

Frame-Consistent Recurrent Video Deraining with Dual-Level Flow

Wenhan Yang¹, Jiaying Liu^{2*}, Jiashi Feng¹

¹ Department of ECE, National University of Singapore, Singapore, 119077

² Institute of Computer Science and Technology, Peking University, Beijing, 100871

Abstract

In this paper, we address the problem of rain removal from videos by proposing a more comprehensive framework that considers the additional degradation factors in real scenes neglected in previous works. The proposed framework is built upon a two-stage recurrent network with dual-level flow regularizations to perform the inverse recovery process of the rain synthesis model for video deraining. The rain-free frame is estimated from the single rain frame at the first stage. It is then taken as guidance along with previously recovered clean frames to help obtain a more accurate clean frame at the second stage. This two-step architecture is capable of extracting more reliable motion information from the initially estimated rain-free frame at the first stage for better frame alignment and motion modeling at the second stage. Furthermore, to keep the motion consistency between frames that facilitates a frame-consistent deraining model at the second stage, a dual-level flow based regularization is proposed at both coarse flow and fine pixel levels. To better train and evaluate the proposed video deraining network, a novel rain synthesis model is developed to produce more visually authentic paired training and evaluation videos. Extensive experiments on a series of synthetic and real videos verify not only the superiority of the proposed method over state-of-the-art but also the effectiveness of network design and its each component.

1. Introduction

Rain, as a most common bad weather condition, will cause the visibility degradation in captured videos, *e.g.* content changes and detail loss, which may fail well-built outdoor computer vision systems by default taking clean video frames as input. For example, *rain streaks* cause intensity fluctuation of image content, obstruct the background to some extent, and blur the scene. Rain will also result

*Corresponding author. Email: liujiaying@pku.edu.cn. This work was supported by National Natural Science Foundation of China under contract No. 61772043, Beijing Natural Science Foundation under contract No. L182002 and No. 4192025, and CCF-DiDi BigData Joint Lab.



Figure 1. Visual results of different deraining methods. Compared with FastDeRain [24] and SpacCNN [7], our method is better at removing rain accumulation and accumulation flow (blue box) with less detail loss (red box).

in *occlusion*, with no background signals through the rain-drop. Another degradation factor due to rain is *accumulation*, where distant streaks are overlapped to look like mist or fog, which obscures the background and significantly reduces the visibility of distant scenes. When the amount of rainfall changes rapidly in a local area, rain accumulation fluctuates and leads to visual degradation like a layer of flowing veils covering on the rain-free background as shown in Fig. 2 (c) and the blue box in Fig. 1, which is called *accumulation flow* in this work. Compared to normal accumulation, the behavior of accumulation flow is more dynamic. Besides static scene transmission, it is also affected by local rain density and atmospheric flow, which makes its estimation much challenging. Furthermore, the accumulation flow possesses complex local motion patterns, which hinders both human perception as well as vision applications, and increases the difficulty to model and handle it.

A lot of research efforts have been dedicated to rain image/video restoration. Some works [25, 21, 42, 36] take a single rain image as input and separate rain streaks and

rain-free images (non-rain images) based on texture appearances. Frequency domain representation [25], sparse representation [36], Gaussian mixture model [31] and deep networks [53, 14] are adopted as basic models to differentiate rain streaks and rain-free images. Besides the above single image-based approaches, video-based methods [1, 2, 3, 8, 12, 16, 18, 19, 58] solve the problem by exploiting both spatial and temporal context. Some [18, 16, 19] leverage the physical aspects of rain, such as its directional and chromatic properties. Others [8, 6, 27, 24] further utilize temporal dynamics, including continuity of background motions, random occurrence of streaks in video frames, and explicit motion cues, to facilitate video rain removal.

Recently, the rapid development of deep networks also leads to the blooming of deep learning-based and video image processing, including denoising [57], JPEG artifacts removal [9, 59], interpolation [52], super-resolution [10, 55, 51, 50], video compression [20, 48, 32], and single-image rain removal [15, 37, 54], *etc.* Likewise, deep learning also brings new progress to video rain removal. In [29], a multiscale convolutional sparse coding is proposed for video rain streak removal. Chen *et al.* [7] propose to first segment a rain image into superpixels, on which a consistency constraint is imposed, and then compensate for the lost details in the aligned superpixels. In [33, 34], a recurrent network is built to jointly perform rain degradation classification, rain removal and background detail reconstruction.

These previous methods achieve good performance in some cases. However, to our best knowledge, they all focus on one or two of the rain degradation factors, and most of them only consider rain streak removal. Some other rain degradation factors for vision tasks in videos are rarely considered, such as rain accumulation flow as shown in Fig. 2 (c) and the blue box in Fig. 1. Furthermore, it is not fully explored how to utilize intra-frame and inter-frame context to facilitate joint estimation of multiple rain-related factors. Besides, in real rain scenes, the motions of objects and regions are interweaved. Their inferences are disturbed by the multiple degradation factors and their mixture effects. It remains unclear how to describe motion information at both pixel and region levels and model the motion patterns of the rain-free frames robustly and accurately.

Based on the above observations, we solve the problem of video rain removal more comprehensively by considering rain streak, accumulation, rain accumulation flow and occlusion. A two-stage recurrent network is designed and a novel video rain synthesis model is built for synthesizing visually authentic rain videos with various rain types. The proposed recurrent network estimates the rain-free frame by two stages. At the first stage, the model recovers a rain-free frame roughly from a single rain frame. Then, based on this estimation and preceding recovered clean frames, a more accurate estimation for the clean frame is inferred at the

second stage by using the temporal context information. We also place two types of flow-based orthogonal constraints at the second stage to regularize the model learning at both pixel and region-levels, which makes our results more temporally continuous and regionally consistent, and facilitates a frame-consistent method.

In summary, our contributions are as follows.

- We develop a new rain synthesis model that includes several visual degradation factors, namely rain streak, accumulation, accumulation flow and occlusion. Based on this model, we synthesize a new rain video dataset, corresponding to light and heavy rain condition respectively, to support development and evaluation of data-driven video rain removal methods.
- We build a recurrent network (RNN) to predict the rain-related variables in our novel rain synthesis model, and to estimate the rain-free frame to perform an inverse process based on these predicted variables. The injection of the inverse recovery module makes our network more effective.
- Our RNN has a two-stage architecture which fully makes use of the potential of single-frame and multi-frame context. It is capable of extracting more reliable motion information based on the initially estimated rain-free frame as guidance for alignment and motion modeling at the second stage.
- To better capture the motion patterns and keep inter-frame consistency, we employ two types of flow-based representations to regularize the learning of our video deraining network at both coarse flow and fine levels.

2. Related Work

Single image deraining is a highly ill-posed problem. To address it, many models and priors are used to perform signal separation and texture classification. These models include sparse coding [25], generalized low rank model [8], nonlocal mean filter [26], discriminative sparse coding [36], Gaussian mixture model [31], rain direction prior [56], transformed low rank model [5]. The presence of deep learning has promoted the development of single image deraining. In [14, 13], deep networks take the image detail layer as their input. Yang *et al.* [53] propose a deep joint rain detection and removal method to remove heavy rain streaks and accumulation. In [56], a novel density-aware multi-stream densely connected CNN is proposed for joint rain density estimation and removal. Wang *et al.* [45] develop a perceptual generative adversarial network to apply the translation from rainy images to clean ones.

Video rain removal can additionally make use of the temporal context and motion information. Garg and Nayar are the first to focus on rain modeling [18] and removal [16, 19, 17]. Later works formulate rain streaks

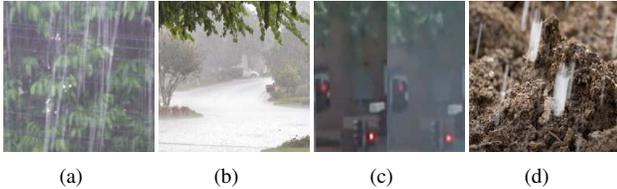


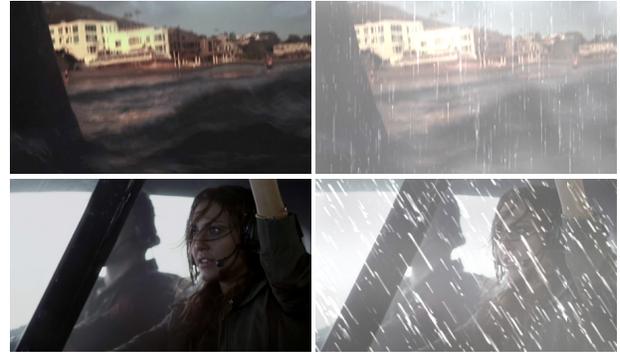
Figure 2. Different types of visibility degradation due to rain. (a) Rain streaks. (b) Rain accumulation. (c) Rain accumulation flow. Due to the atmosphere flow, the density of the veiling layers at the same location of two frames changes. (d) Rain occlusion. The occlusion regions present an identical intensity.

with more flexible and intrinsic characteristics, including temporal and chromatic properties of rain [58, 35], Fourier domain feature [1], phase congruency features [40], the size, shape and orientation of rain streaks [3, 2], spatio-temporal correlation of local patches [8], and overall directional tendency of rain streaks [24]. The presence of learning-based method, with improved modeling capacity, brings new progress. Chen *et al.* [6] propose to embed motion segmentation by a Gaussian mixture model into rain detection and removal. Tripathi *et al.* [43, 44] train a Bayes rain detector based on spatial and temporal features. In [27], Kim *et al.* train an SVM to refine the roughly detected rain maps. Wei *et al.* [47] encode rain streaks as patch-based mixtures of Gaussian, which is capable of finely adapting a wider range of rain variations. In [39], a matrix decomposition model is used to divide rain streaks into sparse and dense ones. In [29], a multiscale convolutional sparse coding method is proposed for video rain streak removal. Chen *et al.* [7] propose to first segment a rain image into superpixel and then enforce the consistency constraints and compensate for lost details on these aligned superpixels. In [33], a recurrent network is built to seamlessly integrate rain degradation classification, rain removal and background details reconstruction. Comparatively, in our work, we aim to handle more types of visibility degradation via our proposed rain synthesis model. To better jointly utilize intra-frame and inter-frame context, we design a two-step RNN for video deraining. To better model motion patterns and keep inter-frame consistency at different granularities, we propose to apply dual-level flow constraints to regularize the model learning.

3. Comprehensive Rain Synthesis Model

To address rain removal problem more comprehensively, we propose a novel comprehensive rain synthesis model to facilitate the research on this topic. The rain images are synthesized from clean ones considering four degradation factors as below and tries to simulate the corresponding effects.

Rain Streaks. The fast falling raindrops in the focus of a camera will obstruct the background to some extent, as



(a) Rain-Free Frame (b) Synthesized Rain Frame

Figure 3. Examples of our synthesis data based on Eqn. (4).

shown in Fig. 2 (a). To synthesize similar effects, they are combined with clean rain-free frames by linear additions [53, 14, 31].

Rain Accumulation. The rain streaks in the distant are interleaved to produce atmospheric veiling effects [53, 30], as shown in Fig. 2 (b). In our synthesis model, the background signals are scattered out to degrade the visibility.

Rain Accumulation Flow. The out-of-focus raindrops (or flowing rain accumulation) form rain accumulation flow as proposed in this work, as shown in Fig. 2 (c). Its degree of transparency is not correlated to the background depth, and it can take any shape and make semi-transparent veiling effects. Its presence in the temporal domain is continuous. It will be added to rain streak contaminated images like a veil.

Rain Occlusion. In moderate or heavy rain, the light transmittance of raindrops becomes low and the rain region has identical intensities [33]. In this case, the background information is totally lost, as shown in Fig. 2 (d). It is generated by an alpha matting process based on a binary mask, with rain-contaminated images and given intensity maps.

We formulate our comprehensive rain synthesis model starting from the simplest case of a widely used single-frame rain model [31, 36, 22]:

$$\mathbf{O} = \mathbf{B} + \mathbf{S}, \quad (1)$$

where \mathbf{B} is the rain-free frame without rain streaks, and \mathbf{S} is the rain streak frame. \mathbf{O} is the captured image with rain streaks. A video rain synthesis model is obtained with a temporal indicator t added:

$$\mathbf{O}_t = \mathbf{B}_t + \mathbf{S}_t, \quad t = 1, 2, \dots, N, \quad (2)$$

where t and N denote the current time-step and the total number of video frames, respectively. Rain streaks \mathbf{S}_t are assumed to be independent identically distributed random samples [41]. Their locations across the frames are assumed uncorrelated. Considering rain accumulation and accumulation flow, Eqn. (2) is further extended as follows:

$$\mathbf{O}_t = \beta_t \mathbf{B}_t + (1 - \beta_t) \mathbf{A}_t + \mathbf{U}_t + \mathbf{S}_t, \quad t = 1, 2, \dots, N, \quad (3)$$

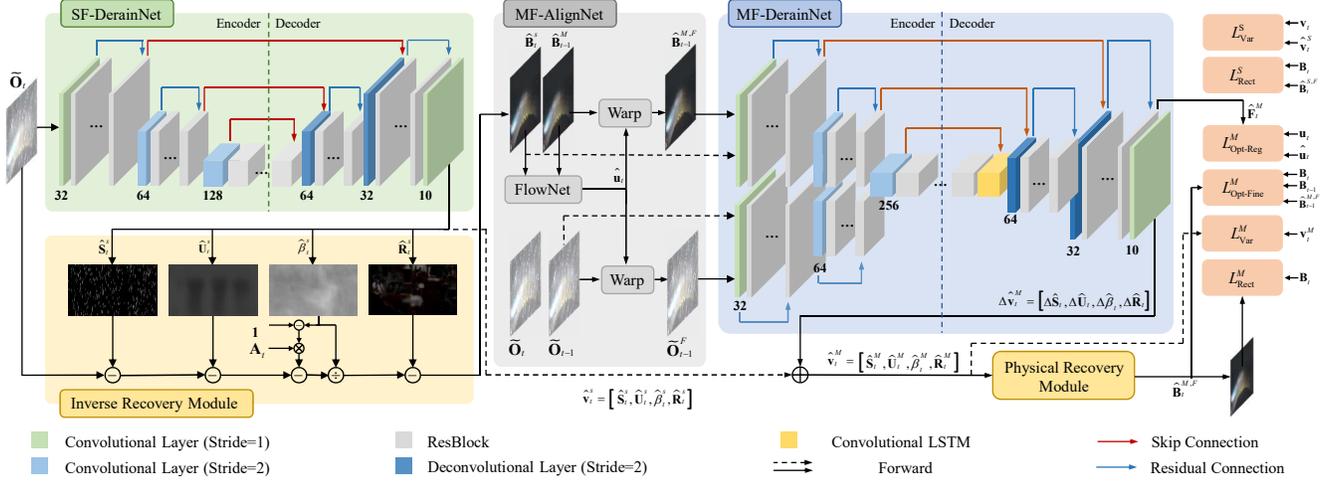


Figure 4. Framework of our two-stage recurrent network for video deraining. The *SF-DerainNet* takes single-frame rain input and outputs four rain-related variables. *Inverse recovery module* converts these predicted rain-related variables into the estimated rain-free frame following the inverse recovery of Eqns. (3) and (4). *Multi-frame alignment network* estimates the optical-flow among frames and aligns previous frames to the current one. After that, *MF-DerainNet* takes the multi-frame rain input and their clean estimations to predict the residual rain-related variables. These variables are further combined with the variables estimated from the single frame to eventually produce the final estimation of the rain-free frame. The whole model is trained end-to-end with the losses for variable prediction, rain-free frame recovery, inter-frame consistency, and motion accuracy. Best viewed in color.

where A_t is the global atmospheric light, β_t is the atmospheric transmission which is correlated with the scene depth, and U_t is the rain accumulation flow layer which is decided by the local raindrop density and atmospheric flow. They are assumed temporally continuous. Given a fixed scene, $\{A_t\}$ and $\{\alpha_t\}$ are only affected by camera motions. $\{U_t\}$ has its own motion trajectory. Finally, we make the rain model capable of describing occlusion by

$$\tilde{O}_t = (1 - \alpha_t) O_t + \alpha_t M_t, \quad (4)$$

where O_t is defined in Eqn. (3), M_t is the rain reliance map and α_t is an alpha matting map.

Based on this comprehensive rain synthesis model (4), we can simulate realistic-looking rain videos. Two synthesized examples are shown in Fig. 3. In our work, we build a new video rain dataset with such a synthesis model. More details are provided in Sec. 5.

4. Frame-Consistent Deraining Network

4.1. Design Methodology

Our framework is built with the following design methodology. First, our method strictly follows the inverse recovery process of Eqns. (3) and (4), which is very tractable. Second, our method adopts a two-step framework, which fully utilizes the benefits of single-frame and multi-frame context. The single-frame derained results are taken as guidance for multi-frame deraining, which facilitates more accurate motion estimation and alignment. Third, two complementary constraints, pixel-level fine flow and region-level regularized flow constraints, are utilized

to regularize the learning of our video deraining network. They jointly make the motion patterns of our results more accurate and consistent across frames.

4.2. Network Architecture

Our framework consists of five modules: single-frame deraining network (*SF-DerainNet*), multi-frame alignment network (*MF-AlignNet*), multi-frame deraining network (*MF-DerainNet*), **Inverse Recovery Module**, and **Loss Function**, as shown in Fig. 4. It performs multi-frame rain removal by two steps. First, the *SF-DerainNet* is utilized to estimate the rain-related variables in Eqns. (3) and (4). Then, the inverse recovery module takes the rain-related variables as its input and estimates the rain-free frame by calculating the inverse recovery process in Eqns. (3) and (4). Then, based on the estimated rain-free frame B_t^S at time-step t from *SF-DerainNet*, and B_{t-1}^M at time-step $(t-1)$ from *MF-DerainNet*, we estimate the optical flow at time-step t . We use this flow to warp O_{t-1} and B_{t-1}^M to get O_{t-1}^F and $B_{t-1}^{M,F}$ by aligning them to O_t and B_t , respectively. After that, *MF-DerainNet* extracts features from the input $[B_t^S, B_{t-1}^{M,F}]$ and $[\tilde{O}_t, \tilde{O}_{t-1}^F]$, and then concatenates and transforms these features into the rain-related variables, which are further fed into the inverse recovery module to obtain the final estimation $B_t^{M,F}$. Multiples losses are used to jointly constrain the recovery of $B_t^{M,F}$ to predict rain-free frames accurately and keep the inter-frame consistency.

Single-Frame Deraining Network. As shown in Fig. 4, *SF-DerainNet* takes a U-Net-like architecture. It transforms

the features by multiple convolutional layers progressively. In intermediate layers, the spatial resolutions of features are first down-sampled (encoder) and then up-sampled (decoder). There are residual connections (denoted by red) to connect the features with the same spatial resolution within the encoder (or decoder), which help local information contained in the features generated by shallow layers reach the output. There are also skip connections (denoted by blue) to link the features with the same spatial resolution from encoder to decoder. The spatial resolution change is achieved by stride convolution and deconvolution in SF-DerainNet. The SF-DerainNet outputs four rain-related variables. We use $\mathbf{G}_{\text{SF}}(\cdot)$ to represent the process of SF-DerainNet:

$$\hat{\mathbf{v}}_t^S = [\hat{\mathbf{S}}_t^S, \hat{\mathbf{U}}_t^S, \hat{\beta}_t^S, \hat{\mathbf{R}}_t^S] = \mathbf{G}_{\text{SF}}(\hat{\mathbf{O}}_t), \quad (5)$$

where $\hat{\mathbf{S}}_t^S$, $\hat{\mathbf{U}}_t^S$, $\hat{\beta}_t^S$, and $\hat{\mathbf{R}}_t^S$ are rain streak, rain accumulation flow, atmospheric transmission and residue estimated from the single frame. The last term tries to remedy the effects of rain occlusion and the estimation errors of other terms.

Inverse Recovery Module. Given $\hat{\mathbf{S}}_t^S$, $\hat{\mathbf{U}}_t^S$, $\hat{\beta}_t^S$, and $\hat{\mathbf{R}}_t^S$, we follow the inverse solution of Eqn. (3) to get the estimation of the clean background frame $\hat{\mathbf{B}}_t^S$ based on a single frame rain input:

$$\hat{\mathbf{B}}_t^S = \frac{\hat{\mathbf{O}}_t - \hat{\mathbf{U}}_t^S - \hat{\mathbf{S}}_t^S - (1 - \hat{\beta}_t^S) \times \hat{A}_t}{\max(\hat{\beta}_t^S, \epsilon)} + \hat{\mathbf{R}}_t^S, \quad (6)$$

where ϵ is a threshold to guarantee the numerical reasonability, which is set to 0.1 [38].

Multi-Frame Alignment Network. At time-step t , we estimate the $(t-1)$ -th rain-free frame $\hat{\mathbf{B}}_{t-1}^M$ from previous rain frames. Then, we estimate the optical flow $\mathbf{u}_t = [\mathbf{u}_{t,x}, \mathbf{u}_{t,y}]$ between $\hat{\mathbf{B}}_{t-1}^M$ and $\hat{\mathbf{B}}_t^S$ by FlowNet [11]. We use $\mathbf{G}_{\text{Flow}}(\cdot)$ and $\mathbf{G}_{\text{Warp}}(\cdot)$ to denote the process of optical flow calculation and warping operation based on the flow:

$$\mathbf{u}_t = \mathbf{G}_{\text{Flow}}(\hat{\mathbf{B}}_{t-1}^M, \hat{\mathbf{B}}_t^S), \quad (7)$$

$$\hat{\mathbf{B}}_{t-1}^{M,F} = \mathbf{G}_{\text{Warp}}(\hat{\mathbf{B}}_{t-1}^M, \hat{\mathbf{u}}_t), \quad (8)$$

$$\hat{\mathbf{O}}_{t-1}^F = \mathbf{G}_{\text{Warp}}(\hat{\mathbf{O}}_{t-1}, \hat{\mathbf{u}}_t). \quad (9)$$

Then, the locations of the pixels in rain inputs are aligned, and the successive MF-DerainNet is capable of removing rain more effectively.

Multi-Frame Deraining Network. The MF-DerainNet takes as input $[\hat{\mathbf{B}}_t^S, \hat{\mathbf{B}}_{t-1}^{M,F}]$ and $[\tilde{\mathbf{O}}_t^S, \tilde{\mathbf{O}}_{t-1}^F]$ and predicts the rain-related variables. It uses a similar architecture to SF-DerainNet. Differently, there are two branches at the encoder side. Features are extracted from $[\hat{\mathbf{B}}_t^S, \hat{\mathbf{B}}_{t-1}^{M,F}]$ and

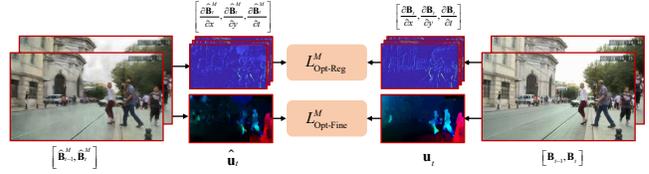


Figure 5. The dual-level flow constraint deduced from Eqn. (17). $[\frac{\partial \mathbf{B}(p)}{\partial x}, \frac{\partial \mathbf{B}(p)}{\partial y}, \frac{\partial \mathbf{B}(p)}{\partial t}]$ and $\mathbf{u} = (\mathbf{u}_x, \mathbf{u}_y)$ provide pixel and region-level regularization. Best viewed in color.

$[\tilde{\mathbf{O}}_t^S, \tilde{\mathbf{O}}_{t-1}^F]$, respectively, and concatenated before down-sampling to the smallest spatial resolution. The skip connections are built to connect the corresponding features from the branch taking as input the estimated clean frames to the decoder side. At the end of the convolutional layers which have the smallest scales at the decoder side, we use a convolutional LSTM to propagate the information at the feature level across frames. Here, we utilize a residual task learning method. We use $\mathbf{G}_{\text{MF}}(\cdot)$ to express the process of MF-DerainNet:

$$\Delta \hat{\mathbf{v}}_t^M = \mathbf{G}_{\text{MF}}(\hat{\mathbf{B}}_t^S, \hat{\mathbf{B}}_{t-1}^{M,F}, \tilde{\mathbf{O}}_t, \tilde{\mathbf{O}}_{t-1}^F), \quad (10)$$

$$\hat{\mathbf{v}}_t^M = \Delta \hat{\mathbf{v}}_t^M + \hat{\mathbf{v}}_t^S. \quad (11)$$

After getting $\hat{\mathbf{v}}_t^M = [\hat{\mathbf{S}}_t^M, \hat{\mathbf{U}}_t^M, \hat{\beta}_t^M, \hat{\mathbf{R}}_t^M]$, we put it into the inverse recovery module to obtain the rain-free frame $\hat{\mathbf{B}}_t^M$.

Loss Function. We train our network in an end-to-end manner. The loss function consists of six terms:

$$L_{\text{all}} = L_{\text{Rect}}^M + L_{\text{Var}}^M + L_{\text{Rect}}^S + L_{\text{Var}}^S \quad (12)$$

$$+ L_{\text{Opt-Reg}}^M + L_{\text{Opt-Fine}}^M, \quad (13)$$

$$L_{\text{Rect}}^S = \left\| \hat{\mathbf{B}}_t^S - \mathbf{B}_t \right\|_2^2, \quad (14)$$

$$L_{\text{Var}}^S = \left\| \hat{\mathbf{v}}_t^S - \mathbf{v}_t \right\|_2^2. \quad (15)$$

The same applies to L_{Rect}^M and L_{Var}^M . $L_{\text{Opt-Reg}}^M$ and $L_{\text{Opt-Fine}}^M$ are the losses measured by coarse flow consistency and fine flow orthogonal feature.

4.3. Dual-Level Flow Constraints

To generate more temporally continuous and regionally consistent videos, we employ two types of constraints for network training, fine flow constraint and regularized flow constraint, which are first proposed by this work.

Brightness Constant Constraint. We first review the brightness constant constraint used in the traditional optical flow:

$$\mathbf{B}(x, y, t) = \mathbf{B}(x + \Delta x, y + \Delta y, t + \Delta t), \quad (16)$$

where $\mathbf{B}(x, y, t)$ denotes the pixel at the location (x, y) of a frame at time t . For frames t and $(t + \Delta t)$, Δx and Δy are

Table 1. PSNR and SSIM results among different rain streak removal methods on *RainSynLight25*, *RainSynComplex25*, and *NTURain*. Best results are denoted in red and the second best results are denoted in blue.

Dataset	Metric	DetailNet	TCLRM	JORDER	MS-CSC	DSC	SE	FastDeRain	J4RNet	SpacCNN	Proposed
<i>RainSynLight25</i>	PSNR	25.72	28.77	30.37	25.58	25.63	26.56	29.42	32.96	32.78	35.80
	SSIM	0.8572	0.8693	0.9235	0.8089	0.8328	0.8006	0.8683	0.9434	0.9239	0.9622
<i>RainSynHeavy25</i>	PSNR	16.50	17.31	20.20	16.96	17.33	16.76	19.25	24.13	21.21	27.72
	SSIM	0.5441	0.4956	0.6335	0.5049	0.5036	0.5293	0.5385	0.7163	0.5854	0.8239
<i>NTURain</i>	PSNR	30.13	29.98	32.61	27.31	29.20	25.73	30.32	32.14	33.11	36.05
	SSIM	0.9220	0.9199	0.9482	0.7870	0.9137	0.7614	0.9262	0.9480	0.9474	0.9676

Table 2. PSNR and SSIM results among different rain removal methods on *RainSynAll100*. † and ‡ denote using ST-MRF and EVD-Net as pre/post-processing, respectively. Best results are denoted in red and the second best results are denoted in blue.

Metric	†FastDeRain	FastDeRain†	SpacCNN†F	†SpacCNN	†MS-CSC	MS-CSC†	†SE	SE†	J4RNet-E	Rain Input
PSNR	19.46	19.40	18.39	19.16	17.81	17.82	17.41	17.58	20.31	12.01
SSIM	0.6875	0.7322	0.7131	0.7214	0.6264	0.6211	0.6213	0.6245	0.6324	0.5739
Metric	‡FastDeRain	FastDeRain‡	SpacCNN‡F	‡SpacCNN	‡MS-CSC	MS-CSC‡	‡SE	SE‡	J4RNet-P	Proposed
PSNR	18.55	18.78	17.93	17.94	16.92	16.92	17.67	17.89	22.93	25.72
SSIM	0.7161	0.7351	0.7259	0.7270	0.6354	0.6346	0.6200	0.6278	0.7746	0.8989

the spatial pixel displacements along x and y axes, respectively. We can approximate Eqn. (16) with a Taylor series:

$$\frac{\partial \mathbf{B}(p)}{\partial x} \mathbf{u}_x + \frac{\partial \mathbf{B}(p)}{\partial y} \mathbf{u}_y + \frac{\partial \mathbf{B}(p)}{\partial t} = 0, \quad (17)$$

where $p = (x, y, t)$ and $\mathbf{u} = (\mathbf{u}_x, \mathbf{u}_y)$ denote the two dimensional velocity of point p , respectively. $\frac{\partial \mathbf{B}(p)}{\partial x}$ and $\frac{\partial \mathbf{B}(p)}{\partial y}$ are the spatial gradients of $\partial \mathbf{B}(p)$ along x and y axes respectively. $\frac{\partial \mathbf{B}(p)}{\partial t}$ is the temporal gradient along the time axis. From Eqn. (17), we get two types of flow-related representations: $\left[\frac{\partial \mathbf{B}(p)}{\partial x}, \frac{\partial \mathbf{B}(p)}{\partial y}, \frac{\partial \mathbf{B}(p)}{\partial t} \right]$ and $\mathbf{u} = (\mathbf{u}_x, \mathbf{u}_y)$, helping our network capture motion clues and keep temporal consistency at pixel and region-levels jointly.

Fine Flow Constraint. $\left[\frac{\partial \mathbf{B}(p)}{\partial x}, \frac{\partial \mathbf{B}(p)}{\partial y}, \frac{\partial \mathbf{B}(p)}{\partial t} \right]$ contains the full motion information among frames. It is a pixel-level feature that is directly calculated from the input frames. The feature may not be robust enough, e.g. disturbed by rain streaks and small occlusion, and may fail to capture motion patterns at the region or object level. With these considerations, following the process of extracting the feature $\mathbf{G}_{\text{Opt-Fine}}(\cdot)$, we build a fine flow constraint to guide the network to recover the accurate pixel motions:

$$L_{\text{Opt-Fine}}^M = \left\| \mathbf{G}_{\text{Opt-Fine}}(\hat{\mathbf{B}}_{t-1}^M, \hat{\mathbf{B}}_t^M) - \mathbf{G}_{\text{Opt-Fine}}(\mathbf{B}_{t-1}, \mathbf{B}_t) \right\|_2^2.$$

Regularized Flow Constraint. The optical flow \mathbf{u} is a more abstract feature. It cannot be directly calculated from the inputs and needs to be solved with other imposed constraints. After estimated and refined with priors, the optical flow is consistent within regions and objects, while some motion details are smoothed. Hence, it is not a pixel-level feature but can be used as an effective description for region and object motions. In our work, we use optical flow to regularize feature learning, making the motion patterns of the

estimated results more region-consistent:

$$L_{\text{Opt-Reg}}^M = \left\| \mathbf{G}_{\text{Flow}}(\hat{\mathbf{B}}_{t-1}^M, \hat{\mathbf{B}}_t^M) - \mathbf{G}_{\text{Flow}}(\mathbf{B}_{t-1}, \mathbf{B}_t) \right\|_2^2.$$

With fine flow constraint and regularized flow constraint, our network is capable of capturing motion trajectories and patterns at both pixel-level and region-level.

5. Experimental Results

Datasets. We compare our model with state-of-the-arts on five benchmark datasets. *RainSynLight25* and *RainSynComplex25* are proposed in [33] with light and heavy rain streaks respectively. *NTURain* [7] has two sub-groups: one taken from a panning and unstable camera with slow movements, and the other from a fast moving car-mount camera. *RainSynAll100* is synthesized by 1,000 non-rain sequences with all degradation factors illustrated in Sec. 3. The non-rain sequences are sampled from Vimeo-90K Dataset [49]. The whole dataset is split into training and testing datasets, including 900 and 100 video sequences, respectively. *Practical rain video sequences* are collected from practical scenes from Youtube website¹, GIPHY² and movie clips. More information about training data, implementation details, and video comparison results are provided in the on-line supplementary material³.

Baselines. We compare D3R-Net with state-of-the-art methods: discriminative sparse coding (DSC) [36], layer priors (LP) [31], joint rain detection and removal (JORDER) [53], deep detail network (DetailNet) [14], stochastic encoding (SE) [47], temporal correlation and low-rank matrix completion (TCLRM) [27], FastDeRain [24], joint recurrent rain removal and reconstruction (J4RNet) [33], superpixel alignment and compensation CNN (SpacCNN) [7]. DSC, LP, JORDER and DetailNet

¹<https://www.youtube.com/>

²<https://giphy.com/>

³<https://github.com/flyywh/Dual-FLow-Video-Deraining-CVPR-2019>

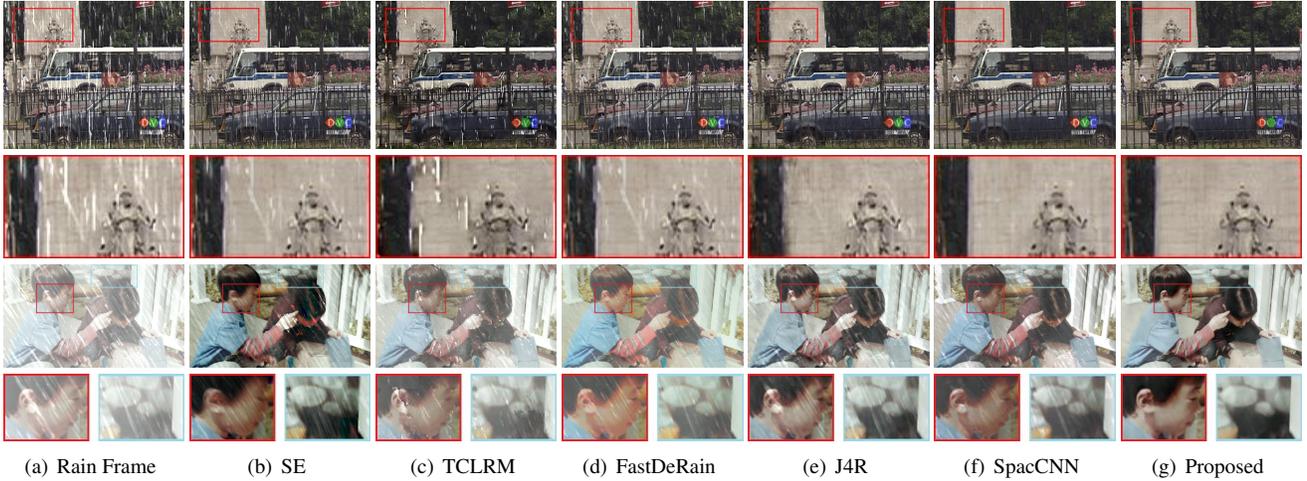


Figure 6. Results of rain removal methods on synthesized datasets. Top panel: the results on *RainSynComplex25*. Bottom panel: the results on *RainSynAll100*. Best viewed in color.

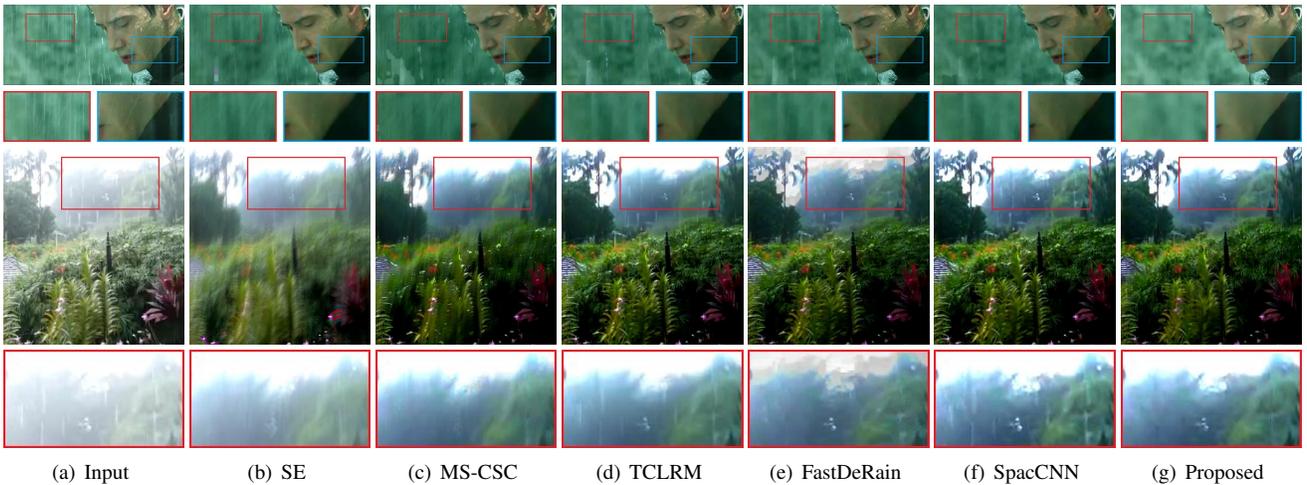


Figure 7. Results of rain removal methods on real images. Best viewed in color.

are single frame deraining methods. SE, TCLRM, FastDerain, J4RNet, and SpacCNN are multi-frame deraining methods. JORDER, DetailNet, J4RNet and SpacCNN are deep-learning based methods. When performing evaluations on *RainSynAll100*, for the methods without rain accumulation removal, end-to-end united video dehazing and detection network (EVD-Net) [28] and spatio-temporal MRF dehazing [4] are used to perform pre-processing or post-processing. For the experiments on synthesized data, Peak Signal-to-Noise Ratio (PSNR) [23] and Structure Similarity Index (SSIM) [46] are used as comparison criteria. Following previous works, we evaluate the results only in the luminance channel, since human visual system is more sensitive to luminance than chrominance information.

Objective Evaluation. We compare our method on datasets with only rain streak degradation in Table 1. Our method shows significant superiority to previous methods. Compared to J4RNet and SpacCNN, our method achieves more

than 2.8 dB on 3.5 dB gains on *RainSynLight25* and *RainSynHeavy25*. We also evaluate all methods on our synthesized rain dataset *RainSynAll100*. For a fair comparison, FastDerain, SpacCNN and MS-CSC are evaluated with two state-of-the-art dehazing methods, ST-MRF [4] and EVD-Net [28], as pre/post-processing to remove rain accumulation. In Table 2, \dagger FastDerain denotes using the sequential combination of ST-MRF and FastDeRain for rain removal. \dagger FastDerain utilizes ST-MRF as post-processing. \ddagger FastDerain and \ddagger FastDerain utilize EVD-Net as pre/post-processing, respectively. The same applies to other compared methods. J4RNet-E applies the method in [33] to directly predict the rain-free frame based on the input rain frame. J4RNet-P incorporates the inverse recovery module to first predict the rain-related variables and then infer the rain-free frame based on these predicted variables. As is observed in Table 2, our method achieves the best performance. The performance gain is up to 2.5dB in PSNR and

Table 3. Running time of different methods (in sec) to remove rain in a video with the spatial resolution 832×512 per frame.

Baseline	DetailNet	JORDER	SE	FastDeRain	TCLRM
Time	1.4698	0.6329	19.8516	0.3962	192.7007
Baseline	SpacCNN	MS-CSC	J4RNet-E	J4RNet-P	Proposed
Time	9.5075	15.7957	0.8401	0.8414	0.8974

0.12 in SSIM.

Subjective Evaluation We also compare visual results of different deraining methods in Figs. 6 and 7. From the top panel of Fig. 6, the results on synthesized data show our significant superiority in rain streak removal and detail preservation. The bottom panel of Fig. 6 demonstrates our advantages in removing rain streak, accumulation, accumulation flow, and even their combinations. Our method also performs better than other methods on real images (Fig. 7). The results in the top panel show that our method successfully removes all sharp streaks, weak streaks and veil streaks, while the results of other methods still display residual rain to some extent. The results in the bottom panel demonstrate our superiority of removing rain streak, rain accumulation and accumulation flow. Our method removes all streaks, generates the sharp boundaries between trees and sky, and better keeps regionally consistent.

Running Time. Table 3 compares the running time of several state-of-the-art methods. J4RNet-E, J4RNet-P and the proposed methods are implemented in Python and based on Pytorch⁴. Other methods are implemented in MATLAB. DetailNet and SpacCNN are based on MatConvNet⁵. JORDER is implemented on the Caffe Matlab wrapper⁶. TCLRM is a CPU-based method and others are GPU-based approaches. The video resolution for testing is 832×512 . In general, the running time of our method is comparable to other state-of-the-art methods.

Ablation Study of Network Architecture We evaluate our methods with different components in Table 4 and Fig. 8. It is clearly observed from Table 4 that SF-DerainNet and LSTM memory among frames significantly improve the objective results (v_1 vs. v_4 and v_2 vs. v_4). The inverse recovery module benefits rain streak removal, which leads to a higher SSIM (v_3 vs. v_4). MF-Alignment further improves PSNR and SSIM (v_4 vs. v_5). From the visual results in Fig. 8, the absence of SF-DerainNet (v_1) leads to obvious detail loss. Though v_4 may achieve very high PSNR in Table 4 without the inverse recovery module, it fails to remove rain streak and capture the intensity distribution. Comparatively, our full version achieves the best visual quality.

Visual Results with and without Dual-Level Flow Constraints We also compare the visual results generated by the models with and without dual flow constraints in Fig. 9, which are used to facilitate generating more temporally consistent and visually authentic results. Two very hard cases

⁴<https://pytorch.org/>

⁵<http://www.vlfeat.org/matconvnet/>

⁶<http://caffe.berkeleyvision.org/>

Table 4. Ablation analysis for network architecture. Best results are denoted in red and the second best results are denoted in blue.

Baseline	v_1	v_2	v_3	v_4	v_5
SF-DerainNet	×	✓	✓	✓	✓
Inverse Recovery	✓	✓	×	✓	✓
LSTM Memory	✓	×	✓	✓	✓
MF-Alignment	×	×	×	×	✓
PSNR	21.70	22.87	26.11	25.89	26.10
SSIM	0.8120	0.8183	0.8683	0.9043	0.9125



Figure 8. Example results of different methods on *RainSynAll100*. Crop results from left to right: ground truth, v_1 , v_2 , v_3 , v_4 , and v_5 .

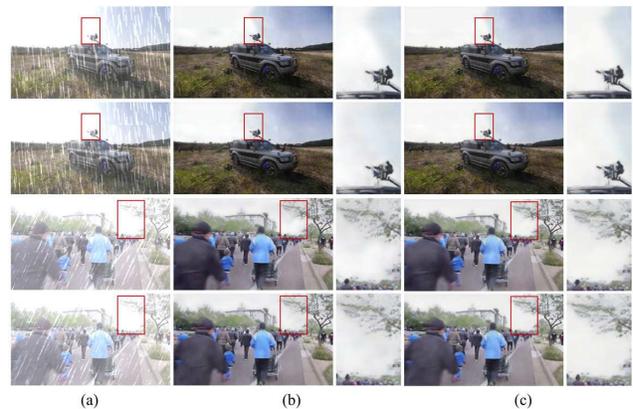


Figure 9. Two examples of successive frame results generated by our methods with and without dual flow constraints on *RainSynAll100*. (a) Input frames. (b) Results without dual flow constraints. (c) Results with dual flow constraints.

with large sky regions, which are easily regarded as the accumulation or accumulation flow are investigated. From the results, it is clearly observed that, without dual flow constraints, our results include less artifacts, and are more regionally consistent and temporally smooth between frames.

6. Conclusion

In this paper, we address the problem of rain removal from videos in a more comprehensive way. The degradation factors of rain streak, accumulation, accumulation flow and occlusion, are considered and a two-stage recurrent network with dual flow constraints is constructed. This two-stage recurrent network extracts more reliable motion information progressively. Rain-related variables (*e.g.* rain streak, atmospheric transmission) are estimated to infer the rain-free frame by the inverse process of the rain synthesis model. Furthermore, two types of flow-based orthogonal representations are proposed to make the network better capture motion patterns and keep inter-frame consistency. Extensive experiments have verified the superiority of our method and the effectiveness of each component.

References

- [1] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *Int'l Journal of Computer Vision*, 86(2-3):256–274, 2010. 2, 3
- [2] J. Bossu, N. Hautière, and J.-P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International journal of computer vision*, 93(3):348–367, 2011. 2, 3
- [3] N. Brewer and N. Liu. Using the shape characteristics of rain to identify and remove rain from video. In *Joint IAPR International Workshops on SPR and SSPR*, pages 451–458, 2008. 2, 3
- [4] B. Cai, X. Xu, and D. Tao. Real-time video dehazing based on spatio-temporal mrf. In *Pacific Rim Conference on Multimedia*, pages 315–325. Springer, 2016. 7
- [5] Y. Chang, L. Yan, and S. Zhong. Transformed low-rank model for line pattern noise removal. In *Proc. IEEE Int'l Conf. Computer Vision*, Oct 2017. 2
- [6] J. Chen and L. P. Chau. A rain pixel recovery algorithm for videos with highly dynamic scenes. *IEEE Trans. on Image Processing*, 23(3):1097–1104, March 2014. 2, 3
- [7] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2018. 1, 2, 3, 6
- [8] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1968–1975, 2013. 2, 3
- [9] C. Dong, Y. Deng, C. C. Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 576–584, Dec 2015. 2
- [10] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(2):295–307, Feb 2016. 2
- [11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *Proc. IEEE Int'l Conf. Computer Vision*, 2015. 5
- [12] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proc. IEEE Int'l Conf. Computer Vision*, December 2013. 2
- [13] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. on Image Processing*, 26(6):2944–2956, June 2017. 2
- [14] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, July 2017. 2, 3, 6
- [15] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 1715–1723, July 2017. 2
- [16] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, volume 1, pages 1–528, 2004. 2
- [17] K. Garg and S. K. Nayar. When does a camera see rain? In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 1067–1074, 2005. 2
- [18] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM Trans. Graphics*, volume 25, pages 996–1002, 2006. 2
- [19] K. Garg and S. K. Nayar. Vision and rain. *Int'l Journal of Computer Vision*, 75(1):3–27, 2007. 2
- [20] Y. Hu, W. Yang, S. Xia, W. Cheng, and J. Liu. Enhanced intra prediction with recurrent neural network in video coding. In *Proc. Data Compression Conference*, pages 413–413, March 2018. 2
- [21] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on multimedia*, 16(1):83–93, 2014. 1
- [22] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang. Context-aware single image rain removal. In *Proc. IEEE Int'l Conf. Multimedia and Expo*, pages 164–169, 2012. 3
- [23] Q. Huynh-Thu and M. Ghanbari. Scope of validity of p-psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 7
- [24] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, July 2017. 1, 2, 3, 6
- [25] L. W. Kang, C. W. Lin, and Y. H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. on Image Processing*, 21(4):1742–1755, April 2012. 1, 2
- [26] J. H. Kim, C. Lee, J. Y. Sim, and C. S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *Proc. IEEE Int'l Conf. Image Processing*, pages 914–917, Sept 2013. 2
- [27] J. H. Kim, J. Y. Sim, and C. S. Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Trans. on Image Processing*, 24(9):2658–2670, Sept 2015. 2, 3, 6
- [28] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. End-to-end united video dehazing and detection. In *Proc. AAAI Conf. on Artificial Intelligence*, Feb. 2018. 7
- [29] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2018. 2, 3
- [30] R. Li, L.-F. Cheong, and R. T. Tan. Single Image Deraining using Scale-Aware Multi-Stage Recurrent Network. *ArXiv e-prints*, December 2017. 3
- [31] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 2736–2744, 2016. 2, 3, 6

- [32] J. Liu, S. Xia, W. Yang, M. Li, and D. Liu. One-for-all: Grouped variation network-based fractional interpolation in video coding. *IEEE Trans. on Image Processing*, 28(5):2140–2151, May 2019. [2](#)
- [33] J. Liu, W. Yang, S. Yang, and Z. Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2018. [2](#), [3](#), [6](#), [7](#)
- [34] J. Liu, W. Yang, S. Yang, and Z. Guo. D3r-net: Dynamic routing residue recurrent network for video rain removal. *IEEE Trans. on Image Processing*, 28(2):699–712, Feb 2019. [2](#)
- [35] P. Liu, J. Xu, J. Liu, and X. Tang. Pixel based temporal analysis using chromatic property for removing rain from videos. In *Computer and Information Science*, 2009. [3](#)
- [36] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 3397–3405, 2015. [1](#), [2](#), [3](#), [6](#)
- [37] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 2482–2491, June 2018. [2](#)
- [38] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *Proc. IEEE European Conf. Computer Vision*, 2016. [5](#)
- [39] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang. Video desnowing and deraining based on matrix decomposition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, July 2017. [3](#)
- [40] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. *Int'l Journal of Computer Vision*, 112(1):71–89, March 2015. [3](#)
- [41] S. Starik and M. Werman. Simulation of rain in videos. In *Texture Workshop, ICCV*, June 2003. [3](#)
- [42] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang. Exploiting image structural similarity for single image rain removal. In *Proc. IEEE Int'l Conf. Image Processing*, pages 4482–4486, 2014. [1](#)
- [43] A. K. Tripathi and S. Mukhopadhyay. A probabilistic approach for detection and removal of rain from videos. *IETE Journal of Research*, 57(1):82–91, 2011. [3](#)
- [44] A. K. Tripathi and S. Mukhopadhyay. Video post processing: low-latency spatiotemporal approach for detection and removal of rain. *IET Image Processing*, 6(2):181–196, March 2012. [3](#)
- [45] C. Wang, C. Xu, C. Wang, and D. Tao. Perceptual adversarial networks for image-to-image transformation. *IEEE Trans. on Image Processing*, 27(8):4066–4079, Aug 2018. [2](#)
- [46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13(4):600–612, 2004. [7](#)
- [47] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Proc. IEEE Int'l Conf. Computer Vision*, Oct 2017. [3](#), [6](#)
- [48] S. Xia, W. Yang, Y. Hu, S. Ma, and J. Liu. A group variational transformation neural network for fractional interpolation of video coding. In *Proc. Data Compression Conference*, pages 127–136, March 2018. [2](#)
- [49] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video Enhancement with Task-Oriented Flow. *ArXiv e-prints*, November 2017. [6](#)
- [50] W. Yang, J. Feng, G. Xie, J. Liu, Z. Guo, and S. Yan. Video super-resolution based on spatial-temporal recurrent residual networks. *Computer Vision and Image Understanding*, 168:79 – 92, 2018. Special Issue on Vision and Computational Photography and Graphics. [2](#)
- [51] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan. Deep edge guided recurrent residual learning for image super-resolution. *IEEE Trans. on Image Processing*, 26(12):5895–5907, Dec 2017. [2](#)
- [52] W. Yang, J. Liu, S. Xia, and Z. Guo. Variation learning guided convolutional network for image interpolation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1652–1656, Sept 2017. [2](#)
- [53] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, July 2017. [2](#), [3](#), [6](#)
- [54] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. [2](#)
- [55] W. Yang, S. Xia, J. Liu, and Z. Guo. Reference guided deep super-resolution via manifold localized external compensation. *IEEE Trans. on Circuits and Systems for Video Technology*, pages 1–1, 2018. [2](#)
- [56] H. Zhang and V. M. Patel. Density-aware single image deraining using a multi-stream dense network. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2018. [2](#)
- [57] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017. [2](#)
- [58] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *Proc. IEEE Int'l Conf. Multimedia and Expo*, pages 461–464, 2006. [2](#), [3](#)
- [59] X. Zhang, W. Yang, Y. Hu, and J. Liu. DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *Proc. IEEE Int'l Conf. Image Processing*, pages 390–394, Oct 2018. [2](#)