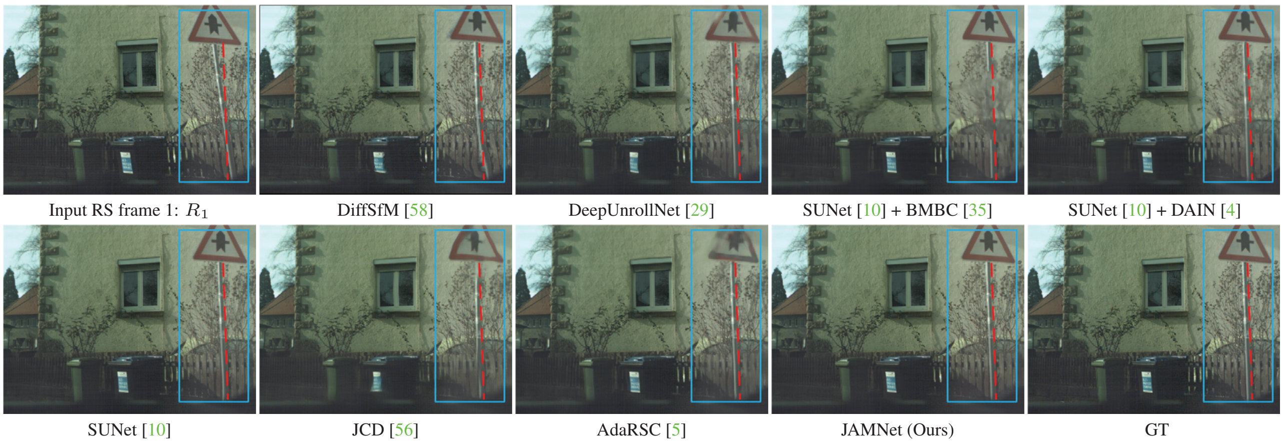
The quantitative results are presented in Tables 1 and 2. It can be seen that our approach consistently achieves excellent RSC performance, outperforming state-of-the-art RSC baselines by a large margin. The two-stage RSC methods [5, 10, 29, 56] have a bottleneck in efficiently removing RS effects, which inevitably consumes a lot of inference time. It is worth mentioning that our method can process consecutive RS images and output a high-fidelity GS frame in real-time, while maintaining a compact and lightweight network design. Note that, to the best of our knowledge, we are the first to implement real-time RSC (also see Fig. 1). In addition, our approach surpasses the SOTA RSC method in PSNR by **4.7 dB** in real-world RS effect removal (*cf*. Table 2). These achievements are of great significance for the practical application of the RSC method. Overall, these experiments validate the superiority of our single-stage RSC architecture that explores complementary appearance and motion information simultaneously.

We illustrate the qualitative results under the Fastec-RS dataset and the BS-RSC dataset in Figs. 5 and 6, respectively. Constrained by the specific RS model, the traditional RSC methods [58] are prone to ghosting artifacts, and are time-consuming due to complex non-linear optimization. Cascaded methods tend to be blurry and locally inaccurate due to error accumulation, and also suffer from the disadvantage of larger and more time-consuming models. Thanks to the joint learning of appearance and motion, our method can automatically reason about occlusions and complex motion contours to efficiently recover crisp and pleasing GS images. As a result, more GS details with fewer artifacts are restored by our method successfully.

4.3. Ablation studies

4.3.1 Ablation on network architecture

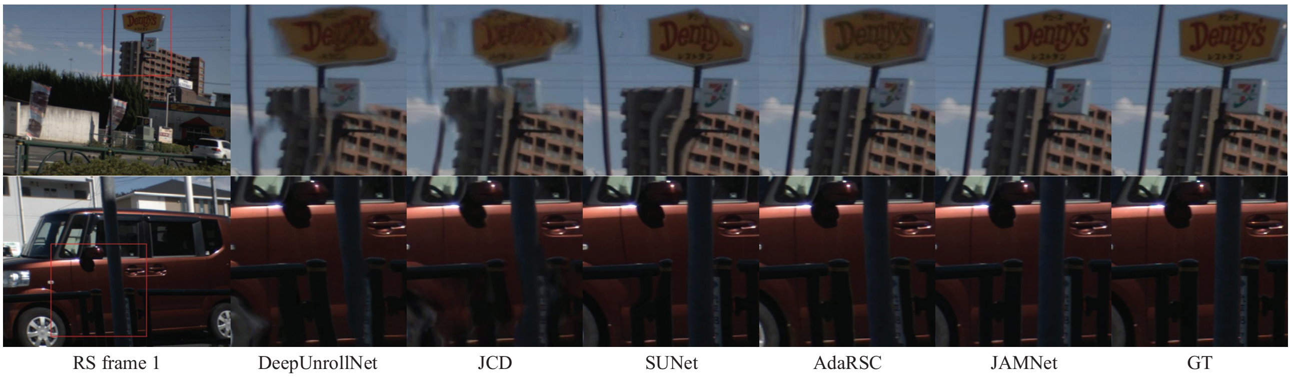
Ablation on model capacity. Based on the base model parameters of JAMNet, we apply width multipliers [21] to the



Input RS frame 1: R_1 | DiffSfM [58] | DeepUnrollNet [29] | SUNet [10] + BMBC [35] | SUNet [10] + DAIN [4]

SUNet [10] | JCD [56] | AdaRSC [5] | JAMNet (Ours) | GT

Figure 5. Qualitative results against baseline methods on the Fastec-RS dataset. Our method competently recovers higher visual quality GS images with more details. The cascade method is prone to blurring or local ghosting due to compounding errors. Zoom in for best view.



RS frame 1 | DeepUnrollNet | JCD | SUNet | AdaRSC | JAMNet | GT

Figure 6. Visual comparison on the BS-RSC dataset for real-world RSC. Our method can successfully remove RS artifacts and generate higher-fidelity GS images. With our JAMNet, clearer and more accurate GS image appearance is restored effectively and efficiently.

number of channels uniformly at each feature pyramid extraction layer. For example, choosing a width multiplier of 0.5 will produce an RSC model with feature channels of $\{8, 14, 20, 32, 48\}$. The performance of these models with different complexity is presented in Table 3a. It can be seen that our model design is flexible and tractable, and increasing the model capacity can effectively ameliorate the model performance. For the balance of efficiency and accuracy, our JAMNet adopts the base setting of $\{16, 28, 40, 64, 96\}$.

Ablation on joint learning mechanism. As shown in Table 3b, we first remove the network branches associated with synthesis-based and warping-based GS candidates, respectively. Then, similar to [29, 56], we solely warp the feature of the second RS frame, denoted as “No context”. One can observe that unified appearance synthesis and motion warping can better explore the underlying complementary information, which facilitates contextual aggregation and occlusion reasoning for better RS effect compensation.

Ablation on pyramid level. To verify the impact of the pyramids, we construct feature pyramids with different levels, as shown in Table 3d. Using 3-level pyramids leads to consistently worse results. Note that DeepUnrollNet exploits a 3-level pyramid involving a two-stage architecture, our method however achieves better RSC performance (27.28 dB *vs.* 26.73 dB) with smaller model capacity (0.76M *vs.* 3.91M), which proves the advantage of our single-stage design. Note also that, since the computation of the transformer is proportional to the square of the image resolution, the 3-level has more calculations than the 5-level and is therefore more time-consuming. With the increase of pyramid levels, the RSC results are significantly improved. We reckon this is because more pyramids are beneficial for the perception and rectification of large pixel displacements.

Ablation on motion embedding module. As reported in Table 3e, removing this module leads to obvious performance degradation. Compared with building cost volume at the top level, the transformer shows a more powerful motion modeling capability such that richer motion information can be injected into subsequent decoders, thereby enhancing the corrected image quality effectively.

Ablation on hidden state. Maintaining a hidden state between adjacent pyramid levels helps the network to transfer information better, as shown in Table 3f. Moreover, increasing its capacity is beneficial to improve the RSC per-

Table 3. **Ablation studies on the Fastec-RS dataset**. Our full model is indicated in the leftmost column with underlining.

(a) Larger-capacity **feature pyramid extractor** has better performance. "$\times N$" denotes a width multiplier of N for the number of channels.

	PSNR	SSIM	#Paras	Time
×0.5	27.98	0.846	1.24	23
×1.0	28.70	0.865	4.73	28
×1.5	28.75	0.867	10.5	35
×2.0	**28.94**	**0.870**	18.5	41

(b) **Joint learning mechanism.** Unified construction of synthesis-based and warping-based GS candidate branches facilitates occlusion inference and motion compensation.

	PSNR	SSIM	#Paras	Time
Full model	**28.70**	**0.865**	4.73	28
No synthesis	28.34	0.860	4.71	27
No warping	26.10	0.787	4.71	20
No context	27.35	0.840	4.72	26

(c) **Loss function.** All four loss terms have positive effects. $\mathcal{L}_{\mathrm{r}}$ and $\mathcal{L}_{\mathrm{mc}}$ losses are crucial to train an effective model.

	PSNR	SSIM
No $\mathcal{L}_{\mathrm{r}}$	26.36	0.691
No $\mathcal{L}_{\mathrm{p}}$	28.45	0.862
No $\mathcal{L}_{\mathrm{mc}}$	27.84	0.849
No $\mathcal{L}_{\mathrm{tv}}$	28.57	0.863

(d) More **feature pyramid levels** provide better perception and rectification of large pixel displacements, thus resulting in significantly better RSC performance.

	PSNR	SSIM	#Paras	Time
3-level	27.28	0.834	0.76	79
4-level	27.95	0.850	2.00	25
5-level	**28.70**	**0.865**	4.73	28

(e) **Motion embedding module.** Removing the motion embedding module (Nothing) leads to lower RSC results. Transformer has better motion modeling capability than cost volume.

	PSNR	SSIM	#Paras	Time
Nothing	28.02	0.853	4.55	25
Cost volume	28.31	0.860	4.68	26
Transformer	**28.70**	**0.865**	4.73	28

(f) **Hidden state.** Removing the hidden state (0) causes moderate performance loss. A larger number of hidden state channels is beneficial to deliver more information.

	PSNR	SSIM	#Paras	Time
0	28.01	0.852	4.58	23
8	28.39	0.860	4.65	27
16	**28.70**	**0.865**	4.73	28

Table 4. **Ablations on our proposed data augmentation** (DA) using the Fastec-RS and BS-RSC datasets. Our DA effectively enhances both synthesized and real-world RSC capabilities.

Methods	Fastec-RS		BS-RSC		DA
	PSNR	SSIM	PSNR	SSIM	
DeepUnrollNet	26.73	0.819	25.21	0.833	✗
	27.01	**0.828**	**26.13**	**0.850**	✓
JAMNet (Ours)	28.33	0.859	32.56	0.938	✗
	28.70	**0.865**	**32.93**	**0.941**	✓

formance, which also reflects the flexibility of our model.

Ablation on other network details. We additionally conduct two ablations to our network design. First, we warp the bottom-level feature map instead of the RS image. Experiments show that decoding in high-resolution feature space brings a small gain, but increases the inference time (31ms *vs*. 28ms). Furthermore, we remove the time offset in estimating the undistortion field, which does not favor the network to explore the scanline-dependent properties of the RS data [5, 9, 29, 56], resulting in a 0.55 dB reduction in PSNR.

4.3.2 Ablation on training strategy

To further understand the effectiveness of our newly proposed data augmentation strategy in Section 3.5, we retrain DeepUnrollNet [29] and our JAMNet with and without the augmentation. As manifested in Table 4, applying our data augmentation can significantly improve the performance of the RSC method, which stems from a deeper exploration of the dataset. As a result, it can act as an effective tool for RSC tasks. Besides, our loss function $\mathcal{L}$ is effective because it performs best when all loss terms are used (*cf*. Table 3c).

4.4. Generalization evaluation

We apply our approach to real RS images provided by [58] and [16] to evaluate the generalization ability. These real RS datas are captured by fast-moving hand-held cameras in outdoor scenes, which are significantly different from the training dataset. Also, they are widely used for usability evaluation of RSC methods, *e.g.*, [10, 27, 39, 40, 59]. As illustrated in Fig. 7, our method can effectively and robustly remove the noticeable RS artifacts and produce geometrically and visually consistent GS images, which verifies the excellent generalization ability of our method.

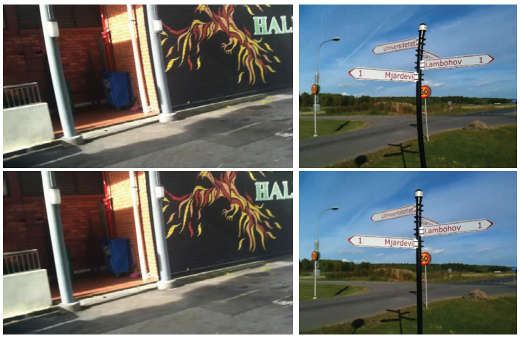


Figure 7. Generalization results on real RS data. The top is the second RS image and the bottom is our recovered GS image.

5. Conclusion

In this paper, we have proposed an efficient and flexible deep architecture, termed JAMNet, for rolling shutter correction. Unlike common two-stage RSC methods, JAMNet gradually refines GS appearance features together with bilateral motion fields in a single-stage framework, enabling much simpler yet more efficient coarse-to-fine GS recovery. Moreover, we have developed a new data augmentation strategy to unlock the potential of the RSC dataset. Experiments on various benchmarks demonstrate our method significantly outperforms prior arts in terms of speed and accuracy. It is hoped that our network design concept can shed light for future research on the RSSR task.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (62271410, 62001394, 61901387). The authors thank the anonymous reviewers and the area chairs 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2505–2513, 2020. 1

[2] Fang Bai, Agniva Sengupta, and Adrien Bartoli. Scanline homographies for rolling-shutter plane absolute pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8993–9002, 2022. 2

[3] Simon Baker, Eric Bennett, Sing Bing Kang, and Richard Szeliski. Removing rolling shutter wobble. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399, 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3703–3712, 2019. 1, 3, 6, 7

[5] Mingdeng Cao, Zhihang Zhong, Jiahao Wang, Yinqiang Zheng, and Yujiu Yang. Learning adaptive warping for real-world rolling shutter correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17785–17793, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[6] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 168–172. IEEE, 1994. 5

[7] Yuchao Dai, Hongdong Li, and Laurent Kneip. Rolling shutter camera relative pose: generalized epipolar geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4132–4140, 2016. 2

[8] 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 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 3, 6

[9] Bin Fan and Yuchao Dai. Inverting a rolling shutter camera: bring rolling shutter images to high framerate global shutter video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4228–4237, 2021. 1, 3, 6, 8

[10] Bin Fan, Yuchao Dai, and Mingyi He. SUNet: symmetric undistortion network for rolling shutter correction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4541–4550, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[11] Bin Fan, Yuchao Dai, and Mingyi He. Rolling shutter camera: modeling, optimization and learning. *Machine Intelligence Research*, 2023. 1

[12] Bin Fan, Yuchao Dai, and Hongdong Li. Rolling shutter inversion: bring rolling shutter images to high framerate global shutter video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3, 4

[13] Bin Fan, Yuchao Dai, and Ke Wang. Rolling-shutter-stereo-aware motion estimation and image correction. *Computer Vision and Image Understanding*, 213:103296, 2021. 1

[14] Bin Fan, Yuchao Dai, Zhiyuan Zhang, Qi Liu, and Mingyi He. Context-aware video reconstruction for rolling shutter cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17572–17582, 2022. 1, 3, 6

[15] Bin Fan, Yuchao Dai, Zhiyuan Zhang, and Ke Wang. Differential sfm and image correction for a rolling shutter stereo rig. *Image and Vision Computing*, 124:104492, 2022. 1

[16] Per-Erik Forssén and Erik Ringaby. Rectifying rolling shutter video from hand-held devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 507–514, 2010. 2, 8

[17] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *Proceedings of IEEE International Conference on Computational Photography*, pages 1–8, 2012. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 3

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

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

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 6

[22] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018. 3, 5

[23] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: correspondence transformer for matching across images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6207–6217, 2021. 3

[24] 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 (ECCV)*, pages 694–711, 2016. 5

[25] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, 2015. 5

[26] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4795–4803, 2018. 1, 2

[27] Yizhen Lao and Omar Ait-Aider. Rolling shutter homography and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2780–2793, 2021. 1, 2, 8

[28] Chia-Kai Liang, Li-Wen Chang, and Homer H Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330, 2008. 2

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

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 3

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

[32] Maxime Meilland, Tom Drummond, and Andrew I Comport. A unified rolling shutter and motion blur model for 3d visual registration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2016–2023, 2013. 2

[33] Marci Meingast, Christopher Geyer, and Shankar Sastry. Geometric models of rolling-shutter cameras. *arXiv preprint arXiv:cs/0503076*, 2005. 2

[34] Eyal Naor, Itai Antebi, Shai Bagon, and Michal Irani. Combining internal and external constraints for unrolling shutter in videos. *arXiv preprint arXiv:2207.11725*, 2022. 2

[35] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. BMBC: bilateral motion estimation with bilateral cost volume for video interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 109–125, 2020. 3, 6, 7

[36] Abhijith Punnappurath, Vijay Rengarajan, and AN Rajagopalan. Rolling shutter super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 558–566, 2015. 2

[37] 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 (WACV)*, pages 903–911, 2018. 2

[38] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in Manhattan world. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 882–890, 2017. 1, 2

[39] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: cnn to correct motion distortions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2291–2299, 2017. 2, 8

[40] Vijay Rengarajan, Ambasamudram N Rajagopalan, and Rangarajan Aravind. From bows to arrows: rolling shutter rectification of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2773–2781, 2016. 1, 2, 8

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

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

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

[44] Shuochen Su and Wolfgang Heidrich. Rolling shutter motion deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1529–1537, 2015. 2

[45] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. 3, 6

[46] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: detector-free local feature matching with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8922–8931, 2021. 3

[47] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419, 2020. 3

[48] Subeesh Vasu, Mahesh MR Mohan, and AN Rajagopalan. Occlusion-aware rolling shutter rectification of 3d scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–645, 2018. 1

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. 3

[50] Ke Wang, Bin Fan, and Yuchao Dai. Relative pose estimation for stereo rolling shutter cameras. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 463–467, 2020. 1

[51] Zhixiang Wang, Xiang Ji, Jia-Bin Huang, Shin'ichi Satoh, Xiao Zhou, and Yinqiang Zheng. Neural global shutter: learn to restore video from a rolling shutter camera with global reset feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17794–17803, 2022. 1

[52] Huicong Wu, Liang Xiao, and Zhihui Wei. Simultaneous video stabilization and rolling shutter removal. *IEEE Transactions on Image Processing*, 30:4637–4652, 2021. 2

[53] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: learning optical flow via global matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8121–8130, 2022. 3

[54] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 6

[55] Zhihang Zhong, Mingdeng Cao, Xiao Sun, Zhirong Wu, Zhongyi Zhou, Yinqiang Zheng, Stephen Lin, and Imari Sato. Bringing rolling shutter images alive with dual reversed distortion. *arXiv preprint arXiv:2203.06451*, 2022. 5

[56] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9219–9228, 2021. 1, 2, 3, 4, 6, 7, 8

[57] Xinyu Zhou, Peiqi Duan, Yi Ma, and Boxin Shi. EvUnroll: neuromorphic events based rolling shutter image correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17775–17784, 2022. 2

[58] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 948–956, 2017. 1, 2, 6, 7, 8

[59] Bingbing Zhuang and Quoc-Huy Tran. Image stitching and rectification for hand-held cameras. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 243–260, 2020. 2, 8

[60] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4551–4560, 2019. 1, 2