Supplementary Materials: Towards Nonlinear-Motion-Aware and Occlusion-Robust Rolling Shutter Correction

A. Overview

The references are quoted in blue to the main text.

- Appendix B: Additional experiments details.
- Appendix C: Additional ablation study.
- Appendix D: Additional performance comparison.
- Appendix E: Motion estimation accuracy.
- Appendix F: The order of Taylor expansion.

B. Additional experiments details

Details of ACC dataset. To validate the ability to handle the complex nonlinear motion of the proposed method on a more challenging dataset, we derived the ACC dataset by excluding frames under constant general motions (even static) from the BS-RSC [1] dataset. The Tab. 1 and Fig. 1 show the removed frame from BS-RSC[1].

Table 1: The removed frames of BS-RSC [1].

Scene ID	Removed frame
Scene 53	00000 - 00004
Scene 56	00000 - 00024
Scene 62	00000 - 00009

C. Additional ablation study

Ablation study of the optical flow. In our implementation, we use the RAFT [12] (for Fastec-RS and BS-RSC datasets) or GMA [7] (for Carla-RS dataset) of OpenMMLab Optical Flow Toolbox [2] to predict optical flow from the 5 consecutive RS frames, followed by Multi-QRS motion solver to predict the correction fields and obtain the latent occluded three RSC frames. To investigate the effect of optical flow for the proposed method, we replace RAFT with a weaker optical flow estimator PWCNet [11]. As shown in Table. 2, the performance of using the PWC optical flow estimator degrades, but it still significantly outperforms the state-of-the-art.

Figure 1: Illustration of removed static and constant frames of BS-RSC[1].

Table 2: Ablation study of RAFT [12] and PWCNet [11] optical flow estimator on the BS-RSC dataset.

Setting	PSNR↑	SSIM ↑	LPIPS ↓
RAFT [12]	33.50	0.946	0.0299
PWCNet [11]	31.57	0.924	0.0464

D. Additional performance comparison.

D.1. Generalization Ability

We provide the generalization capability of the proposed method on PSNR in Fig. 3 by performing cross-tests on Carla-RS, Fastec-RS and BS-RSC datasets. The confusion matrices show the relative decline rate rde in Eq. (13) for evaluation, and a lower rde indicates a better generalization. Our method outperforms DSUN and CVR, demonstrating a satisfied generalization ability. Besides, this is well supported by the visual correction result in Section. D.3.

D.2. RS temporal super resolution

Benefiting from the analytical properties of Eq. (6)(7), the proposed method is able to handle RS temporal super-resolution tasks as [4] by adjusting the hyperparameter τ. This section provides examples of RS temporal super-resolution under nonlinear motions and dynamic scenes. As shown in Fig. 2, our methods achieve rich and satisfying RS correction results among the time stamp τ from 0 to 0.8. Especially in the challenging 3rd row, where the camera was rotating rapidly to film the forwarding train, our method simultaneously removed the RS effect of the high-speed green train and pole. In constant, CVR fails to correct poles in the first and second rows and the door of the green

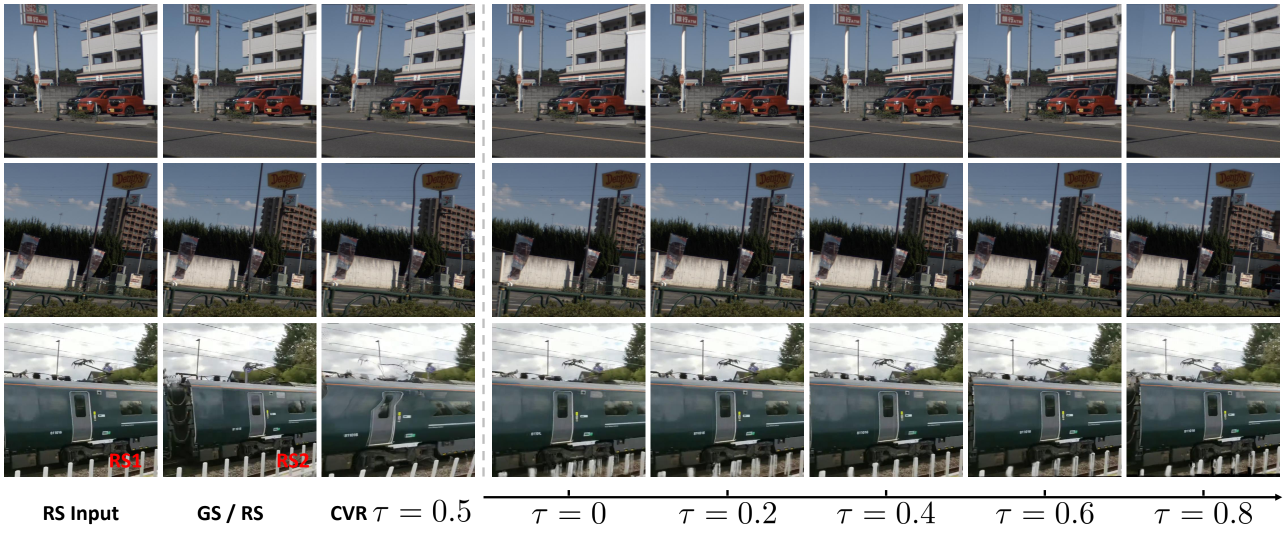

Figure 2: Example results of recovering temporal super-resolution video images from input RS images by using CVR [4] and our method.

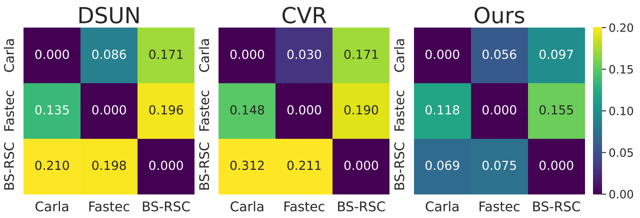

Figure 3: Generalization capability comparisons on PSNR of our method with existing RSC algorithms DSUN [9] and CVR [4] across Carla-RS, Fastec-RS, and BS-RSC datasets.

Table 3: Comparison of runtime with input RS image at varying resolutions.

Resolution	Method			
	CVR (2)	AdaRSC (3)	QRS Solver, Ours (3)	Ours (5)
640 × 480	0.175	0.2264	0.304	1.227
1024 × 768	0.271	0.4220	0.648	2.013
1920 × 1080	0.468	1.0291	1.706	6.201

Table 4: $\epsilon = \frac{\text{duration of original frames in the sequence}}{\text{the processing time of different RSC augmented ORB-SLAM2}} \uparrow$

RS	DSfM	DSUN	JCD	SUNet	CVR	AdaRSC	QRS Solver
1.90404	0.00002	0.08600	0.04680	0.20400	0.21530	0.13766	**1.17030**

train in such a dynamic scene.

D.3. Visual Comparisons

In addition to Carla-RS, Fastec-RS [9] and BS-RSC [1] datasets, We provide more visual comparison details on various datasets with general 6 DoF motion, dynamic scenes, and nonlinear movements, i.e., GPark [6], Seq77 [8], 3GS, and House [5] datasets. Besides, we capture a number of real RS videos, which contain frames in a highly dynamic scene from YouTube, namely, Train at speed and Bus. We merely display visible results because they only provided RS images but no GS images. The compared state-of-the-art RSC methods are geometry-based methods **DSfM** [14], and learning-based methods **DSUN** [9], **SUNet** [3], **JCD** [13], **CVR** [4] and **AdaRSC** [1]. All the results are shown in Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14 and Fig. 15.

D.4. Runtime Comparison

We compare our time cost with others from two aspects:

▷ **Image Correction:** Our code is implemented in python and runs on a server with an Intel Xeon Gold 6248R (96) @ 3.000GHz CPU and NVIDIA A100 GPU. The runtime result is shown in Table. 3. On average, our method uses 5 frames as input and respectively costs 1.227s, 2.013s, and 6.201s on the resolution 640 × 480, 1024 × 768 and 1920 × 1080, about 5 times of AdaRSC [1] and 7 times of CVR [4]. Besides, the Prime QRS Solver takes 3 frames as input and consumes about 1.5 times AdaRSC [1] and 2 times CVR [4]. The proposed method requires dense matching between multiple consecutive frames, which causes the high time cost. However, this is actually acceptable, considering the satisfying performance of image correction handling extremely dynamic occlusion and nonlinear motion.

▷ **3D Vision Applications:** By applying QRS Solver (W/o RSA^{2}Net) to correct sparse features only instead of every pixel as a pre-processing for the downstream 3D analysis with real-time demands, such as ORB-SLAM (in the supplemental video). Our algorithm will eliminate the high time cost, but other GPU-based methods do not have this potential. The real-time factors ϵ reported in Tab. 4 demonstrate the efficiency of QRS Solver on the TUM dataset at more than 5.7 times faster than the SOTA methods.

E. Motion estimation accuracy

As shown in Fig. 4a, to directly measure the motion estimation accuracy, we simulated a cube scene with 216 3D points (with constant and non-linear motion) because the practical correction fields are unavailable on real datasets. This experiment investigated the motion estimation errors in **Euclidean distance** (Fig. 4b) between the predicted **RS correction vector** fields from two motion models (linear and QRS) and the **Ground Truth**, as the ratio of non-linear points increased from 0 to 100. The results in Fig. 4c demonstrate that as the ratio of non-linear points grows, the error of the Linear Solver will increasingly exceed that of the QRS Solver.

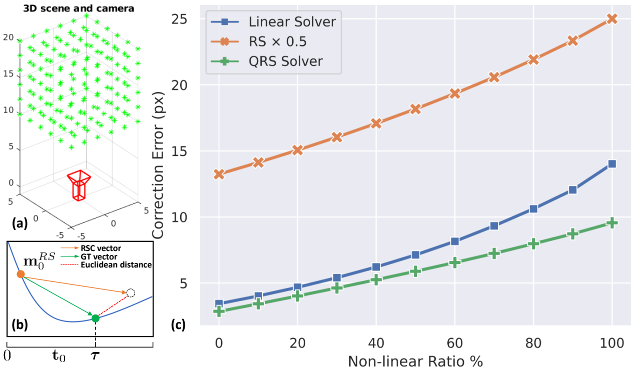

Figure 4: Illustration of motion estimation error.

F. The order of Taylor expansion

Intuitively, higher-order models can improve accuracy but certainly lead to higher computational costs and numerical instability. According to the evaluation results in Tab. 5, the 3-order model performs worse due to its sensitivity to optical flow noise and indicates lower numerical stability and robustness. Instead, the 2-order model achieves a favorable trade-off between accuracy and cost. A similar conclusion is also evidenced by [10] (Fig. 4) that the 2-order model achieves significant improvements, while the 3-order model shows only small improvements. According to our observations, the 2-order model is sufficient to handle daily applications, even on rarely challenging datasets shown in Fig. 8, Fig. 9, Fig. 10 and Fig. 11. (Note that the results are slightly inconsistent with Fig. 10 because the 3-ord model takes 4 frames as input and corrects 1 image, so we evaluated the corrected $N-3$ images on each sequence). Besides, the proposed **RSAdaCof** has addressed QRS Solver's inability to fully model all motion in Sec. 3.5 and ablation study in Tab. 5 demonstrates its remarkable effectiveness.

Table 5: evaluation and runtime (s) comparison.

model	Carla-RS (constant velocity)				BSRSC (non-linear motion)			
	PSNR ↑	SSIM ↑	LPIPS ↓	TIME	PSNR ↑	SSIM ↑	LPIPS ↓	TIME
1 ord	24.32	0.875	0.0650	0.263	26.11	0.902	0.0539	0.483
2 ord	24.33	0.878	0.0650	0.399	26.52	0.913	0.0517	0.643
3 ord	24.09	0.873	0.0663	0.553	26.38	0.911	0.0526	0.818

References

[1] 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

[2] MMFlow Contributors. MMFlow: Openmmlab optical flow toolbox and benchmark. https://github.com/open-mmlab/mmflow, 2021. 1

[3] Bin Fan, Yuchao Dai, and Mingyi He. Sunet: symmetric undistortion network for rolling shutter correction. In *ICCV*, pages 4541–4550, 2021. 2

[4] Bin Fan, Yuchao Dai, Zhiyuan Zhang, Qi Liu, and Mingyi He. Context-aware video reconstruction for rolling shutter cameras. In *CVPR*, 2022. 1, 2

[5] Per-Erik Forssén and Erik Ringaby. Rectifying rolling shutter video from hand-held devices. In *CVPR*, 2010. 2, 4, 5

[6] Chao Jia and Brian L Evans. Probabilistic 3-d motion estimation for rolling shutter video rectification from visual and inertial measurements. In *MMSP*, pages 203–208, 2012. 2, 6

[7] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, pages 9772–9781, 2021. 1

[8] J. H. Kim, C. Cadena, and I. Reid. Direct semi-dense slam for rolling shutter cameras. In *ICRA*, 2016. 2, 4

[9] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *CVPR*, pages 5941–5949, 2020. 2, 4

[10] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*, 2017. 3

[11] 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. 1

[12] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 1

[13] Zhihang Zhong, Yinqiang Zheng, and Imari Sato. Towards rolling shutter correction and deblurring in dynamic scenes. In *CVPR*, pages 9219–9228, 2021. 2

[14] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *ICCV*, 2017. 2

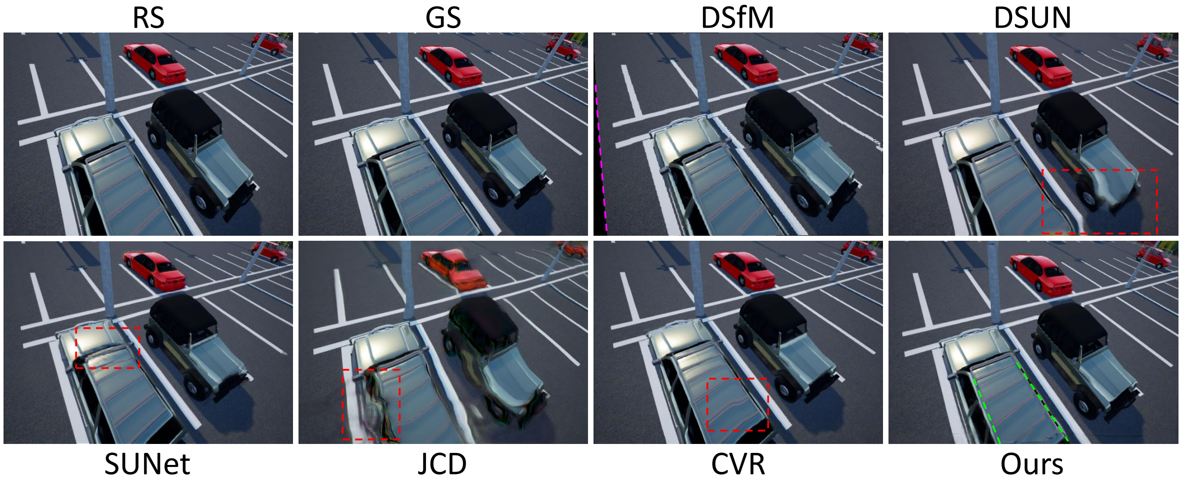

Figure 5: Visual comparison against the state-of-the-art RSC methods in general 6 Dof scenes of Carla-RS [9] dataset.

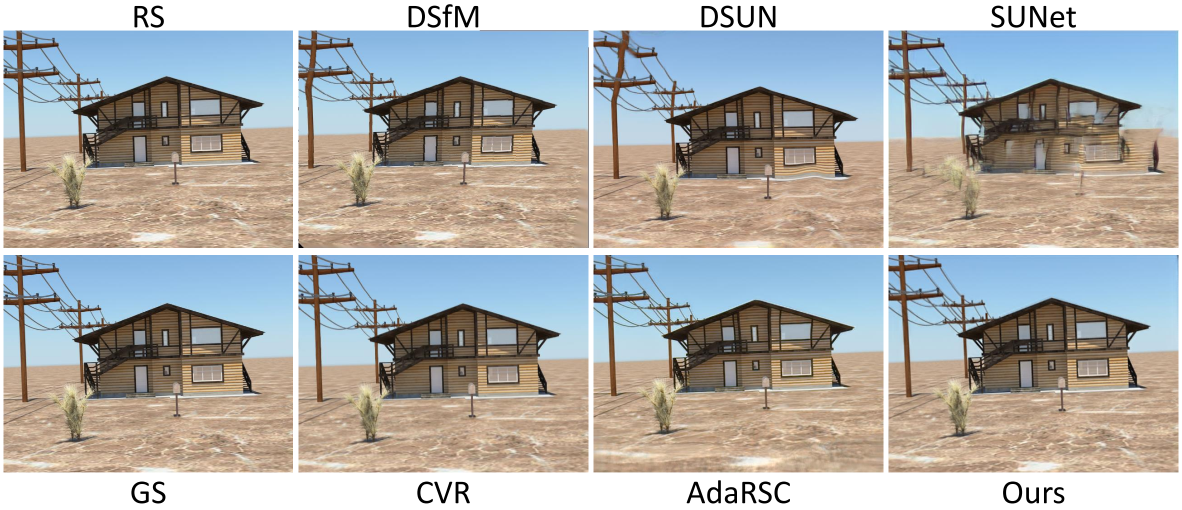

Figure 6: Visual comparison against the state-of-the-art RSC methods on House [5] datasets.

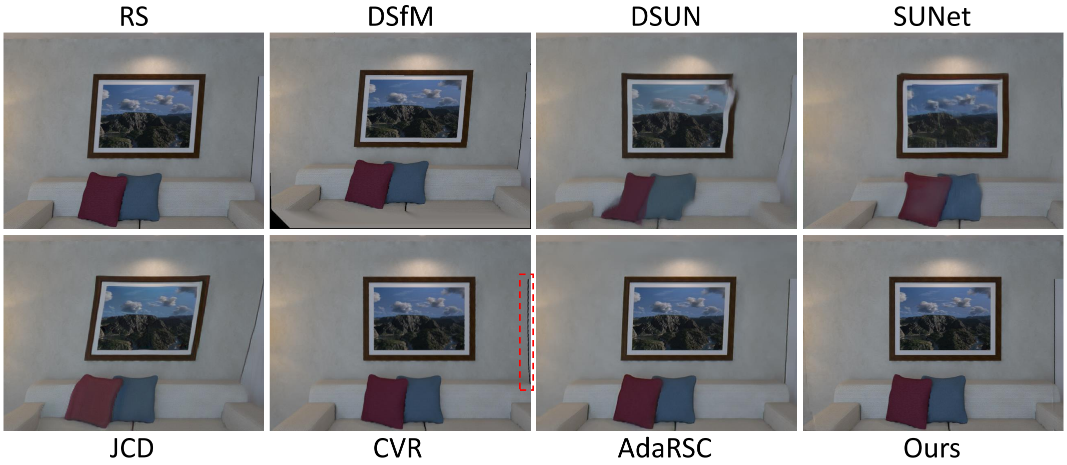

Figure 7: Visual comparison against the state-of-the-art RSC methods on Seq77 [8] dataset.

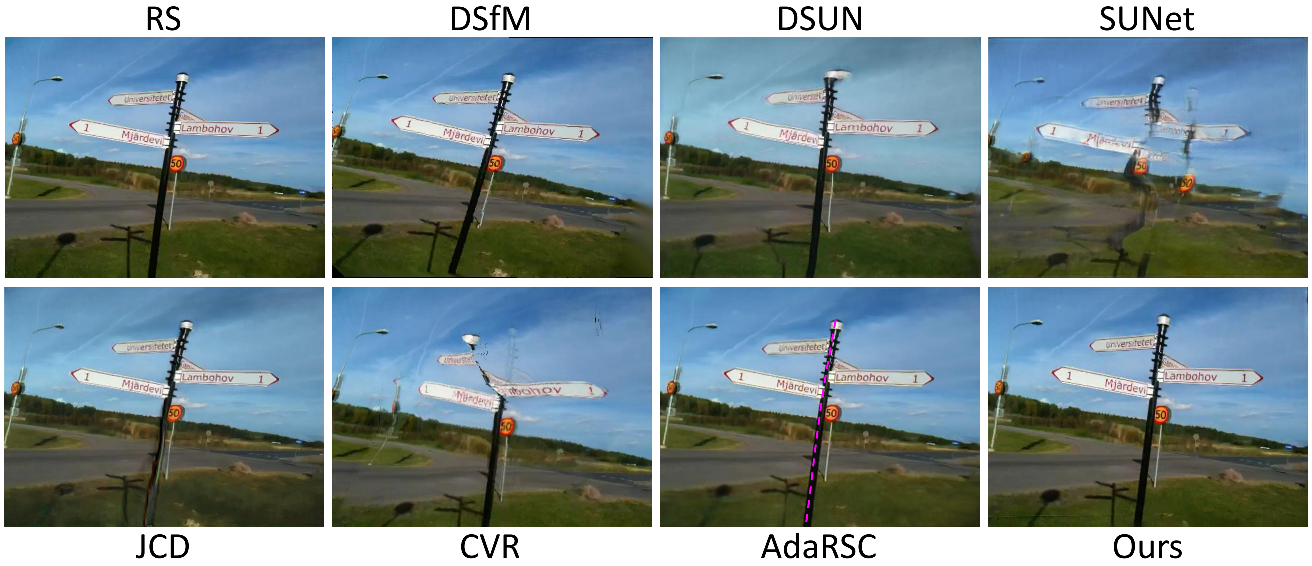

Figure 8: (a) Visual comparison against the state-of-the-art RSC methods in extremely nonlinear motion scenes of 3GS [5] dataset. Existing works fail to handle strong rotation and strong translation.

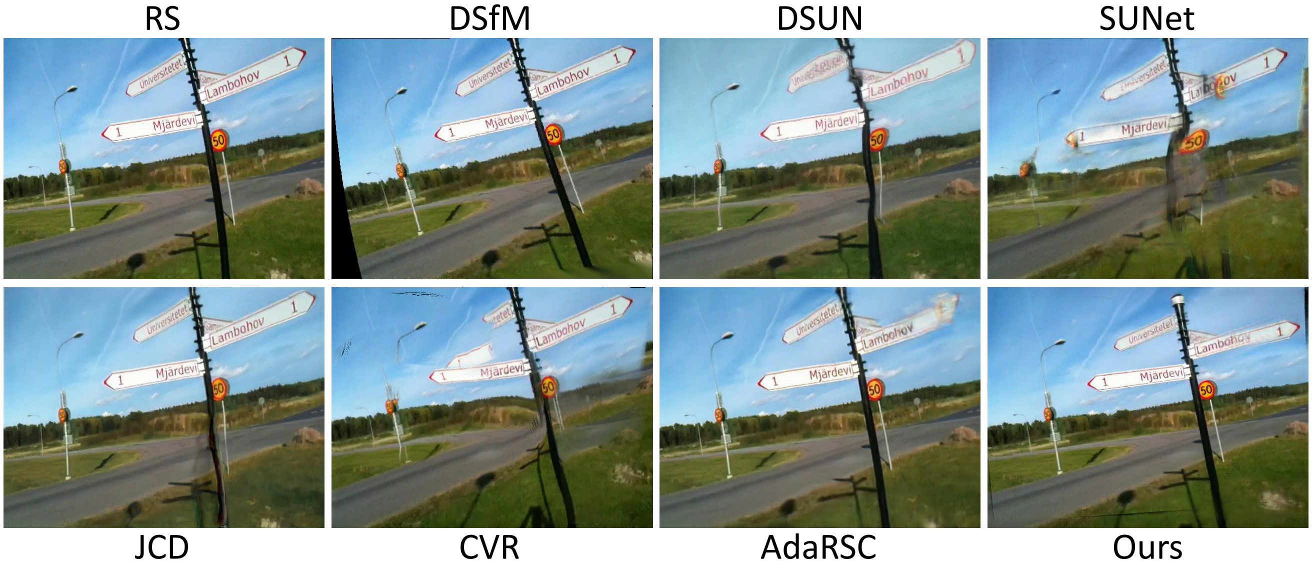

Figure 9: (b) Visual comparison against the state-of-the-art RSC methods in extremely nonlinear motion scenes of 3GS [5] dataset. Existing works fail to handle strong rotation and strong translation.

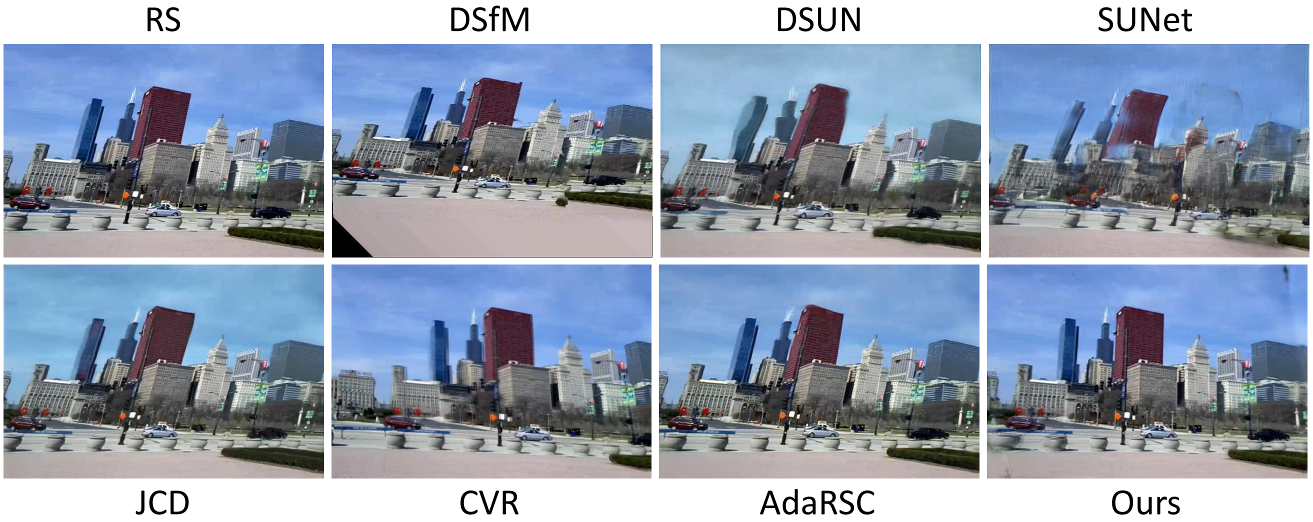

Figure 10: (a) Visual comparison against the state-of-the-art RSC methods in nonlinear motion scenes of Gpark [6] datasets. Existing works fail to handle strong translation and produce local blur.

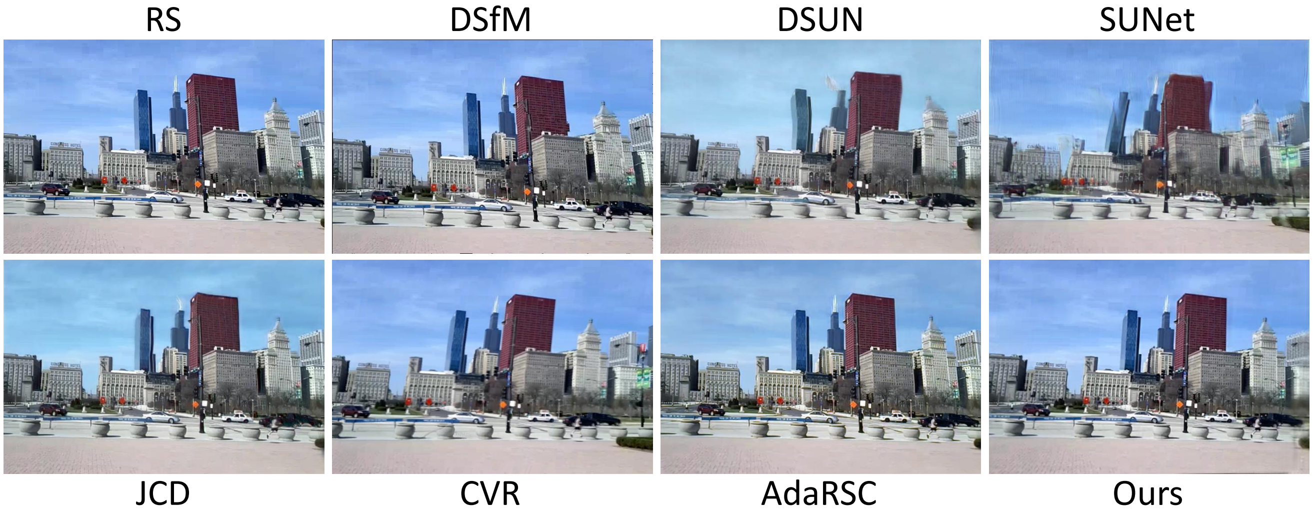

Figure 11: (b) Visual comparison against the state-of-the-art RSC methods in nonlinear motion scenes of Gpark [6] datasets. Existing works fail to handle strong translation and produce local blur.

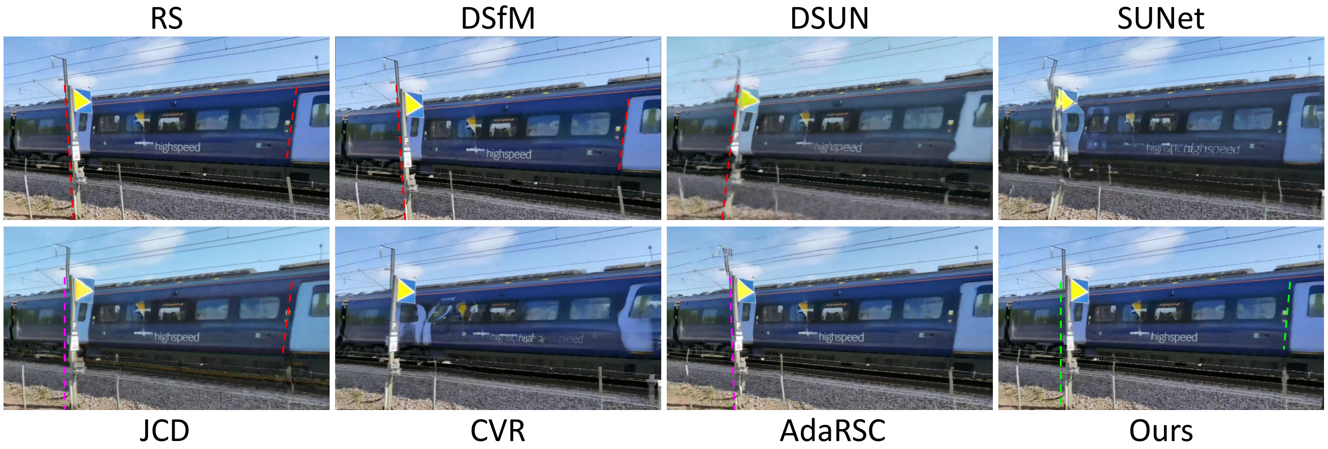

Figure 12: (a) Visual comparison against the state-of-the-art RSC methods in highly dynamic scenes of Train at speed video sequences. The camera was rotating rapidly to film the forwarding train. All existing RSC solutions fail in such a dynamic scene. In contrast, Our method corrects distortions properly.

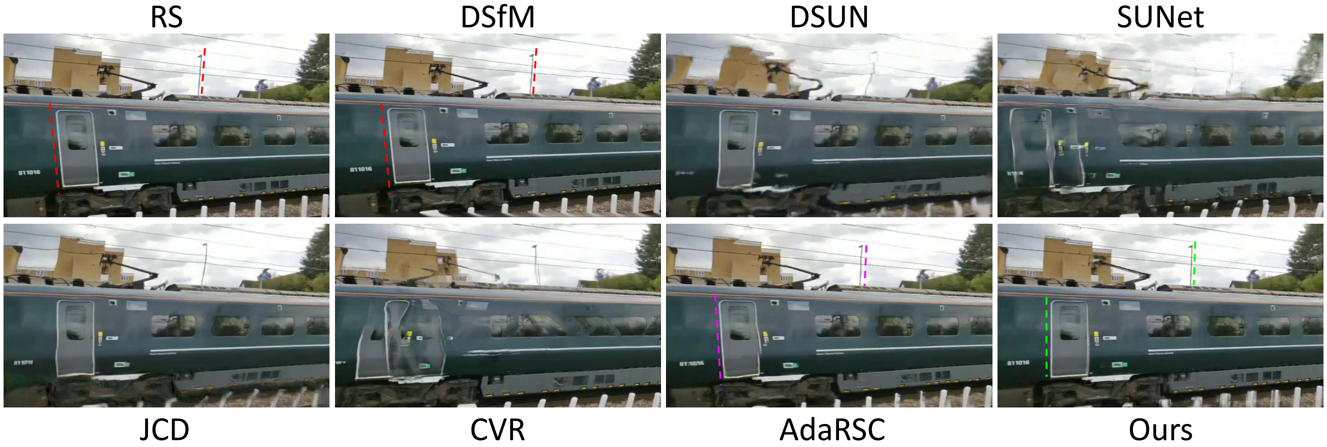

Figure 13: (b) Visual comparison against the state-of-the-art RSC methods in highly dynamic scenes of Train at speed video sequences. The camera was rotating rapidly to film the forwarding train. All existing RSC solutions fail in such a dynamic scene. In contrast, Our method corrects distortions properly.

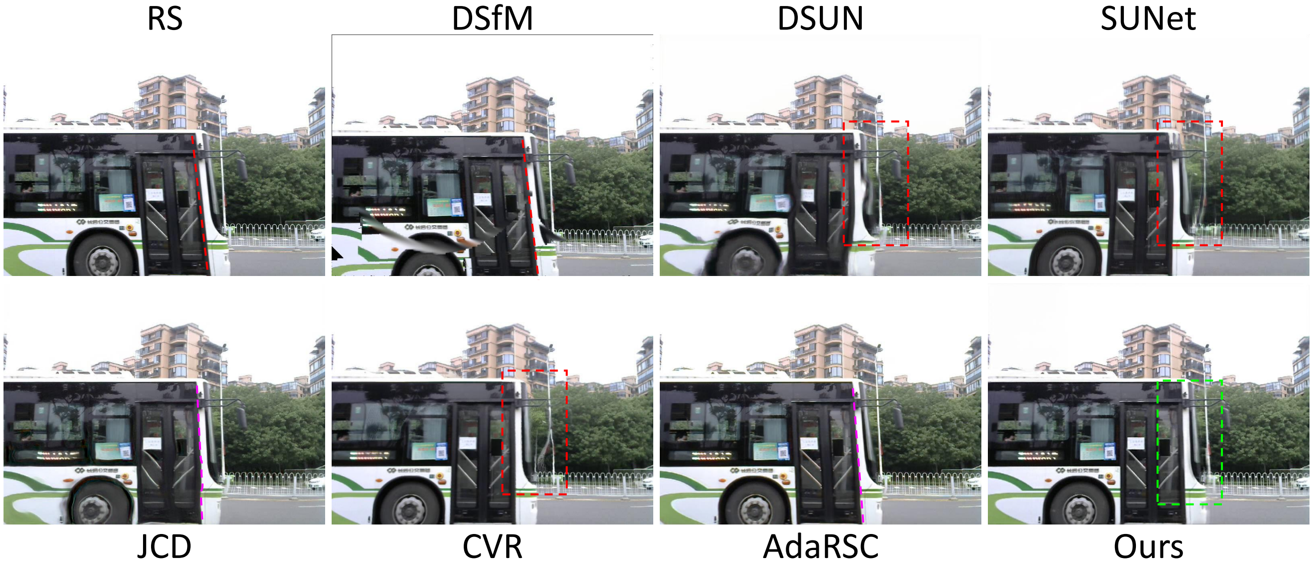

Figure 14: (a) Visual comparison against the state-of-the-art RSC methods in highly dynamic scenes of Bus video sequences. Only the proposed method can rectify the moving bus back to the right position and recover the occlusion, while the others either fail in correction or produce artifacts.

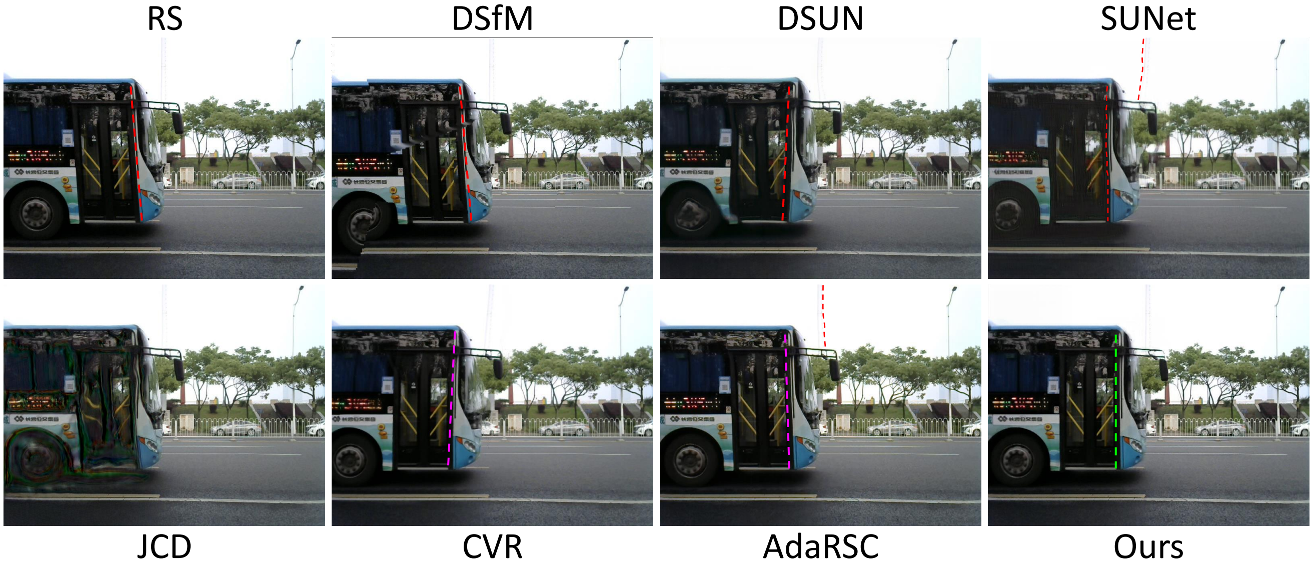

Figure 15: (b) Visual comparison against the state-of-the-art RSC methods in highly dynamic scenes of Bus video sequences. Only the proposed method can rectify the moving bus back to the right position and recover the occlusion, while the others either fail in correction or produce artifacts.