Bringing Rolling Shutter Images Alive with Dual Reversed Distortion – Supplementary Materials

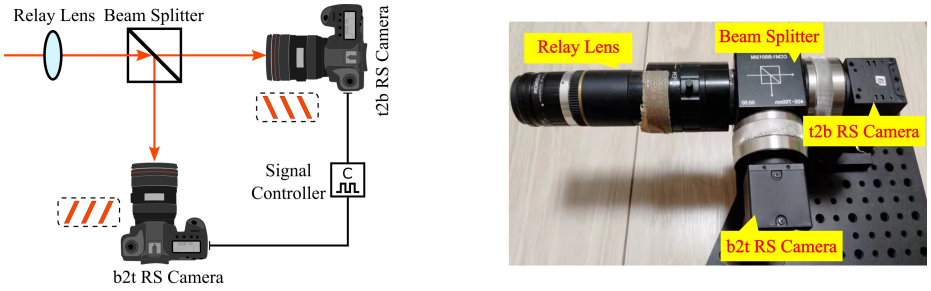
Zhihang Zhong[1,4], Mingdeng Cao[2], Xiao Sun[3], Zhirong Wu[3], Zhongyi Zhou[1], Yinqiang Zheng[1], Stephen Lin[3], and Imari Sato[1,4]

[1] The University of Tokyo, zhong@is.s.u-tokyo.ac.jp
[2] Tsinghua University
[3] Microsoft Research Asia
[4] National Institute of Informatics

1 More Details

1.1 Dual-RS Camera System

To evaluate our method and related works [2,1] on real-world data, we built a beam-splitter-based dual-RS camera system like [4,3], as illustrated in Fig. 1. One camera is installed upside down, such that the two RS cameras have reversed scanning directions. One system with two FL3-U3-13S2C rolling shutter cameras, and the other with two BFS-U3-63S4C rolling shutter cameras have been implemented, with different readout settings. The row-wise exposure time was properly adjusted to avoid motion blur.



(a) (b)

Fig. 1: **Beam-splitter-based dual-RS acquisition system.** (a) is system schematic diagram. (b) shows real system used to collect dual-RS videos.

1.2 Structure of VelocityNet

The details of the subnetwork (VelocityNet) to estimate velocity cube is illustrated in Fig. 2. We totally use 4 subnetworks to iteratively take dual RS images

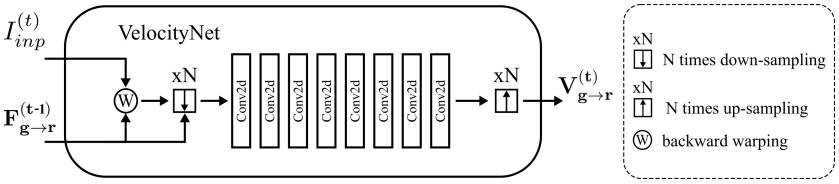


Fig. 2: **Structure of VelocityNet for velocity cube estimation.**

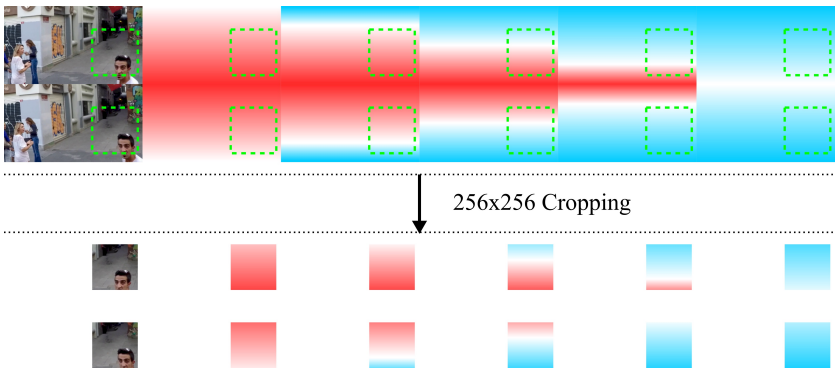


Fig. 3: **Illustration of random cropping for training IFED (f5).**

$I_{inp}^{(t)}$ and previously estimated dual optical flow cube $F_{g\to r}^{(t-1)}$ as inputs for dual velocity cube $\mathbf{V}_{\mathbf{g\to t2b}}^{\mathbf{(t)}}$ estimation. The final velocity cube is equal to the sum of each subnetwork. These sub-networks share the same structure, starting with a warping of the inputs, followed by a series of 2d convolutional layers. All sub-networks have 8 convolutional layers. Before convolution, the scale (resolution) of the warped dual images and optical flow are scaled by linear interpolation. The scale ratio follows a coarse-to-fine manner from the first subnetwork to the last subnetwork, as 1/8, 1/4, 1/2, and 1, respectively. While the dimension of channel is set as 192, 128, 96 and 48, respectively. Note that the initial scale velocity cube estimation is realized without the estimated optical flow cube.

1.3 Training with Cropping

We train the proposed IFED using 256×256 random cropping for data augmentation. Taking the example of extracting 5 frames, the corresponding dual time cube prior will be cropped at the same position, as illustrated in Fig. 3.

2 Additional Results

2.1 Video Results

This work targets extracting undistorted image sequences from rolling shutter images with dual reversed distortion. The closest research to ours is Fan and

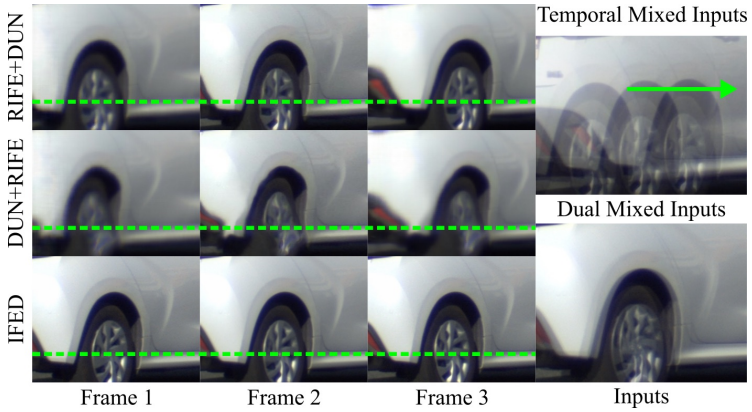


Fig. 4: **Comparison with cascade schemes on real data.** Zoom-in results are shown on the left side of the input. Note that IFED uses the dual mixed inputs and the rest two methods use the temporal mixed inputs.

Dai [2], in which the RSSR algorithm was proposed to extract undistorted frames from two consecutive rolling shutter images. The authors did not release their source code of RSSR, yet kindly agreed to run a few real-world samples for fair comparison. For RSSR, we used two consecutive frames from one camera as input, while for our IFED, we used two dual reserved images (one from each camera) as input. Note that, although RSSR was not trained on our training set, we believe the comparison is fair and meaningful, in the sense that, both algorithms were trained on their own synthetic data, and both were tested on images captured by third-party cameras beyond the training set.

We present video results as `results.mp4`, including results of IFED on RS-GOPRO, as well as results of RSSR and IFED on real-world data. The video clips generated by IFED are more natural and visually appealing, while RSSR cannot generalize on real-world data. RSSR failed because their model only works when there are no moving objects in the scene, a restrictive assumption that rarely holds in practice. Also, the fact that the readout settings are not consistent between the training data and the real test data also poses challenges to RSSR.

2.2 Comparison with Cascaded Schemes on Real-world Data

We show visual results for cascade schemes and IFED on real-world data in Fig. 4 for the case of f3. The results are consistent with those on synthetic data where IFED produces sharper details than RIFE+DUN and DUN+RIFE. Note that IFED even correctly recovers the rotation of the wheel.

2.3 The Effects of The Number Frames Extracted

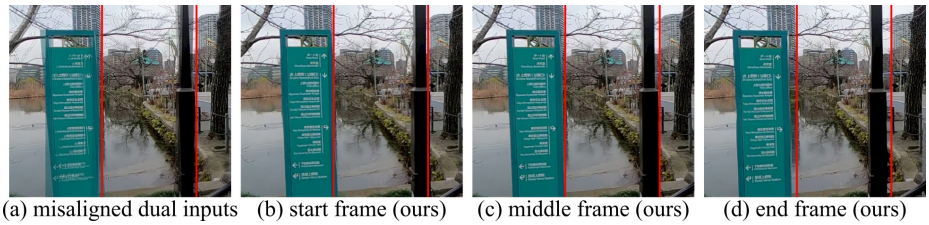
We show the cost of each IFED variant to infer a frame of size 960×540 in Table 1. When we implement models that extract different number of frames, we

	IFED (f1)	IFED (f3)	IFED (f5)	IFED (f9)
Parameters	28.38 M	28.75 M	29.12 M	29.86 M
Runtime	15.11 ms / 8.70 fps	7.26 ms / 13.77 fps	5.95 ms / 16.81 fps	4.16 ms / 24.04 fps

Table 1: **The average inference cost for one frame (960×540) of IFED.** f# denotes the number of frames extracted by the model in a single inference.

	IFED (f1)	IFED (f3)	IFED (f5)	IFED (f9)
1 frame	32.07 / 0.934	30.99 / 0.915	31.21 / 0.919	31.11 / 0.920
3 frames	-	28.48 / 0.87	28.66 / 0.876	28.70 / 0.880
5 frames	-	-	29.79 / 0.897	29.93 / 0.901
9 frames	-	-	-	30.34 / 0.910

Table 2: **Performance of different variants of IFED in terms of PSNR/SSIM.**



(a) misaligned dual inputs (b) start frame (ours) (c) middle frame (ours) (d) end frame (ours)

Fig. 5: **Effectiveness of IFED with clock misalignment.**

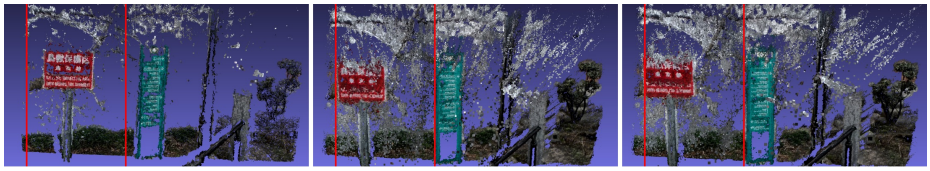
keep the parameters at almost the same level. The test hardware is a GeForce RTX 3090. IFED (f9) can achieve an inference speed of about 24 fps. The quality performance in terms of PSNR/SSIM of each IFED variant calculated on different number of frames is listed in Table 2. When the number of extracted frames is larger than 1, IFED (f9) is able to outperform other variants both in speed and accuracy. One possible reason for this is that training with more continuous ground-truth provides more time continuity to support learning.

2.4 Synchronization of Dual-RS Cameras

Today's clock synchronization circuits usually have errors below 10 µs. In Fig. 5, we show that our method is still effective when frames are misaligned by two rows to simulate out of synchronization.

2.5 3D Reconstruction Evaluation

In this section, we present the evaluation of method on 3D reconstruction task. We implemented SfM (OpenSfM) to generate 3D models using images in the



(a) Reconstructed 3D with RS (b) Reconstructed 3D with ours (c) Reconstructed 3D with GS

Fig. 6: **Reconstruction performance.** 3D model from our corrected and interpolated RS images is closer to the model from GS.

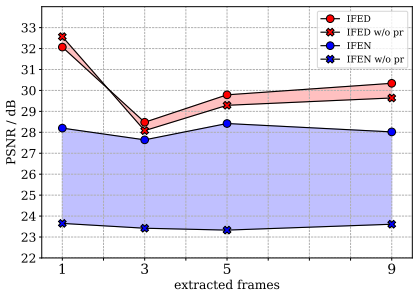


Fig. 7: **Effectiveness of dual-RS setup and time cube prior.**

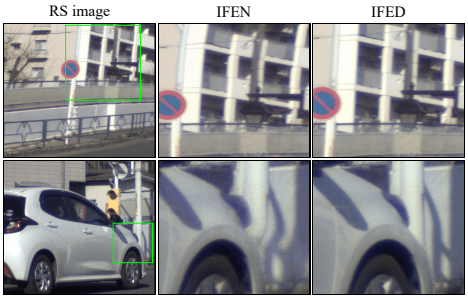


Fig. 8: **The merit of dual-RS setup on real data.**

presence of camera rotations and translations as illustrated in Fig. 6. The structure built by using our RS corrected and interpolated images is closer to the one built by GS image sequence, which validates that our method can further serve to high-level tasks.

2.6 Ablation Studies for Dual-RS Setup

We further show experiments of training our network using consecutive frames on RS-GOPRO, denoted as IFEN. The results also include whether to use time cube prior, as shown in Fig. 7. We can see dual-RS setup bring huge performance gain, and the time cube prior is more helpful in consecutive frame setup. When testing on real images with inconsistent readout settings with the training dataset, the advantage of dual-RS further extends because it avoids the undesired distortion caused by ambiguity, as shown in Fig. 8.

3 Limitations

We have succeeded in handling an electric fan with a spinning speed of up to 500 rpm. However, it is likely to fail for larger objects at the same angular velocity, or smaller objects with faster angular velocity, as both have a faster linear speed around the outer edges.

References

1. Albl, C., Kukelova, Z., Larsson, V., Polic, M., Pajdla, T., Schindler, K.: From two rolling shutters to one global shutter. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2505–2513 (2020)
2. Fan, B., Dai, Y.: Inverting a rolling shutter camera: bring rolling shutter images to high framerate global shutter video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4228–4237 (2021)
3. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: European Conference on Computer Vision. pp. 184–201. Springer (2020)
4. Zhong, Z., Gao, Y., Zheng, Y., Zheng, B.: Efficient spatio-temporal recurrent neural network for video deblurring. In: European Conference on Computer Vision. pp. 191–207. Springer (2020)