Supplementary Material: Learning Structure-and-Motion-Aware Rolling Shutter Correction

Bingbing Zhuang[1] Quoc-Huy Tran[2] Pan Ji[2] Loong-Fah Cheong[1] Manmohan Chandraker[2,3]

[1]National University of Singapore [2]NEC Labs America [3]University of California, San Diego

In this supplementary material, we first provide additional analyses to Sec. 3 and Sec. 4.3 of the main paper. We then present additional results on both synthetic and real data. Finally, we provide additional details of sequences for training and testing and implementations of competing methods.

1. Additional Analyses

1.1. Degeneracy Analysis on Pure Translation with $t_z = 0$

In Sec. 3 of the main paper, we have analysed the degeneracy in RS two-view geometry in the case of pure translation with $t_z \neq 0$. Below we will discuss the remaining case of pure translation with $t_z = 0$. Before that, we would like to clarify that the pure translational camera motion here refers to the camera motion throughout the two images of interest, not just the camera motion within the exposure period of each individual image. Furthermore, $p_i\boldsymbol{t}$ and $q_j\boldsymbol{t}$ represent the per-scanline camera positions in the world coordinate system, which is defined as that of the first scanline in the first image (i.e. $p_1 = 0$), and hence the projection matrices $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ can be expressed as $\boldsymbol{P}_i = [\boldsymbol{I} \;\; -p_i\boldsymbol{t}]$ and $\boldsymbol{P}_j = [\boldsymbol{I} \;\; -q_j\boldsymbol{t}]$.

In the case of pure translation with $t_z = 0$, i.e. camera motion is lateral, we denote $\boldsymbol{T}^{ij} = [T_X^{ij}, T_Y^{ij}, 0]^\top = (q_j - p_i)\boldsymbol{t}$, and still, the 3D points $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ can be related by $\boldsymbol{S}_2 = \boldsymbol{S}_1 - \boldsymbol{T}^{ij}$. Projecting this relationship into 2D images, we get the below equation, which corresponds to Eq. (1) in the main paper,

$$\boldsymbol{s}_2 = \boldsymbol{s}_1 - \frac{1}{Z_1}[T_X^{ij},\ T_Y^{ij}]^\top = \boldsymbol{s}_1 - \frac{(q_j - p_i)}{Z_1}[t_x,\ t_y]^\top. \tag{S1}$$

This equation indicates that all 2D points move in the same direction, i.e. $[t_x, t_y]^\top$, which is also what happens when a GS camera is used. We illustrate such ambiguity in Fig. S1.

Moreover, even if the camera being used for capturing the 2D point displacements is known to be a RS one, the per-scanline camera positions along the translational direction, i.e. p_i and q_j, cannot be determined from 2D correspondences only. This is because, beyond the global scale ambiguity, there are still infinite number of fake p_i' and q_j' that can produce physically possible (i.e. positive) and yet distorted depth $Z_1' = \frac{(q_j' - p_i')}{(q_j - p_i)} Z_1$ such that Eq. (S1) still holds.

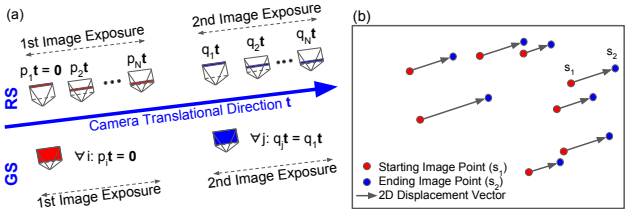

Figure S1. Degeneracy in RS two-view geometry. Both RS and GS lateral translation in (a) produce parallel displacement in the 2D points in (b). The red and blue lines in (a) represent the scanlines in the image planes.

We note here that assuming constant velocity throughout the exposure period of the two images does not remove the degeneracy. If the two images are taken from two consecutive frames from a video and the readout time is further assumed known, as was done in [5], the degeneracy disappears. However, readout time calibration is nontrivial; thus, this requirement poses significant restrictions on the applications.

1.2. Ambiguity between w_x-Induced Distortion and Vertical Image Resizing

In Sec. 4.3 of the main paper, we have discussed this ambiguity under the case of pure rotational camera motion w_x for ease and clarity of explanation. One should realize that, for the more general 6-DOF camera motion that is simulated for each RS image in the training data, such confounding is reduced due to the (un-)distortion flow induced by the other 5 DOFs. However, the overall confounding effect of this ambiguity still exists, and as shown empirically in Sec. 5.1 of the main paper, the training (on RS images with 6-DOF camera motions) is still affected. We also note that since resizing changes the focal length associated with the image, it is thus possible to distinguish the resizing effect by embedding the appropriately updated focal length into the training; however, cropping is a more straightforward solution.

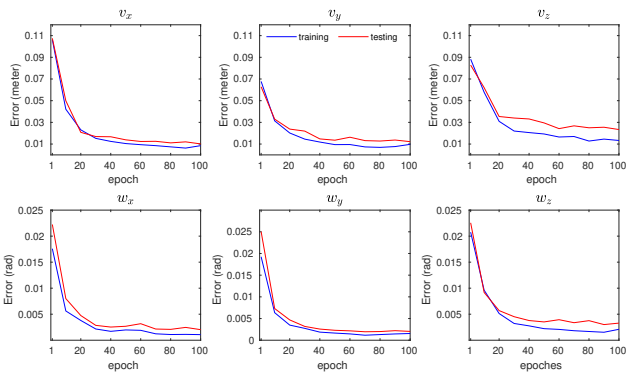

Figure S2. Training and testing errors at different epochs.

2. Additional Results

2.1. Camera Velocity Estimation Errors

Although the quality of RS rectification is the main focus of our evaluation, to better understand the behavior of our method, we also report camera velocity estimation errors at different epochs during training and testing. To make the results more intuitive, we plot the errors of $(N-1)\boldsymbol{v}$ and $(N-1)\boldsymbol{w}$, which correspond respectively to the total translation and total rotation between the first scanline and last scanline. As shown in Fig. S2, despite its ill-posed nature, our network can indeed learn to predict reasonable camera motion from the distortion in the image appearance.

2.2. Additional Results on Synthetic Data

We show a few additional results on synthetic RS images in Fig. S3. In particular, (a) represents a typical example in which MH and 2DCNN are only able to rectify some regions of the image (e.g. see red boxes) but not for other regions (e.g. see blue boxes), whereas our method can undistort the entire image relatively well. (b) shows a typical example where the distortions are subtle, however, our method can predict the camera motion and remove most of the subtle distortions effectively. (c) and (d) represent two cluttered scenes, which pose extra challenges for RS correction. Nevertheless, the results show that our method is able to extract the underlying geometry from the complicated scenes and achieve satisfactory rectification results.

2.3. Additional Results on Real Data

Fig. S4 presents some additional results on real RS images. In general, our method achieves superior performances compared to those of MH and 2DCNN. In particular, we note that although all methods can rectify the distortion in the background house in (a) reasonably well, only our method can remove the marginal deformation in the rear of the car (i.e. see red boxes). In (b), one can clearly see that the rectangular shape of the street sign is best recovered by our method (i.e. see red boxes). In (c), the curves are rectified as straight lines more effectively by our method (e.g. see red boxes). (d) represents a typical example where MH may return grossly erroneous results if the lines forming the Manhattan world are not presented in the input image or not detected properly. In addition, although the pole remains slightly bended in our rectified image, it is still visually better than that of 2DCNN. Similarly, superior performances of our method are observed in (e), (f) and (g).

3. Additional Details

3.1. Sequences for Training and Testing

We exploit the 'City' and 'Residential' categories of the KITTI Raw dataset [1] to generate synthetic RS images for training and testing. In particular, for rendering testing images, we use the 2 sequences '2011_10_03_drive_0027' and '2011_09_29_drive_0071', which are representatives of uncluttered and cluttered traffic scenes respectively and comprise more than 5,500 frames. For synthesizing training images, we use all the remaining sequences in the 'City' and 'Residential' categories except for the 5 sequences where the scenes are mostly static (the vehicle is stopped), including '2011_09_26_drive_0017', '2011_09_26_drive_0018', '2011_09_26_drive_0057', '2011_09_26_drive_0060', and '2011_09_28_drive_0002', This results in totally 42 sequences with around 30,000 frames for generating training images.

3.2. Implementations of Competing Methods

For both MH [2] and 2DCNN [3], we use the source codes provided by the authors in all of our experiments. The rectified images are taken directly from the outputs of their codes for qualitative evaluation. In addition, we update their codes to produce the undistortion flows for quantitative evaluation.

References

[1] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[2] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in manhattan world. In *ICCV*, 2017. 2

[3] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *CVPR*, 2017. 2

[4] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 3

[5] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutteraware differential sfm and image rectification. In *ICCV*, 2017. 1

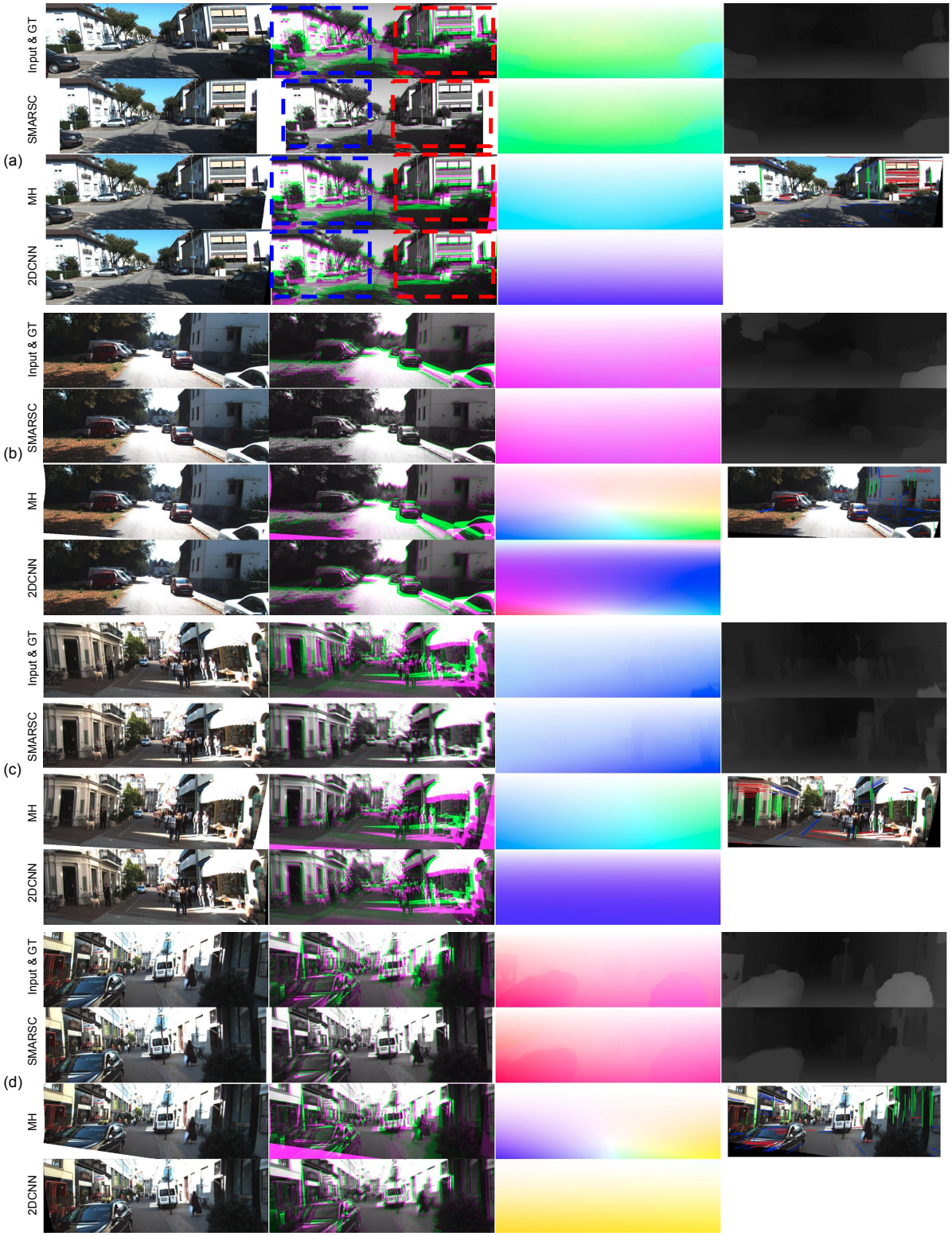

Figure S3. Additional qualitative comparisons on synthetic RS images. For each example (a)-(d), the first row shows the input RS image, input RS image overlaid on ground truth GS image, ground truth undistortion flow, and ground truth depth map respectively, while the next three rows plot the results of our method (SMARSC), MH, and 2DCNN respectively with each row showing from left to right the rectified image, rectified image overlaid on ground truth GS image, estimated undistortion flow, and estimated depth map. Note that since MH and 2DCNN do not predict depths, we instead show the line detection result for MH (different colors indicate the associations with different vanishing points) and leave an empty figure for 2DCNN. The undistortion flow is visualized following [4]. In the depth map, bright and dark colors mean small and large depth values respectively.

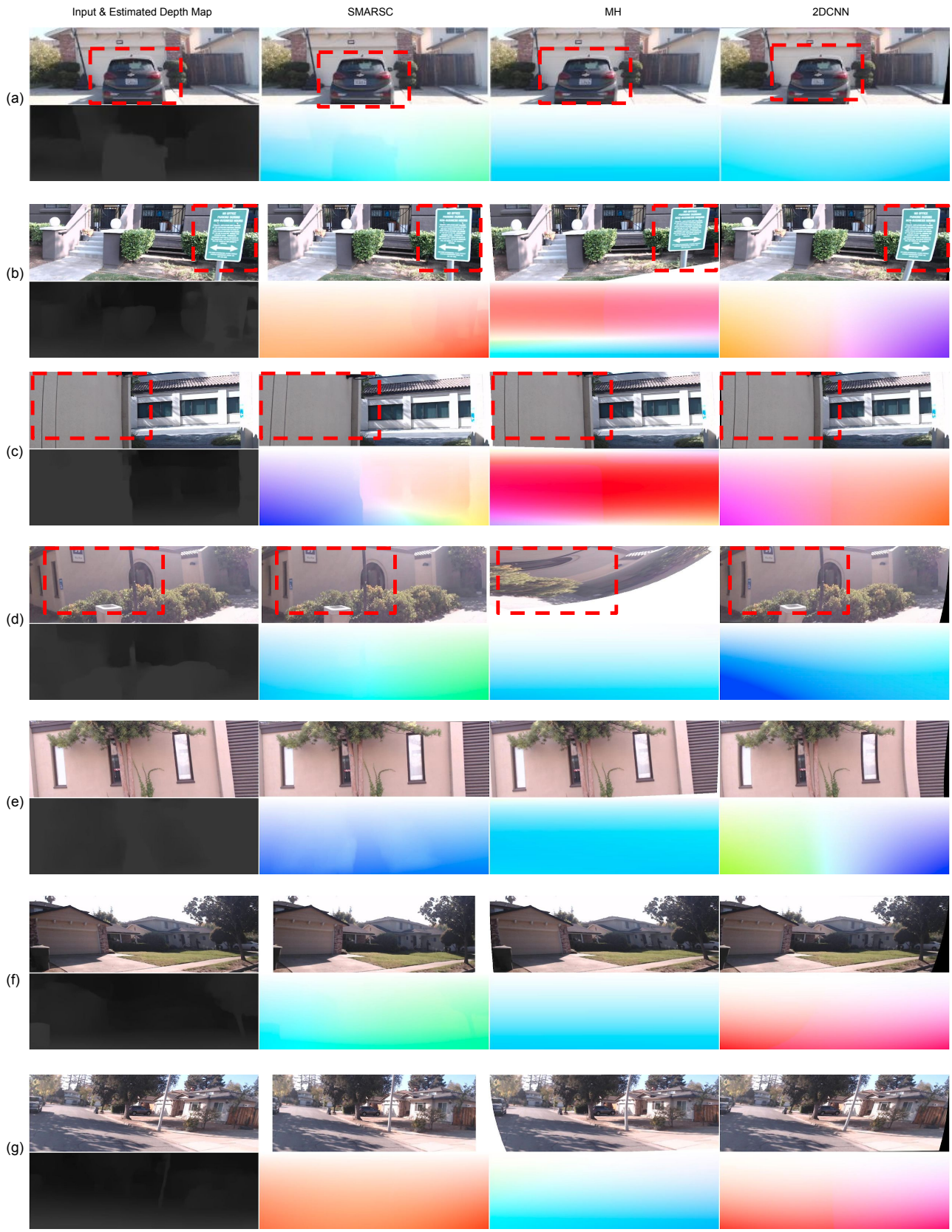

Figure S4. Additional qualitative comparisons on real RS images. For each scene (a)-(g), the first row shows from left to right the input RS image and the rectified images by our method (SMARSC), MH, and 2DCNN respectively. The second row shows the estimated depth map by our method and the undistortion flows by our method, MH, and 2DCNN respectively.