Deep Homography Mixture for Single Image Rolling Shutter Correction

Weilong Yan[1] Robby T. Tan[2] Bing Zeng[1] Shuaicheng Liu[1]*
University of Electronic Science and Technology of China[1]
National University of Singapore[2]
1092443660ywl@gmail.com, robby.tan@nus.edu.sg, {eezeng, liushuaicheng}@uestc.edu.cn

Abstract

We present a deep homography mixture motion model for single image rolling shutter correction. Rolling shutter (RS) effects are often caused by row-wise exposure delay in the widely adopted CMOS sensor. Previous methods often require more than one frame for the correction, leading to data quality requirements. Few approaches address the more challenging task of single image RS correction, which often adopt designs like trajectory estimation or long rectangular kernels, to learn the camera motion parameters of an RS image, to restore the global shutter (GS) image. In this work, we adopt a more straightforward method to learn deep homography mixture motion between an RS image and its corresponding GS image, without large solution space or strict restrictions on image features. We show that dividing an image into blocks with a Gaussian weight of block scanlines fits well for the RS setting. Moreover, instead of directly learning the motion mapping, we learn coefficients that assemble several motion bases to produce the correction motion, where these bases are learned from the consecutive frames of natural videos beforehand. Experiments show that our method outperforms existing single RS methods statistically and visually, in both synthesized and real RS images. Our code and dataset are available at https://github.com/DavidYan2001/Deep_HM.

1. Introduction

Rolling shutter (RS) refers to the effects caused by different row exposures that has been widely adopted in CMOS sensors. Such a row-wise exposure often introduces artifacts, such as blending of straight lines and skewing of image contents, which are not only visually unpleasant but also harmful for downstream tasks [9, 26, 1, 25]

Existing RS rectification methods can be categorized into: multi-frame [3, 13, 2, 15, 31] and single-frame [22, 21, 20, 38, 10]. Correcting RS from a single image is more challenging but important under data-starved situations. Most existing single RS methods are non-learning methods (e.g., [22, 20, 12, 19]), only few works are learning-based ones (e.g., [21, 10]). Non-learning methods rely on prior assumptions for the correction, such as "straight lines remain straight". However, they often fail when such salient priors are not available in an image. In contrast, many deep learning methods have been proposed but they are based on multi-frame approaches. Only few methods target on single RS correction, which is inherently an ill-posed problem.

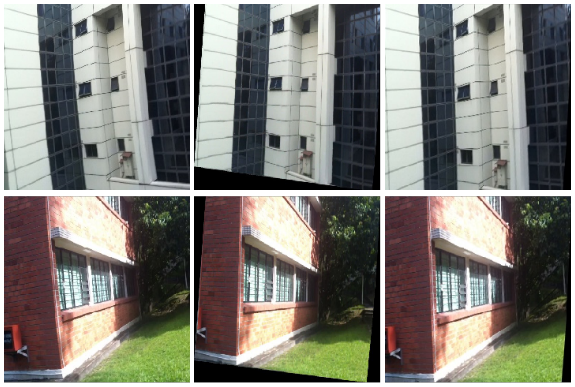

Figure 1. Rectification result on real examples. Column 1: real RS images. Column 2: results by [37]. Column 3: our results.

Existing deep-learning single RS methods learn the row-wise camera motion between RS and global shutter (GS) pairs (e.g., [21]). However, row-wise motion introduces a large solution space that increases the learning difficulty. To facilitate the learning, [38] requires an additional depth as the input, where the depth can be estimated by an off-the-shelf method. Unfortunately, the quality of the correction is affected by the quality of the estimated depth, and estimating depth from a single image is still an open problem.

In this paper, we present a novel method that learns a deep homography mixture (HM) motion model for single RS correction. HM [6] is originally proposed to align adjacent frames for the task of joint video RS removal and stabilization. An HM divides a frame into several equally spaced

[*] Corresponding author

horizontal blocks. Each block follows a homography transformation with Gaussian smoothing of neighboring blocks for the spatial smoothness. The model holds when **1)** depth of the scene is plane or at infinity, **2)** camera motion across rows maintains piece-wise smoothness. Previously, the HM [6] is estimated by detecting [24] and tracking [28] image feature points between consecutive frames, and solve multiple smoothed homographies with a DLT [7]. Unlike these methods, we use the model as the motion correction model under our single RS rectify scenarios. Directly learning the homography mixture between RS and GS cannot produce satisfactory results. Recently, BasisHomo [34] proposes a method to learn deep homography by combining 8 pre-defined homography flow bases, each of which is a flow map created by modifying one of a homography matrix element. In this way, the learning of a homography is converted to the learning of coefficients corresponding to each flow basis, which demonstrates superior performances compared to directly regressing homography matrix elements, or the 4pt motion vector representation [4]. Here, we adopt flow basis representation, not for the frame registration [34], but for the estimation of rectifying motion as HM between GS and RS. Moreover, we notice that pre-defined bases are not optimal. We propose to learn these bases from natural videos. PCA-Flow [32] shows that optical flow can be estimated by first learning several optical flow bases from a movie and then combine them for the flow estimation. These bases are extracted by the PCA [8]. In this work, instead of learning the complex basis with object-level motion details as in optical flow [32], we learn global homography flow basis that reflects camera motions.

On the other hand, many of the rolling shutter datasets are designed for multi-frame cases. We follow the single RS method [21] to synthesize RS and GS pairs. We notice that many of the existing datasets contain rich textures, full of salient lines, such as urban scenes [33]. A potential reason is that rich salient lines are more friendly for the RS correction. Here, we move a step further by creating a new dataset named as RS-Homo, based on the CA-Homo dataset [35], which is designed for homography estimation of two images, and contains many adverse scenarios, such as poor texture and low light. Trained on RS-Homo, we show that our method can work well not only for synthesized images in many scenarios but also for real RS captures.

In summary, our main contributions are:

- We propose deep homography mixture motion model for the task of single image rolling shutter correction, which neither requires camera intrinsics nor the additional IMU hardware.
- We introduce a pipeline that learns the motion mapping between GS and RS by combining motion bases, which are learned from natural videos. We train our network on the proposed RS-Homo dataset, delivering high quality results even under adverse cases.
- Our method achieves state-of-the-art performances when compared to previous single-frame approaches, with 2.7% higher SSIM and 56% lower motion RMSE (per pixel). The ablation study verifies the effectiveness of each component.

2. Related Works

Classical Single Image Methods Classical single RS correction methods rely heavily on the image contents, e.g., salient straight lines. Rengarajan *et al.* [22] rectified an RS image by enforcing constraints of straight lines should remain straight. Purkait *et al.* [20] assumed Manhattan world assumption. Lao *et al.* [12] used four straight lines to estimate the camera motion. Our deep network is free from these strong assumptions, and can work in natural scenes.

Classical Multi-Image Methods Multi-image methods can utilize temporal information for rectifications where accurate image alignment becomes important. Liang *et al.* [13] estimated per-pixel motion vectors. Forssèn *et al.* [23] adopted KLT [28] for image feature tracking. Karpenko *et al.* [11] adopted the gyroscope to estimate the rotational motion. Baker *et al.* [2] estimated motions with up to 30 row blocks with affine or translational motion model. Besides, 3D information can also be utilized. Zhuang *et al.* [37] estimated SfM from RS frames. Vasu *et al.* [31] addressed the occlusion issue in RS with the estimation of a latent layer mask. Some methods take an entire video as input, which optimizes RS and video stabilization jointly. Grundmann *et al.* [6] first proposed the HM for the registration of neighboring frames with RS rectified during stabilization. In contrast, we only have one frame, thus there is no image alignment. We use HM as the rectification model instead of registration model.

Deep Multi-Image Methods Deep RS network often consists of an encoder for the feature extraction, a rectification module and a decoder for result reconstruction. Fan *et al.* [5] adopted PWC-based flow estimator and warped the features to GS image decoder for the rectification. Liu *et al.* [15] took two consecutive frames as input, and estimated a pixel-wise velocity field and applied a differentiable forward warping for the deep frame unrolling. Zhong *et al.* [36] not only corrected the RS, but also deblurred the frames jointly. In this work, we target the more challenging single frame RS correction.

Deep Single Image Methods Only few works address the task of deep single RS correction. These works train the network with GS and synthesized RS pairs. Rengarajan *et al.* [21] adopted simple affine motion model. Zhuang *et al.* [38] additionally estimated a depth map from a single

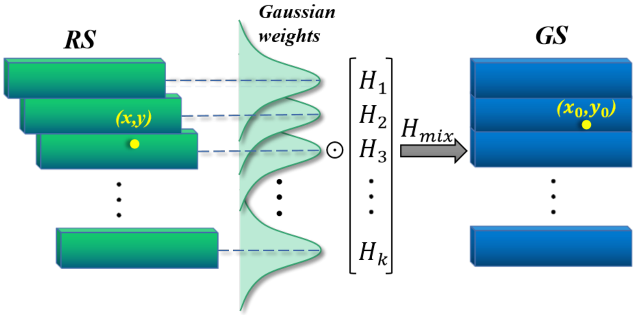

Figure 2. The illustration of Homography Mixture motion model. The transformation of a point (x, y) not only corresponds to its belonging block with homography H_i, but also affected by neighboring homographies gaussian weighted within a certain range.

image. Besides, additional sensors can be adopted, such as IMU [16]. In this work, we adopt HM for the rectification. We show that it is much easier to learn indirect motion mappings by assembling motion bases compared to directly output the correction motion.

3. Proposed Method

3.1. Homography Mixture Model

Considering the camera model, a 3D location X can be imaged by the corresponding projection matrix $P \in \mathbb{R}^{3\times3}$ by $x = PX$, where P can be decomposed into a camera intrinsic matrix K and the camera pose parameters R and T by $P = K[R|T]$ and x is the projection of X on the image.

In the case of rolling shutter, assuming that the depth of the plane is on a plane or at infinity, each row corresponds to a certain P. Considering a position x with its homogeneous coordinate $(x, y, 1)^T$ on an RS image and the matched pair x_0 with $(x_0, y_0, 1)^T$ on the related GS image, we have

$$x = P_{\text{RS}}X, \quad x_0 = P_{\text{GS}}X \tag{1}$$

From Eq. (1), x and x_0 can be related as follows:

$$x_0 = P_{\text{GS}}P_{\text{RS}}^{-1}x \Rightarrow x_0 = Hx, \tag{2}$$

where $H \in \mathbb{R}^{3\times3}$ is seen as a homography matrix [7]. Thus, each row corresponds to one homography matrix with the assumption above. This may lead to some limitations when captured scene has depth variations. In Sec. 5.4, we find that the method still effective in this kind of image. Moreover, estimating the homography matrices of all the rows is unnecessary, given an assumption for the piecewise smoothness of the camera motion across rows. Therefore, we divide an RS image into k blocks and estimate $H_i, i = 1, 2, ..., k$ for each block instead. As shown in Fig. 2, a Gaussian weight is adopted to align the scanline of each block, which maintains continuities between different blocks. Having k homography matrices matched to k blocks, the homography mixture for x can be defined as:

$$H_{\text{mix}} = \sum_{i=1}^{N} \omega_i H_i \Rightarrow x_0 = H_{\text{mix}}x, \tag{3}$$

where ω_i expresses the normalized Gaussian weight corresponding to the i^{th} block. Fig. 2 shows an example.

3.2. Our Network Pipeline

Our network takes a single rolling shutter image as input, and outputs N numbers of coefficients for each homography block, yielding $k \times N$ in total as the network outputs. These coefficients combine N numbers of motion bases into k motion maps for k blocks. The motion bases are learned from consecutive frames in natural videos. To obtain the correction maps, we multiply the normalized Gaussian weights to former motion maps to build final homography mixture motion. Bilinear interpolation method is adopted with the motion output to make corrections. This mixture motion is adopted into rolling shutter correction. The pipeline structure can be seen in Fig. 3.

Network Architecture We adopt a VGG-style network [29] as our backbone, which takes a single RGB image of size $H \times W \times 3$ as input. After the convolutional layers and maxpool layers, the feature map is flattened and sent into 2 fully connected layers and finally a bases weight layer of size $k \times N$ is acquired. Note that, there are k blocks with each block corresponding to N weights in the output layer.

Homography Flow Bases A homography matrix can be represented as 4 corner offsets [4, 35]. However, this representation cannot work well in our single image RS rectification task as we will show in our ablation studies. A homography matrix can be converted into a flow map given the image coordinates, yielding a homo flow representation. We show that it is more accurate for neural networks to predict basis coefficients than the 4 corner offsets. Instead of directly learning the homo flows, we learn coefficients that combine motion bases as shown in [34]. The difference is that, [34] combines 8 pre-defined bases, whereas our bases are learned from real data as we will illustrate in Sec. 3.3.

After predicting the $k \times N$ bases weights from our network, they are divided into k groups and each group is sent to multiply the learned bases by dot product as follows:

$$\text{flow}_i = \sum_{j=1}^{N} \alpha_{ij} h_j (i = 1, 2, ..., k), \tag{4}$$

where flow_i of size $H \times W \times 2$ is the homography motion flow for the i^{th} block, h_j of size $H \times W \times 2$ is the j^{th} learned basis and α_{ij} expresses the predicted coefficient to the j^{th} learned basis in the i^{th} block. The bases are visualized in Fig. 3. Hence, the k homography motion

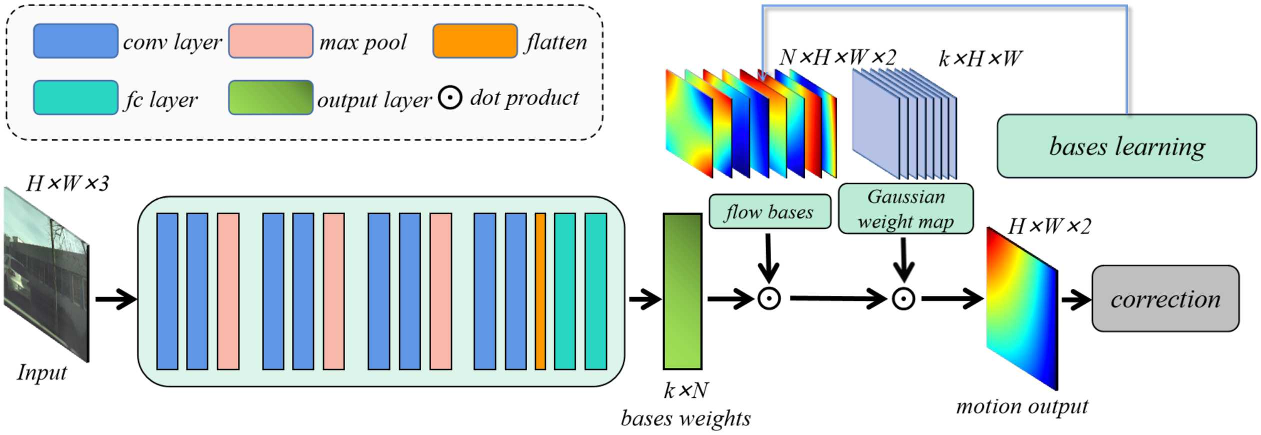

Figure 3. Our system pipeline. Our network takes a single rolling shutter image of size $H \times W \times 3$ as input, and outputs $k \times N$ bases weights, where k is the number of blocks within a frame, and N is the number of bases. These weights combine N flow bases with Gaussian weights for the spatial smoothness to create a motion output, which corrects the input image for the result. These flow bases are learned from videos beforehand. For training, we calculate flow differences between motion outputs and synthesized ground-truth motions.

flows of size $H \times W \times 2$ corresponding to k blocks are acquired.

Gaussian Weight Map As explained in Sec. 3.1, a Gaussian weight is more appropriate for the continuities between blocks. The predefined Gaussian weight maps of size $k \times H \times W$ are combined with the homography motion flows:

$$m = \sum_{i=1}^{k} \text{flow}_i \circ \text{map}_i, \tag{5}$$

where the entire $H \times W \times 2$ output m is the mixture motion flow that is used for rolling shutter correction, flow_i is previously obtained motion flow and map_i is the Gaussian weight map corresponding to the i^{th} block homography motion flow. $\circ$ expresses the element-wise multiplication. Noted that all weights with respect to the same location in an image are normalized before the combination.

Flow Loss Function In our method, the motion flow between RS and GS images is essential for the correction. Considering the endpoint error (EPE), we design the following flow loss function between the predicted motion flow m and the ground truth synthesized motion flow M:

$$\mathcal{L} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} \|m_{ij} - M_{ij}\|_2. \tag{6}$$

Thus, our goal is to minimize the mean EPE between m and M by predicting the bases coefficients.

3.3. Motion Basis Learning

Basis learning is an interesting and effective method in low-rank representations [14]. Tang *et al.* [30] illustrates that low-level vision problems are potential to be solved by discovering the subspaces. PCA-Flow [32] learns the optical flow bases from real movies and demonstrates that flow estimation can be the combination of weighted bases. BasisHomo [34] adopts 8 pre-defined homography flow bases to the homography estimation task. Following this, we learn the global homography flow bases from natural videos in the CA-Homo dataset showed in [35].

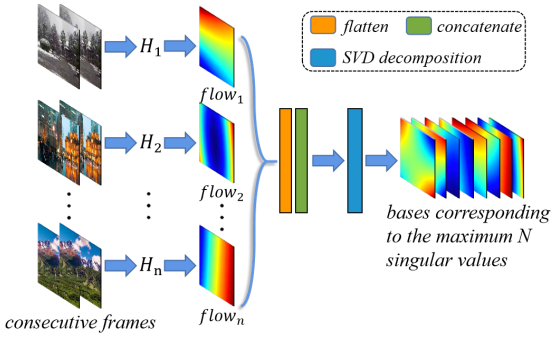

Figure 4. Motion Basis Learning. We estimate homography H_i from pairs of images extracted from various type of videos. The homographies are converted into homography flows, which are then flattened and concatenated before SVD. The bases are column vectors that correspond to the top N singular values.

As shown in Fig. 4, we extract n pairs of consecutive frames from natural videos. For the stability and robustness reasons, we choose the regular(RE) video category [35]. Frame pairs are selected from videos with random skip, creating different rates. We adopt SIFT/SURF and RANSAC algorithms to n RE frame pairs to estimate homography matrices and transform the matrices into n $H \times W \times 2$ homography flows.

The flows are flattened and concatenated to a $2HW \times n$ combination matrix F which is processed by SVD decom-

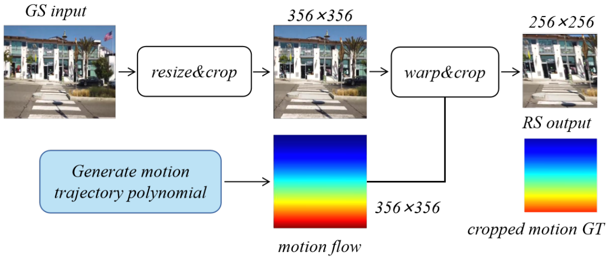

Figure 5. Dataset Generation. We collect images from CA-Homo dataset [35] and synthesize motion flow as GT flow label by trajectory polynomial [21]. The GS image is warped based on the motion flow to create RS image as the network input.

position as $F = U\Sigma V^T$, where U is a matrix of size $2HW \times 2HW$ containing the flatten bases, Σ is a $2HW \times n$ matrix with all the singular values and V is a $n \times n$ matrix. Noted that U and V are both orthogonal matrices. With PCA analysis [8], only N flatten bases corresponding to the maximum N singular values are chosen as the motion bases in our method. Therefore, N motion bases of size $H \times W \times 2$ in Fig. 3 are obtained.

4. Dataset Generation

Since most of the available datasets are designed for multi-frame cases and capturing real RS-GS pairs is difficult, we use synthesized camera motion from a second-degree polynomial for RS-GS image pair generation. As shown in Fig. 5, an original image is first resized and cropped into a $356 \times 356 \times 3$ image with maintenance of proportional structure. With the generation of motion trajectory polynomials, a ground-truth motion flow of size $356 \times 356 \times 2$ can be obtained. The flow is used to warp the processed image and finally a cropped RS image of size $256 \times 256 \times 3$ is output. The flow is also cropped in size of $256 \times 256 \times 2$ as the ground truth. Noted that there are no missing parts in the boundaries of the generated RS images.

5. Experiments

We evaluate our method on 3 datasets. As the previous single RS image correction methods [21, 10] evaluate their results on the building dataset [33, 27, 18], we adopt the same datasets for fair comparisons with previous single RS methods. In addition, we use the Carla-RS dataset [15] to compare with multi-frame methods [15, 5] as a reference. Moreover, we generate the **RS-Homo** dataset based on the CA-Homo dataset [35]. We notice that trained on this dataset can make our method works even in adverse cases such as low light and poor texture. Not that, many of the previous methods are not open-sourced, we have to reuse many reported experiment results from the published papers [21, 10, 5].

5.1. Implementation Details

For building datasets, there are 6,000 clean images. Following [21, 10], we randomly choose 2,000 images with each image of 150 synthesized motions from camera motion parameters to form the training data of size 300k and 10 sythesized motions for the other different 40 images as the testing data. For RS-Homo dataset, we randomly choose 5 frames for each of the 218 videos and each frame is warped with 100 synthesized motions to constitute the training data of size 109k. In 32 test videos, we randomly choose one frame from each video and generate 10 motions for each frame to create a test dataset. Our network is trained with 100k iterations by the Adam optimizer [17] with $l_r = 10^{-4}$. The batch size is 16 and the l_r is reduced by 20% for every 5k iterations. The implementation is based on PyTorch and trained on two 1080 Ti. The inference time of our model on a single test image is 21.9ms.

5.2. Comparison with Existing Methods

We mainly compare our method with previous related RS methods which only require image as input on the building dataset [33, 27, 18]. Traditional approaches include Rengarajan *et al.* [22], Grundmann *et al.*[6] and Purkait *et al.* [20]. The learning-based methods include Rengarajan *et al.*[21] and Kandula *et al.*[10]. Although not that fair, we also compare with some recent mult-frame methods as a reference, DSUN [15] and SUNet [5] on Carla-RS dataset [15].

Quantitative Comparison For fair comparisons, peak signal-to-noise ratio (PSNR)(dB), structural similarity index measure (SSIM) and endpoint error (EPE) are used as metrics to measure the similarity between the RS rectified image and the GS image as adopted by previous methods. Pixels with no information are neglected in calculation. We report the results in Table 1. Traditional approaches rely on curve detection [22, 20] to realize motion parameters estimation, which usually fails to correct RS images of adverse cases without salient structures, leading to lower PSNR and SSIM and higher EPE. The RS video correction method [6] works only when frames contain rich texture or appropriate light. The learning-based methods [21, 10] can sometimes work in images with strong outliers, but large solution space for the camera motion parameters increases the difficulty of network prediction. Since our method adopts combination of learned homography flow bases, it does not suffer from the mentioned weaknesses and achieves higher PSNR and SSIM and lower EPE.

When comparing with multi-frame methods [15, 5] and a single-frame method [38] on Carla-RS dataset [15]. Noted that [38] needs depth for training, thus it is not involved in the building dataset experiment. Moreover, in Carla-RS, there is only the ground truth GS images at the centered

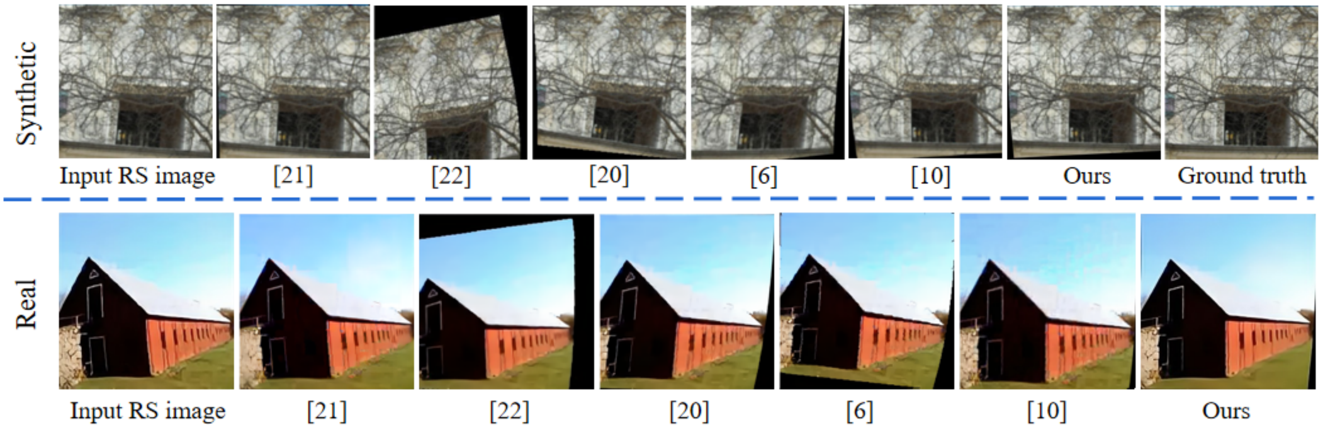

Figure 6. Comparison with previous methods [22, 20, 6, 21, 10]. Top and bottom are synthetic and real examples.

Method	PSNR(dB)↑	SSIM↑	EPE↓
Rengarajan *et al.*[22]	29.82	0.67	11.89
Purkait *et al.*[20]	29.22	0.55	8.32
Rengarajan *et al.*[21]	32.25	0.70	3.76
Grundmann *et al.*[6]	32.57	0.72	3.34
Kandula *et al.*[10]	32.85	0.73	2.84
Ours	**33.34**	**0.75**	**1.25**

Table 1. Comparison of PSNR, SSIM and EPE between ours and other single RS image methods on building dataset [33, 27, 18].

Method	PSNR(dB)↑	SSIM↑
Single-frame[38]	18.47	0.58
DSUN[15]	26.46	0.81
SUNet[5]	29.18	0.85
Ours	**20.62**	**0.62**

Table 2. Comparison of PSNR and SSIM between ours and multi-frame methods [15, 5] on Carla-RS dataset [15]. Noted that although we do not outperform multi-frame methods, ours is better than previous single-frame method [38], which requires depth information as the additional inputs.

time between two RS frames. The multi-frame methods use two consecutive RS frames to restore the GS frame at the centered time, while ours aims at restoring the GS frame at the time when the first row of a single RS frame is captured. From Table 2, it can be seen that the metrics of ours are lower than the multi-frame methods for the time difference mentioned above. However, the PSNR and SSIM are obviously higher than the single-frame method [38].

Visual Comparison Fig. 6 illustrates rectified results in both synthetic RS image and real RS image. Considering the synthetic RS image in the first row, because of the occlusion from branches, traditional methods [22, 20, 6] can not work well under this situation. Previous learning-based methods [21, 10] show better results, but the contours are seen to be curved. Our result can display straight contours even in this occlusion condition. For the real RS image in the second row, part of the house is covered with straw and there is low texture with the sky at infinity, resulting in failed correction with methods [22, 20, 6]. The learning-based method [21] also fails. Results of another learning-based method [10] is similar to our result, while it is visually more skewed than ours. Fig. 7 displays results compared with multi-frame methods [15, 5]. They take RS frame 1 and 2 as input and ours only input RS frame 2. The result can be seen as competitive and even clearer than method [15].

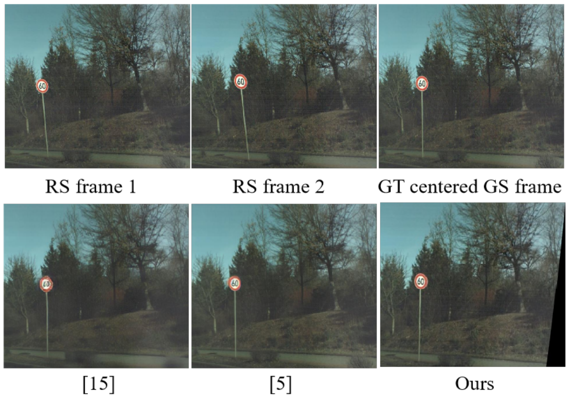

Figure 7. Comparison with multi-frame methods [15, 5]. The multi-frame methods take RS frame 1 and 2 as input, while ours only inputs RS frame 2. The ground truth GS image is captured in the centering time of capturing RS frame 1 and 2.

5.3. Ablation Studies

The ablation study is carried on the proposed RS-Homo dataset. There are 5 types of images in CA-Homo [35], so does ours. They are regular (RE), low texture (LT), low light (LL), small foreground (SF) and large foreground (LF) scenes. We use PSNR, SSIM, EPE as our metrics in ablation study. Table 3 reports our ablation results.

Component		PSNR↑	SSIM ↑	EPE↓
weight	average	23.74	0.73	5.87
	single	24.86	0.74	5.18
	Gaussian	26.15	0.77	4.10
block	4	25.10	0.75	4.25
	8	26.15	0.77	4.10
	12	25.61	0.76	4.35
offset and basis	offsets	18.37	0.61	15.58
	fixed 8	25.95	0.76	4.27
	learned 8	26.15	0.77	4.10
	learned 12	25.88	0.76	4.11

Table 3. Ablation study with different components including weight, block, offset and basis. The table shows results for 5 categories in RS-Homo, RE, LT, LL, SF, LF. The optimal choice of components is emphasized.

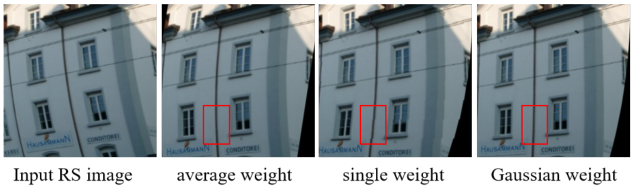

Figure 8. Comparison among different weight maps. With a single RS image on the left as input, the following are results with average weight map, single weight map, Gaussian weight map.

Weight Maps We design 3 types of weight maps in our experiment. The average weight map means N homographies play the same role to each block and the single weight map represents that one block only corresponds to one homography. From Table 3, it can be observed that Gaussian weight map has the best metircs in our test dataset. Holding the assumption of smoothness in camera motion, Gaussian weight can handle distortions more smoothly and the other two weights tend to handle linear variations more often. As shown in Fig.8, correction with average weight map still has distortions and correction with single weight map even has discontinuous parts. Correction with designed Gaussian weight map is more effective than the other types. We select Gaussian weight in other experiments.

Block Quantities Block quantity is related to the number of homographies that need to predict. We conduct experiments on block quantities of 4, 8, 12. Theoretically, more blocks can lead to higher prediction accuracy. However, the increase of the complexity of network prediction will reduce the prediction accuracy. As shown in Table 3, 8 blocks has more precise prediction results than the others, for lower EPE, higher PSNR and SSIM. One example of occlusion is shown in Fig. 9. It is obvious that correction with 8 blocks has the least distortion for the building covered by branches and results with block= 4 and 12 are visually less optimal. We select 8 blocks in all other experiments.

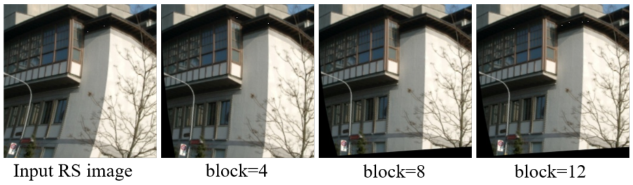

Figure 9. Comparison among different block quantities. Top right: the input RS image. The others are the correction results with block quantities of 4,8,12, respectively.

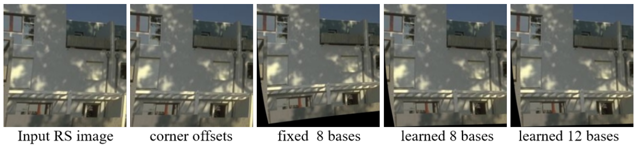

Figure 10. Comparison among different methods of regression. Column 1 is a single RS image, columns 2∼5 are regression with corner offsets in each block, coefficients of fixed bases, coefficients of learned 8 bases and 12 bases, respectively.

Offsets vs. Homography Flow Bases In order to find the most efficient motion bases, we compare the results from previous corner offsets estimation [4, 35], coefficients of fixed 8-bases [34], learned 8-bases and learned 12-bases. In [34], it is justified that bases coefficient prediction is better than corner offsets prediction. The fixed bases are generated by modifying each element of an identity matrix one at a time, which produce bases that may not equally important during the prediction, thus less effective. Because the real camera motions only span a subspace of the entire solution space. In contrast, learning bases in real data with PCA can effectively overcome these problems by extracting the most important bases corresponding to the real camera motion. From Table. 3, the offset prediction method has the worst metrics. Prediction with bases is significantly better as shown. Moreover, correction with learned 8-bases is quantitatively better than correction with fixed 8-bases. Noted that results of learned 8-bases and 12-bases are similar, since the energy of the principal component has reached 99.9% while $N = 8$ in motion bases learning (in the supplementary material). Fig. 10 displays an RS example in low light. Result with offset regression is quite similar to the input RS image so that it fails in correction. The results of learned bases are similar to each other and both remove the distortions much better than the rectified image from fixed 8-bases, which is consistent with our quantitative results.

5.4. Results on Images with Varying Depth

As mentioned in Sec.3.1, the HM model holds when depth of the scene is on a plane or at infinity. The blending of homography flows is actually smooth interpolated to avoid discontinuities across scanline blocks. Here, we study how strict this requirement is by exploring some real examples. In the experiments, under different depth varia-

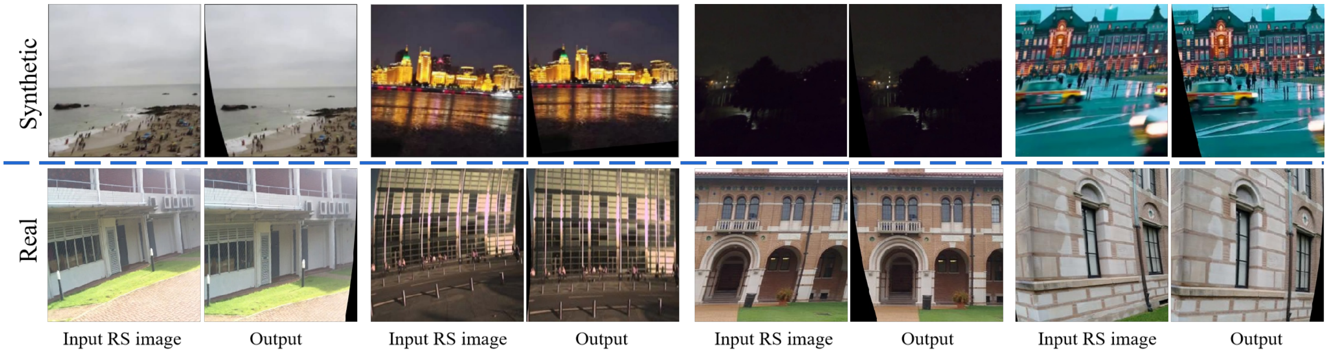

Figure 11. Our rectification results on synthetic data and real data. Noted that examples in row 1 are all images of adverse scenarios. Row 2 are all real RS examples to verify our method effectiveness. More results can be seen in the supplementary.

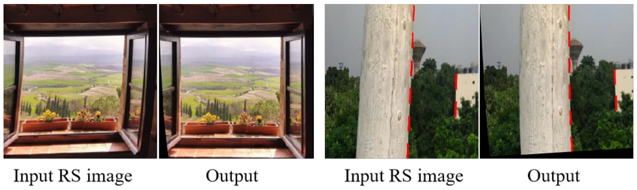

Figure 12. Scenes consisted of several planes with varying depth. Left: depth variation with camera rotation. Right: depth variation with camera translation.

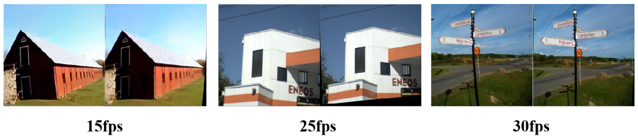

Figure 13. Correction on real RS images with different framerates. In each pair, the left is RS image and the right is corrected result, which shows the generalizability of the learned-bases.

tion cases, our method can still work. Fig. 12 shows two real examples, where the left has discontinuous depth variation with camera rotation, while the right has discontinuous depth variation with domination camera translation. Our method is not designed for such settings but can work in several cases in experiments.

5.5. Results on Images with Different Framerates

Our correction method and the learned bases are effective with respect to different timescales due to the following reasons, 1) frame pairs are selected from videos with random skip, creating different rates; 2) our RS-Homo dataset is further synthesized using the same parameters as [37, 10], which simulates different timescales, and 3) we learn from the natural video data to obtain effective global homography flow bases that own good generalizability. We show three captured examples with 15fps, 25fps and 30fps in Fig.13, with house, sky or vertical bars. It is shown that our method can correct them successfully with the same learned basis.

5.6. Results on Synthetic and Real Images

To demonstrate the effectiveness of our method, more examples are shown in Fig. 11. Row 1 displays the RS image correction examples on the synthetic data from RS-Homo dataset. Column 1∼2 is an example of sky and sea with poor texture (little can be found on the beach), column 3∼6 are examples with low light (column 3 shows buildings and river at night, column 5 depicts light in a park at night), and column 7∼8 is an example with large foreground (cars and pedestrians in the foreground). However, when we implement method [6], it always fails with these types of images due to lack of rich or stable image features. Row 2 shows correction on real RS images with our method. Different types of real RS images are successfully corrected by removing the distortions. Our method is trained on synthetic data, but can work on real data.

6. Conclusion

We have presented the deep homography mixture model for the task of single RS correction. We learn the correction motion from a synthesized dataset RS-Homo consisting of GS and RS pairs with adverse cases. We learn coefficients that combine homo-bases learned from natural videos. We show that this pipeline can work on a typical architecture such as VGG. Experiments show that our method outperforms existing single RS image correction methods both quantitatively and qualitatively, both on the synthesized and real images. We also compare with multi-frame methods in their proposed datasets. Our ablation studies show the effectiveness of each component in our method.

Acknowledgements This work was supported in part by Sichuan Science and Technology Program of China under grants No.2023NSFSC0462, No.2022YFQ0079, No.2022YFG0011, No.2023NSFSC1972 and in part by National Natural Science Foundation of China (NSFC) under grant No.62031009.

References

[1] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla. R6p-rolling shutter absolute camera pose. In *Proc. CVPR*, pages 2292–2300, 2015. 1

[2] Simon Baker, Eric Bennett, Sing Bing Kang, and Richard Szeliski. Removing rolling shutter wobble. In *Proc. CVPR*, pages 2392–2399, 2010. 1, 2

[3] Won-ho Cho and Ki-Sang Hong. Affine motion based cmos distortion analysis and cmos digital image stabilization. *IEEE Trans. on Consumer Electronics*, 53(3):833–841, 2007. 1

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 2, 3, 7

[5] Bin Fan, Yuchao Dai, and Mingyi He. Sunet: symmetric undistortion network for rolling shutter correction. In *Proc. CVPR*, pages 4541–4550, 2021. 2, 5, 6

[6] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *Proc. ICCP*, pages 1–8, 2012. 1, 2, 5, 6, 8

[7] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2, 3

[8] Soren Hauberg, Aasa Feragen, and Michael J Black. Grassmann averages for scalable robust pca. In *Proc. CVPR*, pages 3810–3817, 2014. 2, 5

[9] Johan Hedborg, Per-Erik Forssén, Michael Felsberg, and Erik Ringaby. Rolling shutter bundle adjustment. In *Proc. CVPR*, pages 1434–1441, 2012. 1

[10] Praveen Kandula, T Lokesh Kumar, and AN Rajagopalan. Deep end-to-end rolling shutter rectification. *JOSA A*, 37(10):1574–1582, 2020. 1, 5, 6, 8

[11] Alexandre Karpenko, David Jacobs, Jongmin Baek, and Marc Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR*, 1(2):13, 2011. 2

[12] Yizhen Lao and Omar Ait-Aider. A robust method for strong rolling shutter effects correction using lines with automatic feature selection. In *Proc. CVPR*, pages 4795–4803, 2018. 1, 2

[13] Chia-Kai Liang, Li-Wen Chang, and Homer H Chen. Analysis and compensation of rolling shutter effect. *IEEE Trans. on Image Processing*, 17(8):1323–1330, 2008. 1, 2

[14] Guangcan Lin, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2012. 4

[15] Peidong Liu, Zhaopeng Cui, Viktor Larsson, and Marc Pollefeys. Deep shutter unrolling network. In *Proc. CVPR*, pages 5941–5949, 2020. 1, 2, 5, 6

[16] Jiawei Mo, Md Jahidul Islam, and Junaed Sattar. Imu-assisted learning of single-view rolling shutter correction. In *Conference on Robot Learning*, pages 861–870. PMLR, 2022. 3

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 5

[18] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 5, 6

[19] Pulak Purkait and Christopher Zach. Minimal solvers for monocular rolling shutter compensation under ackermann motion. In *Proc. WACV*, pages 903–911, 2018. 1

[20] Pulak Purkait, Christopher Zach, and Ales Leonardis. Rolling shutter correction in manhattan world. In *Proc. ICCV*, pages 882–890, 2017. 1, 2, 5, 6

[21] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *Proc. CVPR*, pages 2291–2299, 2017. 1, 2, 5, 6

[22] Vijay Rengarajan, Ambasamudram N Rajagopalan, and Rangarajan Aravind. From bows to arrows: Rolling shutter rectification of urban scenes. In *Proc. CVPR*, pages 2773–2781, 2016. 1, 2, 5, 6

[23] 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

[24] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proc. ECCV*, pages 430–443, 2006. 2

[25] Olivier Saurer, Kevin Koser, Jean-Yves Bouguet, and Marc Pollefeys. Rolling shutter stereo. In *Proc. ICCV*, pages 465–472, 2013. 1

[26] Olivier Saurer, Marc Pollefeys, and Gim Hee Lee. Sparse to dense 3d reconstruction from rolling shutter images. In *Proc. CVPR*, pages 3337–3345, 2016. 1

[27] H. Shao, T. Svoboda, and L. Van Gool. Zubud-zurich buildings database for image based recognition. *Tech. Rep(Swiss Federal Institute of Technology*, 260:6–8, 2003. 5, 6

[28] Jianbo Shi et al. Good features to track. In *Proc. CVPR*, pages 593–600, 1994. 2

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *CoRR*, page abs/1409.1556, 2014. 3

[30] Chengzhou Tang, Lu Yuan, , and Ping Tan. Lsm: Learning subspace minimization for low-level vision. In *Proc. CVPR*, pages 6235–6246, 2020. 4

[31] Subeesh Vasu, AN Rajagopalan, et al. Occlusion-aware rolling shutter rectification of 3d scenes. In *Proc. CVPR*, pages 636–645, 2018. 1, 2

[32] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proc. CVPR*, pages 120–130, 2015. 2, 4

[33] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, pages 3485–3492, 2010. 2, 5, 6

[34] Nianjin Ye, Chuan Wang, Haoqiang Fan, and Shuaicheng Liu. Motion basis learning for unsupervised deep homography estimation with subspace projection. In *Proc. ICCV*, pages 13117–13125, October 2021. 2, 3, 4, 7

[35] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *Proc. ECCV*, 2020. 2, 3, 4, 5, 6, 7

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

[37] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Rolling-shutter-aware differential sfm and image rectification. In *Proc. ICCV*, pages 948–956, 2017. 1, 2, 8

[38] Bingbing Zhuang, Quoc-Huy Tran, Pan Ji, Loong-Fah Cheong, and Manmohan Chandraker. Learning structure-and-motion-aware rolling shutter correction. In *Proc. CVPR*, pages 4551–4560, 2019. 1, 2, 5, 6