Rethinking Video Rain Streak Removal: A New Synthesis Model and A Deraining Network with Video Rain Prior

Shuai Wang[1], Lei Zhu[2,3] ⋆, Huazhu Fu[4], Jing Qin[5], Carola-Bibiane Schönlieb[6], Wei Feng[7], and Song Wang[8]

[1] Tianjin University
[2] The Hong Kong University of Science and Technology (Guangzhou)
[3] The Hong Kong University of Science and Technology
[4] IHPC, ASTAR
[5] The Hong Kong Polytechnic University
[6] University of Cambridge
[7] School of Computer Science and Technology, Tianjin University
[8] University of South Carolina

Abstract. Existing video synthetic models and deraining methods are mostly built on a simplified video rain model assuming that rain streak layers of different video frames are uncorrelated, thereby producing degraded performance on real-world rainy videos. To address this problem, we devise a new video rain synthesis model with the concept of rain streak motions to enforce a consistency of rain layers between video frames, thereby generating more realistic rainy video data for network training, and then develop a recurrent disentangled deraining network (RDD-Net) based on our video rain model for boosting video deraining. More specifically, taking adjacent frames of a key frame as the input, our RDD-Net recurrently aggregates each adjacent frame and the key frame by a fusion module, and then devise a disentangle model to decouple the fused features by predicting not only a clean background layer and a rain layer, but also a rain streak motion layer. After that, we develop three attentive recovery modules to combine the decoupled features from different adjacent frames for predicting the final derained result of the key frame. Experiments on three widely-used benchmark datasets and a collected dataset, as well as real-world rainy videos show that our RDD-Net quantitatively and qualitatively outperforms state-of-the-art deraining methods. Our code, our dataset, and our results on four datasets are released at `https://github.com/wangshauitj/RDD-Net`.

Keywords: Video deraining, new video deraining model, video rain direction prior, disentangled feature learning

1 Introduction

As the most common type of rain degradation, rain streaks often cause the visibility degradation in captured rainy images or videos, and thus lead to failure of outdoor

⋆ Lei Zhu (leizhu@ust.hk) is the corresponding author of this work.

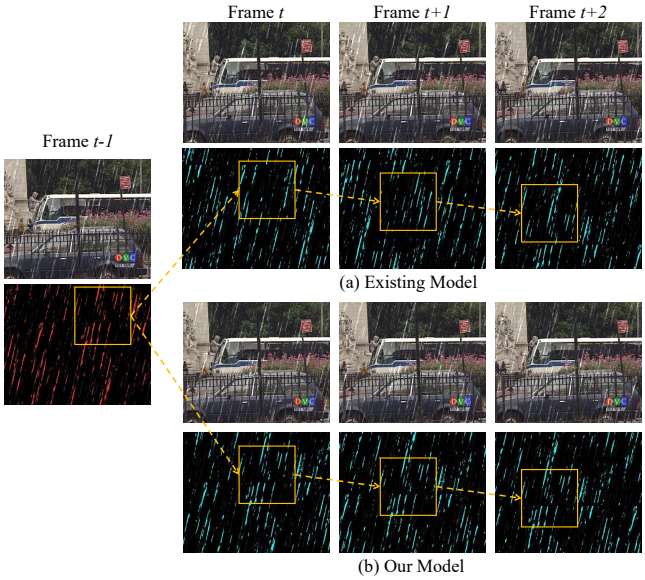

Fig. 1: Synthesized rain streak images and rainy video frames using (a) the existing video rain synthesis model [36] and (b) our proposed video rain synthesis model with rain streak motion. Apparently, the rain streaks at the yellow rectangle move along the dominated rain direction of the video in our method, while these synthesized rain streaks in existing models are not correlated.

computer vision systems, which generally take clean video frames as input by default. Rain streaks also lower the subsequent video analysis since they partially occlude a background scene, change image appearance, make the scene blurred, etc. A number of methods have been developed to remove rain streaks in past decades. They are typically classified by their input type. Single-image methods [21, 44, 7, 37, 20, 41, 26] remove rain streaks given only a single image by examining image priors of the underlying background scene and rain streaks, while video-based methods [27, 22, 36, 38] leverage rich temporal information across video frames to locate and remove rain streaks.

Traditionally, most video deraining methods attempt to generalize the single-image rain models to videos [23, 22, 38], and, in principle, the formulation is:

$$I_t = B_t + S_t, \text{ where } \quad t \in [1, T], \tag{1}$$

where I_t is the t-th frame of the video with rain streaks, B_t is the corresponding rain-free background layer, while S_t is the rain streak layer. T is the total number of frames. As shown in Fig. 1(a), these methods often assume that rain streaks at adjacent video frames are independent and identically distributed random samples. Recently, several researchers further modified this model by considering other rain degradation factors (*e.g.*, fog) [36], or rain occlusions [23, 22]. However, *without considering the temporal coherence among frames, in these models, the rain streak layers of neighboring frames are discontinuous and messy*. To the end, the rainy videos synthesized based on these models are not realistic as real rainy videos, such that the video deraining models trained on these synthesized videos cannot achieve satisfactory results on real rainy videos.

In this paper, we rethink the video rain streak removal problem, based on a video rain observation (prior) that rain streaks usually fall within a limited region along the directions in dynamic video scenes, which indicates that rain streaks often moves along several directions. Thus, we introduce a concept of "**rain streak motion**" to model such a video rain prior. Specifically, we devise a new video rain synthesis model embedding the motions of rain streaks, as shown in Fig. 1(b). Simultaneously, we also develop a novel recurrent disentangled deraining network (RDD-Net) to recurrently estimate additional rain motion from adjacent video frames for boosting video deraining. In our RDD-Net, we first develop a disentangling temporal (DT) module to decouple temporal features of adjacent video frames to sequentially predict a rain streak layer, a rain motion layer, and a rain-free background layer according to our video rain synthesis model. Then, we develop attentive recovery (AR) modules to integrate multiple output layers for generating our final result of video deraining. Overall, the major contributions of this work are:

- First, in this paper, we introduce a new prior of "**rain streak motion**", which models the rain streak motion in video. Based on this term, we devise a new video rain synthesis model to generate a more realistic rainy video dataset for network training.
- Second, we devise a novel recurrent disentangled deraining network (RDD-Net) by attentively aggregating predictions from temporal features of adjacent video frames.
- Third, a disentangled temporal (DT) module is introduced to disentangle temporal features from each pair of adjacent video frames into several features for predicting rain streak layers, rain motion layers, and clean background layers in sequence based on our video rain synthesis model. Simultaneously, an attentive recovery (AR) module is utilized to integrate three predictions of multiple DT modules for fully exploiting complementary information among these predictions, and generate the final video deraining result.
- Final, we evaluate the proposed method on real-world rainy videos and four synthesized video datasets (three widely-used benchmarks and a new synthesized video dataset) by comparing it with state-of-the-art deraining methods. The experimental results show that the proposed method outperforms all competitors on all benchmarks and real hazy images. Overall, Our method sets a new state-of-the-art baseline on video deraining.

2 Related Work

2.1 Single-image rain streak removal

Early single-image deraining methods examined diverse image priors based on the statistics of the rainy and clean images for removing rain streaks of the single input rainy image. [16, 29, 12, 25, 6, 21, 44]. Inspired by the observation that rain streaks usually fall within a narrow band of directions, Zhu et al. [44] developed rain direction prior, sparse prior, and rain patch prior to form a joint optimization for image deraining. These methods suffers from failures in handling complex rainy cases [41, 20], since their hand-crafted image priors are not always correct [44].

Recent methods [43] mainly focused on designing different convolutional neural networks (CNNs) to address image deraining from collected data. Fu et al. [7] learned a mapping between rainy and clean detail layers, and then add a predicted detail layer into a low-pass filtered base layer for outputting a derained image. Then, Yang et al. [37] presented a multi-task deep learning architecture to jointly detect and remove rain streaks from CNN features of a contextualized dilated network. Later CNNs utilized a residual learning to learn a rain streak image for image deraining [18]. More recently, Zhang et al. [41] classified the rain density of the input rainy image and incorporated the density information to multiple densely-connected blocks for predicting a rain streak image. Ren et al. [26] combined ResNet, recurrent layers, and a multi-stage recursion for building a deraining network. Jiang et al. [13] formulated a multi-scale progressive fusion network (MSPFN) to unify the input image scales and hierarchical deep features for image deraining. Although we can generalize image deraining methods to remove rain streaks of a video in a frame by frame manner, the temporal information among video frames enables video deraining methods to work better than image deraining ones [19, 22, 36, 38].

2.2 Video rain streak removal

Garg and Nayar first modeled the video rain and addressed video rain streak removal [10, 9]. Many subsequent methods [42, 24, 3, 2, 28, 1, 6, 4, 30, 31, 17, 27] examined more hand-crafted intrinsic priors of rain streaks and clean background details for video deraining. Wei et al. [34] encoded the rain streaks as a patch-based mixture of Gaussians to make the developed method to better adapt a wider range of rain variations. Please see [23] for reviewing video deraining methods with diverse hand-crafted priors.

Recently, deep neural networks [5, 23, 38] have also been employed to handle video deraining. Li et al. [19] formulated a multiscale convolutional sparse coding to decompose the rain layer into different levels of rain streaks with physical meanings for video deraining. Yang et al. [36] constructed a two-stage recurrent network with dual-level flow regularizations for video deraining. Recently, Yan et al. [35] developed a self-alignment network with transmission-depth consistency to solve the problem of rain accumulation. To address the gap between synthetic and real dataset, Yue et al. [39] presented a semi-supervised video deraining method with a dynamical rain generator. Although improving overall visibility of input rainy videos, existing CNN-based video deraining methods often randomly synthesized rain streak layers of neighboring video frames without considering motions of rain streaks. *Hence, these synthesized videos do not have realistic rain streaks, thereby degrading deraining performance when training video deraining network on these synthesized data.* To alleviate this issue, we devise a new video rain model with rain streak motions to more accurately model rain streak layers of videos and develop a video deraining network based on the new rain model for enhancing video deraining performance.

3 Video Rain Synthesis Model

Observing a video rain prior that rain streaks often fall within a limited range along the directions in real rainy, even for heavy rain, we introduce a new concept of "**rain**

streak motion" to model this phenomenon. With the concept of rain streak motion, we can figure out two kinds of rain streaks contributing to the rain streak layer in the rain video. *First*, the new rain streaks will appear when the video camera moves to capture a dynamic scene. *Second* and more importantly, some rain streaks in the $(t-1)$-th frame moves along the dominated rain direction into the t-th video frame to form its rain streaks. Thus, we propose a new video rain synthesis model by embedding these two kinds of rain streaks:

$$I_t = \begin{cases} B_1 + S_1 \ , & t = 1, \\ B_t + (S_{t-1} \bigoplus M_{(t-1)\to t}) + R_t \ , & t \in [2, T], \end{cases} \tag{2}$$

where S_{t-1} and S_t denote the rain streak layers in $(t-1)$-th and t-th video frames, respectively. $M_{(t-1)\to t}$ is the rain streak motion from the $(t-1)$-th frame to the t-th frame, which represents the dominated rain streak directions of a rainy video. It is computed by multiplying a random integer with the rain streak angle of the first rain streak map S_1. R_t denotes the new rain streaks, which appear due to the camera movement when taking the video, $\bigoplus$ represents the point-wise addition between S_{t-1} and $M_{(t-1)\to t}$.

Fig. 1 shows the synthesized rain streak images and the corresponding rainy video frames by using the traditional method (Fig. 1(a)) and our model (Fig. 1(b)), respectively. It is clearly observed that the rain streaks in adjacent frames generated from the traditional model are uncorrelated and, somehow, messy, while our rain model generates more consistent rain streaks along a dominated rain direction, as shown in the yellow rectangles of Fig. 1(b), which more faithfully reflects real rainy scenes. In this regard, the networks trained on our synthesized rain videos have potential to achieve better deraining performance on real rain videos than those trained on previous synthesized training datasets.

Difference of "rain motion in Garys model. Gary et al. [10] presented a raindrop oscillation model to render a complex falling raindrop (i.e., rain motion among pixels of an image) produce for generating a realistic rain streak image, or generating rain streak maps for all video frames independently, which totally ignores the rain streak movement between video frames. Unlike this, our rain video model at Eqn. (2) takes the rain streak map S (S can be generated by Garys method) of the first video frame as the input, and then follows the rain streak direction (i.e., rain motion) of the input rain streak map to generate the rain streak maps of all other video frames. Hence, the usage of the rain motion in our video rain model and Garys model [10] are different. More importantly, our video rain model equation enables us to formulate a CNN to decompose the rain motion, rain streak, and non-rain background and our network outperforms state-of-the-art methods in four synthesized datases and real data.

4 Proposed Deraining Network

Fig. 2 shows the illustration of the proposed recurrent disentangled video deraining network (RDD-Net). The intuition behind our RDD-Net is to recurrently disentangle temporal features from adjacent frames for predicting a background layer, a rain streak

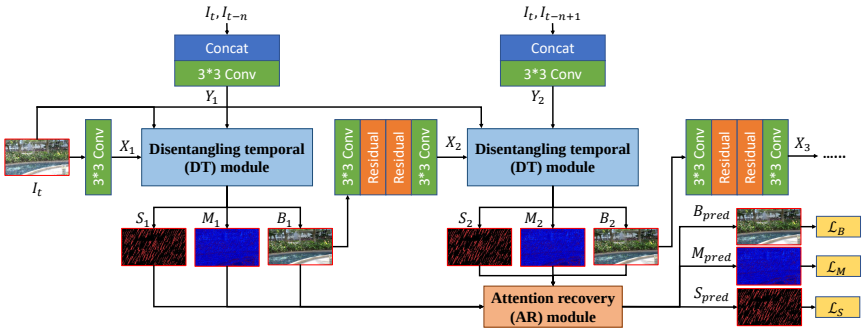

Fig. 2: Schematic illustration of the proposed recurrent disentangled video deraining network (RDD-Net). S_{pred}, M_{pred}, and B_{pred} denote predictions of the rain layer, the rain motion layer, and the clean background layer for the input target video frame I_t. $\{I_{t-n}, \ldots, I_{t-1}, I_{t+1}, \ldots, I_{t+n}\}$ are $2n$ adjacent video frames of I_t.

layer, and a rain motion layer, and then attentively integrate these predictions from multiple adjacent frames to produce the derained results.

RDD-Net takes $2n+1$ video frames $\{I_{t-n}, \ldots, I_{t-1}, I_t, I_{t+1}, \ldots, I_{t+n}\}$ as the inputs and predicts a derained result of the target frame I_t. To leverage video temporal information, our RDD-Net starts by grouping the input $2n+1$ video frames into $2n$ image pairs, i.e., $\{(I_t, I_{t-n}), \ldots, (I_t, I_{t-1}), (I_t, I_{t+1}), \ldots, (I_t, I_{t+n})\}$. Then, for the first image pair (I_t, I_{t-n}), we apply a 3×3 convolution on $[I_t, I_{t-n}]$ (i.e., concatenation of I_t and I_{t-n}) to generate a feature map (denoted as "Y_1"), while apply a 3×3 convolution on the target video frame I_t to obtain features X_1. To further extract the rain motion of (I_t, I_{t-n}), we devise a disentangled temporal (DT) module and pass X_1 and Y_1 into the DT module to predict a background image B_1, a rain streak image S_1, and a rain motion image M_1, according to our video rain model defined in Eq. 2.

After that, we utilize a 3×3 convolution, two residual blocks, and a 3×3 convolution on B_1 to produce features X_2, and a 3×3 convolution is applied on the subsequent image pair (I_t, I_{t-n-1}) to obtain features Y_2. Meanwhile, the second DT module is utilized to compute another three image B_2, S_2, and M_2 from X_2 and Y_2. By repeating this process, we can use DT modules to sequentially predict the background image, the rain streak image, and the rain motion image for all $2n$ image pairs. Finally, we develop an attention recovery (AR) module to predict a final background image (see P_B of Fig. 2) from $\{B_1, \ldots, B_{2n}\}$, an AR module to predict a final rain-free image P_S from $\{S_1, \ldots, S_{2n}\}$, and an AR module to predict a rain motion image P_M from $\{M_1, \ldots, M_{2n}\}$ for the target video frame I_t,

Loss function. Unlike existing video deraining methods only predicting the rain and background layers, our RDD-Net predicts an additional rain motion map for each target

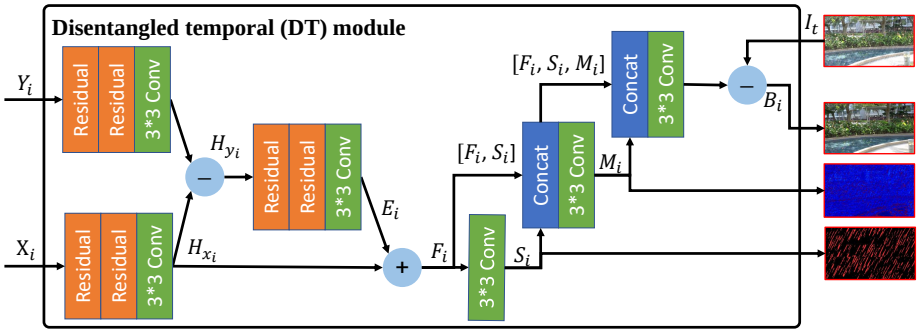

Fig. 3: Schematic illustration of the DT module of our RDD-Net.

frame. Hence, the total loss $\mathcal{L}_{total}$ of our RDD-Net is:

$$\mathcal{L}_{total} = \mathcal{L}_S + \mathcal{L}_M + \mathcal{L}_B \ ,$$
$$where\ \mathcal{L}_S = \|S_{pred} - S_{gt}\|_1 \ , \mathcal{L}_M = \|M_{pred} - M_{gt}\|_1 \ , \mathcal{L}_B = \|B_{pred} - B_{gt}\|_1 \ . \tag{3}$$

Here, S_{pred} and S_{gt} denote the predicted rain layer and the corresponding ground truth. M_{pred} and M_{gt} are the predicted rain motion layer and the corresponding ground truth. B_{pred} and B_{gt} denote the predicted background layer and the corresponding ground truth. $\mathcal{L}_S$ is the prediction loss of S_{pred} and S_{gt}, and we use L_1 loss to compute $\mathcal{L}_S$. $\mathcal{L}_M$ is the L_1 loss of M_{pred} and M_{gt}, while $\mathcal{L}_B$ is the L_1 loss of B_{pred} and B_{gt}.

4.1 Disentangling Temporal (DT) Module

Existing video deraining networks [36, 38] predict a rain layer and a background layer by capturing the temporal correlations of adjacent video frames. As presented in our video rain model (see Eq. 2), rain motions enable us to more accurately approximate the underlying rain streak distributions over rainy videos. In this regard, we develop a disentangled temporal (DT) module to learn temporal features from adjacent video frames and decouple these temporal features to sequentially compute a rain layer, a clean background layer, and a rain motion layer since they are intrinsically correlated.

Fig. 3 shows the schematic illustration of the disentangled temporal (DT) module, which takes features X_i of the target video frame, another features (Y_i) from two adjacent video frames, and the target video frame I_t as the inputs. To learn the temporal features, our DT module starts by fusing features of adjacent video frames. We first apply two residual blocks and a 3×3 convolution on X_i to obtain H_{x_i}, and two residual blocks and a 3×3 convolution are utilized on Y_i to obtain H_{y_i}. After that, we subtract H_{y_i} from H_{x_i} and apply two residual blocks and a 3×3 convolution on the subtraction result to obtain features E_i. We then add E_i and H_{x_i} to obtain the temporal feature map (denoted as F_s) of adjacent video frames. Mathematically, the temporal feature map F_s is computed by

$$F_s = H_{x_i} + \mathcal{D}_1(H_{x_i} - H_{y_i}) \ , where\ H_{x_i} = \mathcal{D}_2(X_i),\ and\ H_{y_i} = \mathcal{D}_3(Y_i). \tag{4}$$

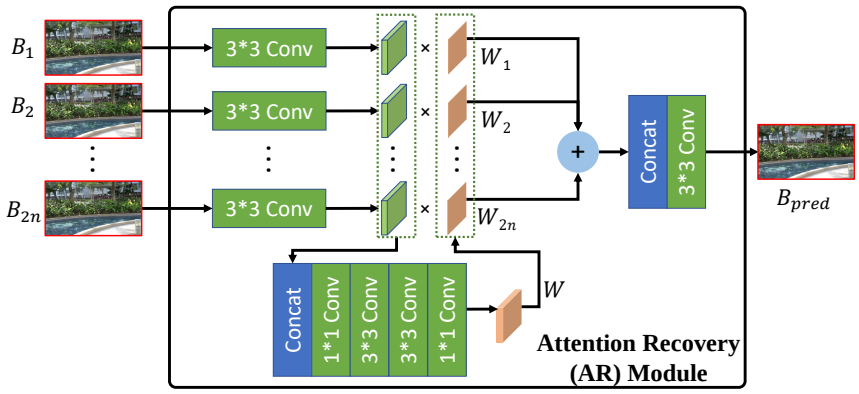

Fig. 4: Schematic illustration of the attentive recovery (AR) module of our RDD-Net.

Here, $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ denote three blocks, which has two residual blocks and a 3×3 convolution. And $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ do not share the same convolutional parameters.

Once obtaining the temporal feature map F_s (see Eq. 4), our DT module predicts the clean background layer, the rain layer, and the rain motion layer one by one. Specifically, F_i is passed to a 3×3 convolutional layer to predict a rain layer S_i. Then, we concatenate the obtained rain layer S_i with the temporal feature map F_i and utilize a 3×3 convolutional layer on the concatenation result (see $[F_i, S_i]$ of Fig. 3) to compute a rain motion layer M_i. Finally, we subtract a result of a 3×3 convolutional layer on the concatenation (see $[F_i, S_i, M_i]$ of Fig. 3) of F_i, S_i, and M_i from the target video frame I_t to get a clean background layer B_i. In summary, the rain layer S_i, the rain motion layer M_i, and the clean background layer B_i are computed as:

$$S_i = conv(F_i)\ , M_i = conv([F_i, S_i])\ , B_i = I_t - conv([F_i, S_i, M_i])\ , \tag{5}$$

where $conv$ denotes a 3×3 convolutional layer, and three $conv$ operations for computing S_i, M_i, and B_i do not share convolutional parameters.

4.2 Attention Recovery (AR) Module

Rather than stacking or warping adjacent video frames together in most video restoration networks, our method follows a back-projection framework [11] to treat each adjacent frame as a separate source of temporal information, and then recurrently combine multiple sources from several adjacent video frames, as shown in Fig. 2. Unlike simply concatenating predictions at each recurrent step in original back-projection framework, we find that there are complemental information among these prediction results at different recurrent steps. Hence, we develop an attention recovery (AR) module to attentively aggregate these different predictions for further improving the network prediction accuracy. Note that our DT module at each recurrent step has three predictions, including a rain layer prediction, a rain motion layer prediction, and a clean background layer prediction; see Fig. 3. In this regard, we develop an AR module to aggregate $2n$ background predictions (i.e., $\{B_1, B_2, \ldots, B_{2n}\}$) to produce a final result of estimating the

Table 1: Quantitative comparisons of our network and compared methods on three widely-used video rain streak removal benchmark datasets. Best results are denoted in red and the second best results are denoted in blue.

		CVPR'17	TIP'15	CVPR'17	CVPR'18	ICCV'15	ICCV'17	TIP'18	CVPR'18	CVPR'18	CVPR'19	CVPR'20	CVPR'21	
Dataset	Metric	DetailNet [8]	TCLRM [17]	JORDER [37]	MS-CSC [19]	DSC [25]	SE [34]	FastDerain [15]	J4RNet [23]	SpaCNN [5]	FCDN [36]	SLDNet [38]	S2VD [39]	RDD-Net
RainSynLight25	PSNR↑	25.72	28.77	30.37	25.58	25.63	26.56	29.42	32.96	32.78	35.80	34.28	34.66	38.61
	SSIM↑	0.8572	0.8693	0.9235	0.8089	0.9328	0.8006	0.8683	0.9434	0.9239	0.9622	0.9586	0.9403	0.9766
RainSynHeavy25	PSNR↑	16.50	17.31	20.20	16.96	17.33	16.76	19.25	24.13	21.21	27.72	26.51	27.03	32.39
	SSIM↑	0.5441	0.4956	0.6335	0.5049	0.5036	0.5293	0.5385	0.7163	0.5854	0.8239	0.7966	0.8255	0.9318
NTURain	PSNR↑	30.13	29.98	32.61	27.31	29.20	25.73	30.32	32.14	33.11	36.05	34.89	37.37	37.71
	SSIM↑	0.9220	0.9199	0.9482	0.7870	0.9137	0.7614	0.9262	0.9480	0.9474	0.9676	0.9540	0.9683	0.9720

Table 2: Quantitative comparisons of our network and compared methods on our synthesized rainy video dataset. Best results are denoted in red and the second best results are denoted in blue.

	CVPR'16	CVPR'17	CVPR'17	CVPR'19	CVPR'20	ACM MM'20	CVPR'21	CVPR'17	TIP'18	CVPR'18	CVPR'19	CVPR'20	CVPR'21	
Metric	LP [21]	JORDER [37]	DetailNet [8]	PReNet [26]	MSPFN [13]	DCSFN [32]	MPRNet [40]	DIP [14]	FastDerain [15]	MS-CSC [19]	FCDN [36]	SLDNet [38]	S2VD [39]	RDD-Net
PSNR↑	19.42	15.94	21.42	27.06	22.99	26.77	28.42	19.35	23.66	17.36	24.81	20.31	24.09	31.82
SSIM↑	0.6841	0.5334	0.7826	0.9077	0.8325	0.9052	0.9203	0.6518	0.7893	0.5968	0.8658	0.6272	0.7944	0.9423

background layer of the target video frame I_t. Meanwhile, we develop an AR module to aggregate $2n$ rain layer predictions (i.e., $\{S_1, S_2, \ldots, S_{2n}\}$) to produce a final result of estimating the rain layer of I_t, while another AR module is devised to aggregate $2n$ rain motion layer predictions (i.e., $\{M_1, M_2, \ldots, M_{2n}\}$) to produce a final result of estimating the rain motion layer of I_t.

Here, we only show the schematic illustration of the developed AR module of computing a final background layer; see Fig. 4. Specifically, taking $2n$ predictions of the clean background layer as the inputs, our AR module first utilize a 3×3 convolutional layer on each background layer prediction to obtain $2n$ feature maps, which are denoted as $\{Q_1, Q_2, \ldots, Q_{2n}\}$. Then, we concatenate these $2n$ feature maps and utilize four convolutional layers and a softmax layer on the concatenated feature map to produce an attention map W with $2n$ channels. The four convolutional layers includes a 1×1 convolution, two 3×3 convolutions, and a 1×1 convolution. After that, we multiply all $2n$ channels of W with $2n$ feature maps to produce weighted feature maps, which are then added together to produce the final background layer prediction (see B_{pred} of Fig. 4) by using a 3×3 convolutional layer. Hence, B_{pred} is computed by:

$$B_{pred} = conv(cat(W_1Q_1, W_2Q_2, \ldots, W_{2n}Q_{2n})) \,, \tag{6}$$

where $conv$ denotes a 3×3 convolution. $cat(\cdot)$ is a feature concatenation operation.

5 Experiments

Benchmark datasets. We evaluate the effectiveness of our network on three widely-used benchmark datasets and a new dataset (denoted as "RainMotion") synthesized in our work. Table 3 summarizes the details of four video deraining datasets. Three benchmark datasets are RainSynLight25 [23] with 215 light rain videos, RainSynComplex25 [23] with 215 heavy rain videos, and NTURain [5] with 24 rain videos. Regarding our dataset, we use the same 16 clean background videos (8 videos for training and 8 videos for testing) of NTURain [5] to generate 80 rain videos based on our video rain model (see Eq. 2). Specifically, for each clean background videos with k frames, we

Table 3: Comparison between different datasets.

Dataset	Split	Video Num	Video Length	Video Frame Num
RainSynLight25	train	190	9	1710
	test	25	31	775
RainSynHeavy25	train	190	9	1720
	test	25	31	755
NTURain	train	24	80-138	-
	test	8	116-298	-
Our RainMotion	train	40	50	2000
	test	40	20	800

Fig. 5: Visual comparison of different deraining methods on a real rain video sequence. The blue box indicates the comparison of rain streak removal. The red box indicates the comparison of detail retention.

first generate five large rain streak masks (with 10 times spatial resolutions of the clean video frame), including three masks obtained by using Photoshop, a mask randomly selected from RainSynComplex25 [23], and a mask from [10], and the spatial resolution of each rain streak mask is than that of the clean video frame. Then, k images are cropped from each rain streak mask along the rain direction of the mask to simulate the rain with video rain motion, and then we add these k images with the clean background layer to generate a rainy video. By doing so, we can obtain five rainy videos for each background video, thereby resulting in 40 rainy videos in our training set, and 40 rainy videos in our testing set.

Implementation details. We implement our RDD-Net with PyTorch, and use the Adam optimizer to train the network on a NVIDIA GTX 2080Ti. We empirically set n=3, which means that our network receives seven frames as the inputs for video deraining; see our network in Fig. 2. We crop the target video frame to 128×128. The initial learning rate, weight decay, and batch size are empirically set as 0.0001, 0.00005 and 8, respectively. The total epoch number is empirically set as 1500 for RainSynLight25, 1500 for RainSynComplex25, 150 for NTURain, and 500 for RainMotion. Our RDD-Net contains 30.64MB parameters, and the average running time of our network is about 0.8633s for one video frame with a resolution of 832×512. We employed peak signal to noise ratio (PSNR) and structural similarity index (SSIM) [33] to quantitatively compare different methods.

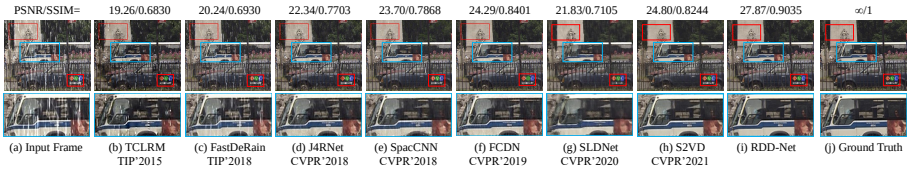

Fig. 6: Visual comparison of different deraining methods on RainSynHeavy25 dataset. The blue box indicates the comparison of rain streak removal. The red box indicates the comparison of detail retention.

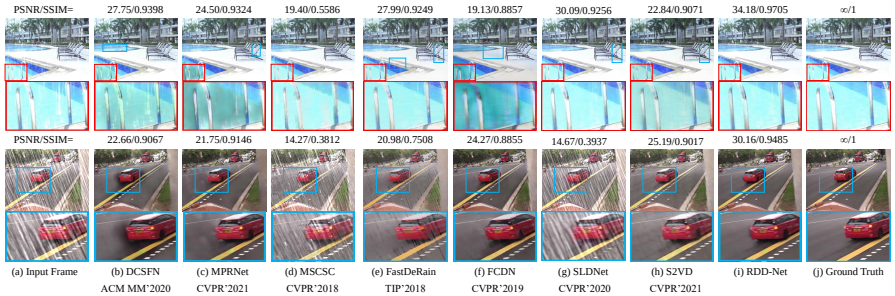

Fig. 7: Visual comparison of different deraining methods on RainMotion dataset. The blue box indicates the comparison of rain streak removal. The red box indicates the comparison of detail retention.

Comparative methods. We compare our network against 18 state-of-the-art methods, including eight single-image deraining methods, and ten video deraining methods. Eight single-image deraining methods are DSC [25], LP [21], DetailNet [8], JORDER [37], PReNet [26], MSPFN [13], DCSFN [32], and MPRNet [40], while ten video deraining techniques are TCLRM [17], DIP [14], MS-CSC [19], SE [34], FastDerain [15], J4RNet [23], SpacCNN [5], FCDN [36], SLDNet [39] and S2VD [39]. To provide fair comparisons, we obtain FCDN's results from the authors. For other comparing methods, we use their public implementations, and re-train these networks on same benchmark datasets to obtain their best performance for a fair comparison.

5.1 Results on Real-world Rainy Videos

To evaluate the effectiveness of our video raining network, we collect 11 real-world rainy videos from Youtube website by comparing our network against state-of-the-art methods. Fig. 5 shows the derained results produced by our network and compared methods on real-world video frames. Apparently, DCSFN, MPRNet, MSCSC, FastDeRain, SLDNet and S2VD cannot fully remove rain streaks. Although eliminating rain streaks, FCDN tends to over-smooth clean background details. On the contrary, our method effectively removes rain streaks and better maintains background details than FCDN; see these magnified tree regions of Figs. 5 (f) and (i).

Table 4: Quantitative comparisons of ablation study on the RainMotion dataset.

Network	DT	AR	PSNR↑	SSIM ↑
basic	×	×	30.89	0.9351
basic+DT	✓	×	31.22	0.9367
our method	✓	✓	**31.82**	**0.9423**

5.2 Results on Synthetic Videos

Quantitative comparison. Table 1 reports PSNR and SSIM scores of our network and compared methods on the three existing benchmark datasets, while Table 2 compares the metrics results on our RainMotion dataset. As presented in Table 1, FCDN and S2VD has largest PSNR and SSIM scores among all compared methods. Compared with these two methods, our method has achieved a PSNR improvement of 7.85% and a SSIM improvement of 1.50% on RainSynLight25, a PSNR improvement of 16.85% and a SSIM improvement of 12.88% on RainSynHeavy25, and a PSNR improvement of 0.91% and a SSIM improvement of 0.38% on NTURain. Moreover, our method has larger PSNR and SSIM scores than all the competitors on our dataset, demonstrating that our network can more accurately recover clean video backgrounds; see Table 2.

Visual comparison. Fig. 6 shows the visual comparison between our network and state-of-the-art methods on an input rainy frame of RainSynHeavy25, which is the most challenging among existing benchmark datasets. Apparently, after removing rain streaks, our method can better preserve clean background image than state-of-the-art methods. It shows that our method has a more accurate video deraining result, which is further verfied by the superior PSNR/SSIM values of our method. Moreover, Fig. 7 presents visual comparisons between our network and state-of-the-art methods on our dataset. From these visual results, we can find that DCSFN, MPRNet and S2VD tend to produce artifacts with dark pixels, while MSCSC, FastDeRain and SLDNet maintain many rain steaks in their derained results. FCDN also cannot fully remove rain streaks, such as the grass region at the second row of Fig. 7. By progressively predicting additional rain motions, our RDD-Net can effectively eliminate rain streaks and better maintain non-rain background details; see our larger PSNR/SSIM scores.

5.3 Ablation Study

Baseline design. We also conduct ablation study experiments on the RainMotion dataset to evaluate two major modules (i.e., DT module and AR module) of our network. To do so, we construct two baseline networks. The first baseline (denoted as "basic") is to remove all AR modules from our network and modifying DT modules to predict only clean background layers for video deraining. It means that all DT modules in "basic" do not predict rain motion layers and rain streak layers. The second baseline (denoted as "basic+DT") is to add DT modules into "basic".

Quantitative comparison. Table 4 reports PSNR and SSIM scores of our method and two constructed baseline networks. First, "basic+DT" has a larger PSNR and SSIM scores than "basic", demonstrating that utilizing our DT modules to decouple temporal

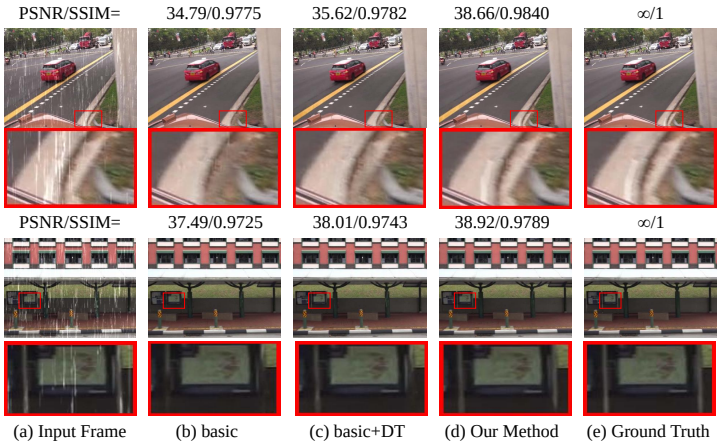

Fig. 8: Visual comparison of ablation study. (a) Input video rain frame. The results of (b) basic, (c) basic+

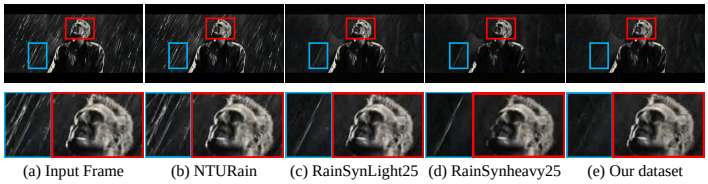

Fig. 9: Results of our network trained on different datasets.

features for predicting additional rain streak layers and rain motions layers helps our network to better recover the underlying clean background layer. Moreover, our network outperforms "basic+DT" in terms of PSNR and SSIM metrics. It indicates that leveraging our AR modules to attentively aggregate predictions of different recurrent steps enables our network to achieve superior video deraining performance.

Visual comparison. Fig. 8 shows derained results produced by our network and two baseline networks for different rain video frames. Apparently, "basic" and "basic+DT" tend to over-smooth background details when removing rain streaks of the video frames. On the contrary, our method is capable to better preserve these background details; as shown in Fig. 8. Moreover, our method has the larger PSNR and SSIM values than baseline networks, showing the superior video deraining performance of our method.

5.4 Discussions

Advantage of our dataset. One of main advantages of our dataset is that we introduce the **rain streak motion** to generate a more realistic rainy video data for network training. To prove the this advantage over other datasets, we re-train our network on our dataset and other three datasets (i.e., NTURain, RainSynLight25, and RainSynheavy25), separately, and test them on the same real-world rainy videos. Fig. 9 shows the results, where our network trained on our dataset (e) gets better result than those trained on other datasets.

Table 5: The results of our network with and without $\mathcal{L}_S$ and $\mathcal{L}_M$ on RainMotion.

	w/o $\mathcal{L}_S, \mathcal{L}_M$	w/o $\mathcal{L}_S$	w/o $\mathcal{L}_M$	RDD-Net
PSNR↑	30.68	31.09	30.92	**31.82**
SSIM↑	0.9353	0.9372	0.9364	**0.9423**

(a) Input Frame (b) FCDN (c) RDD-Net (d) Ground Truth

Fig. 10: A failure case of our RDD-Net.

The effect of the video rain prior. Compared to existing methods, our network utilizes additional supervisions (i.e., $\mathcal{L}_S$ and $\mathcal{L}_M$) on predicting the rain motion image (i.e., video rain prior) and the non-rain clean image for training due to the disentangled feature learning. Table 5 reports PSNR and SSIM scores of our method with and without $\mathcal{L}_S$ and $\mathcal{L}_M$. It shows that the performance of our network is reduced when removing $\mathcal{L}_S$ or $\mathcal{L}_M$, showing that the additional $\mathcal{L}_S$ and $\mathcal{L}_M$ help our network to achieve a better video deraining accuracy. Moreover, our method without $\mathcal{L}_S$ or $\mathcal{L}_M$ (see $w/o\mathcal{L}_S, \mathcal{L}_M$ of Table 5) still outperforms all existing video deraining methods, since the largest PSNR and SSIM scores of existing methods are 28.42 and 0.9203 (*i.e.*, MPRNet in Table 2).

Failure cases. Like other video deraining methods, our method cannot work well for a very heavy rain case, where the background details are almost completely covered by rain streaks; see Fig. 10 for an example input and results of FCDN and our method. We argue that such heavy rain case is rare in our daily life, and we can alleviate this issue by generating or collecting similar video data samples.

6 Conclusion

This paper presents a novel network for video rain streak removal. One of our key contributions is to devise a new video rain model by first embedding rain streak motions and collect a new dataset based on the rain model. The second contribution is the development of a novel network for video rain streak removal by decoupling the aggregated features from each pair of adjacent video frames into features for predicting a background layer, a rain motion, a background layer, and a rain layer, and then attentively integrate decoupled features from several pairs of adjacent frames. Experimental results on benchmark datasets and real-world rainy videos show that our network consistently outperforms state-of-the-art methods by a large margin. In future, we will also incorporate other rain degrading factors (e.g., fog/haze/raindrops) into our video rain model to further improve the robustness of our video rain streak removal network.

Acknowledgments: The work is supported by the National Natural Science Foundation of China (Grant No. 61902275, NSFC-U1803264), and a grant of Hong Kong Research Grants Council under General Research Fund (no. 15205919).

References

1. Barnum, P.C., Narasimhan, S., Kanade, T.: Analysis of rain and snow in frequency space. IJCV **86**(2-3), 256–274 (2010)
2. Bossu, J., Hautière, N., Tarel, J.P.: Rain or snow detection in image sequences through use of a histogram of orientation of streaks. IJCV **93**(3), 348–367 (2011)
3. Brewer, N., Liu, N.: Using the shape characteristics of rain to identify and remove rain from video. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). pp. 451–458. Springer (2008)
4. Chen, J., Chau, L.P.: A rain pixel recovery algorithm for videos with highly dynamic scenes. IEEE TIP **23**(3), 1097–1104 (2013)
5. Chen, J., Tan, C.H., Hou, J., Chau, L.P., Li, H.: Robust video content alignment and compensation for rain removal in a cnn framework. In: CVPR. pp. 6286–6295 (2018)
6. Chen, Y.L., Hsu, C.T.: A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In: ICCV. pp. 1968–1975 (2013)
7. Fu, X., Huang, J., Ding, X., Liao, Y., Paisley, J.: Clearing the skies: A deep network architecture for single-image rain removal. IEEE TIP **26**(6), 2944–2956 (2017)
8. Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X., Paisley, J.: Removing rain from single images via a deep detail network. In: CVPR. pp. 3855–3863 (2017)
9. Garg, K., Nayar, S.K.: Detection and removal of rain from videos. In: CVPR. pp. 528–535 (2004)
10. Garg, K., Nayar, S.K.: Photorealistic rendering of rain streaks. ACM Transactions on Graphics (TOG) **25**(3), 996–1002 (2006)
11. Haris, M., Shakhnarovich, G., Ukita, N.: Recurrent back-projection network for video super-resolution. In: CVPR (2019)
12. Huang, D.A., Kang, L.W., Wang, Y.C.F., Lin, C.W.: Self-learning based image decomposition with applications to single image denoising. IEEE Transactions on Multimedia **16**(1), 83–93 (2014)
13. Jiang, K., Wang, Z., Yi, P., Chen, C., Huang, B., Luo, Y., Ma, J., Jiang, J.: Multi-scale progressive fusion network for single image deraining. In: CVPR. pp. 8346–8355 (2020)
14. Jiang, T.X., Huang, T.Z., Zhao, X.L., Deng, L.J., Wang, Y.: A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In: CVPR. pp. 4057–4066 (2017)
15. Jiang, T.X., Huang, T.Z., Zhao, X.L., Deng, L.J., Wang, Y.: Fastderain: A novel video rain streak removal method using directional gradient priors. IEEE TIP **28**(4), 2089–2102 (2018)
16. Kang, L.W., Lin, C.W., Fu, Y.H.: Automatic single-image-based rain streaks removal via image decomposition. IEEE TIP **21**(4), 1742–1755 (2012)
17. Kim, J.H., Sim, J.Y., Kim, C.S.: Video deraining and desnowing using temporal correlation and low-rank matrix completion. IEEE TIP **24**(9), 2658–2670 (2015)
18. Li, G., He, X., Zhang, W., Chang, H., Dong, L., Lin, L.: Non-locally enhanced encoder-decoder network for single image de-raining. In: ACM Multimedia. pp. 1056–1064 (2018)
19. Li, M., Xie, Q., Zhao, Q., Wei, W., Gu, S., Tao, J., Meng, D.: Video rain streak removal by multiscale convolutional sparse coding. In: CVPR. pp. 6644–6653 (2018)
20. Li, X., Wu, J., Lin, Z., Liu, H., Zha, H.: Recurrent squeeze-and-excitation context aggregation net for single image deraining. In: ECCV. pp. 254–269 (2018)
21. Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Rain streak removal using layer priors. In: CVPR. pp. 2736–2744 (2016)
22. Liu, J., Yang, W., Yang, S., Guo, Z.: D3r-net: Dynamic routing residue recurrent network for video rain removal. IEEE TIP **28**(2), 699–712 (2018)

23. Liu, J., Yang, W., Yang, S., Guo, Z.: Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In: CVPR. pp. 3233–3242 (2018)
24. Liu, P., Xu, J., Liu, J., Tang, X.: Pixel based temporal analysis using chromatic property for removing rain from videos. Computer and information science **2**(1), 53–60 (2009)
25. Luo, Y., Xu, Y., Ji, H.: Removing rain from a single image via discriminative sparse coding. In: ICCV. pp. 3397–3405 (2015)
26. Ren, D., Zuo, W., Hu, Q., Zhu, P., Meng, D.: Progressive image deraining networks: A better and simpler baseline. In: CVPR. pp. 3937–3946 (2019)
27. Ren, W., Tian, J., Han, Z., Chan, A., Tang, Y.: Video desnowing and deraining based on matrix decomposition. In: CVPR. pp. 4210–4219 (2017)
28. Santhaseelan, V., Asari, V.K.: Utilizing local phase information to remove rain from video. IJCV **112**(1), 71–89 (2015)
29. Sun, S.H., Fan, S.P., Wang, Y.C.F.: Exploiting image structural similarity for single image rain removal. In: IEEE ICIP. pp. 4482–4486 (2014)
30. Tripathi, A.K., Mukhopadhyay, S.: A probabilistic approach for detection and removal of rain from videos. IETE Journal of Research **57**(1), 82–91 (2011)
31. Tripathi, A., Mukhopadhyay, S.: Video post processing: low-latency spatiotemporal approach for detection and removal of rain. IET image processing **6**(2), 181–196 (2012)
32. Wang, C., Xing, X., Wu, Y., Su, Z., Chen, J.: Dcsfn: Deep cross-scale fusion network for single image rain removal. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1643–1651 (2020)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
34. Wei, W., Yi, L., Xie, Q., Zhao, Q., Meng, D., Xu, Z.: Should we encode rain streaks in video as deterministic or stochastic? In: ICCV. pp. 2516–2525 (2017)
35. Yan, W., Tan, R.T., Yang, W., Dai, D.: Self-aligned video deraining with transmission-depth consistency. In: CVPR. pp. 11966–11976 (2021)
36. Yang, W., Liu, J., Feng, J.: Frame-consistent recurrent video deraining with dual-level flow. In: CVPR. pp. 1661–1670 (2019)
37. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Joint rain detection and removal from a single image. In: CVPR. pp. 1685–1694 (2017)
38. Yang, W., Tan, R.T., Wang, S., Liu, J.: Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence. In: CVPR. pp. 1720–1729 (2020)
39. Yue, Z., Xie, J., Zhao, Q., Meng, D.: Semi-supervised video deraining with dynamical rain generator. In: CVPR. pp. 642–652 (2021)
40. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: CVPR (2021)
41. Zhang, H., Patel, V.M.: Density-aware single image de-raining using a multi-stream dense network. In: CVPR. pp. 695–704 (2018)
42. Zhang, X., Li, H., Qi, Y., Leow, W.K., Ng, T.K.: Rain removal in video by combining temporal and chromatic properties. In: IEEE International Conference on Multimedia and Expo. pp. 461–464 (2006)
43. Zhu, L., Deng, Z., Hu, X., Xie, H., Xu, X., Qin, J., Heng, P.A.: Learning gated non-local residual for single-image rain streak removal. IEEE Transactions on Circuits and Systems for Video Technology **31**(6), 2147–2159 (2020)
44. Zhu, L., Fu, C.W., Lischinski, D., Heng, P.A.: Joint bi-layer optimization for single-image rain streak removal. In: ICCV. pp. 2526–2534 (Oct 2017)