Recurrent Multi-Frame Deraining: Combining Physics Guidance and Adversarial Learning

Wenhan Yang, *Member, IEEE*, Robby T. Tan, *Member, IEEE*, Jiashi Feng, *Member, IEEE*, Shiqi Wang, *Member, IEEE*, Bin Cheng, and Jiaying Liu, *Senior Member, IEEE*

Abstract—Existing video rain removal methods mainly focus on rain streak removal and are solely trained based on the synthetic data, which neglect more complex degradation factors, e.g., rain accumulation, and the prior knowledge in real rain data. Thus, in this paper, we build a more comprehensive rain model with several degradation factors and construct a novel two-stage video rain removal method that combines the power of synthetic videos and real data. Specifically, a novel two-stage progressive network is proposed: recovery guided by a physics model, and further restoration by adversarial learning. The first stage performs an inverse recovery process guided by our proposed rain model. An initially estimated background frame is obtained based on the input rain frame. The second stage employs adversarial learning to refine the result, i.e., recovering the overall color and illumination distributions of the frame, the background details that are failed to be recovered in the first stage, and removing the artifacts generated in the first stage. Furthermore, we also introduce a more comprehensive rain model that includes degradation factors, e.g., occlusion and rain accumulation, which appear in real scenes yet ignored by existing methods. This model, which generates more realistic rain images, will train and evaluate our models better. Extensive evaluations on synthetic and real videos show the effectiveness of our method in comparisons to the state-of-the-art methods. Our datasets, results and code are available at: https://github.com/flyywh/Recurrent-Multi-Frame-Deraining.

Index Terms—Multi-frame, video rain removal, physics recovery guidance, adversarial learning

1 Introduction

Rain degrades videos, causing outdoor computer vision systems to be erroneous, as most of them assume clear input videos. There are a few factors of rain degradation. *Rain streaks* lead to intensity changes in image content, obscuring the background and blurring the scene. Rain streaks can also completely occlude some background signals, where no background signals go through, a phenomenon we call *rain occlusion*. *Rain accumulation* (also known as rain veiling effect), where individual rain streaks and water particles accumulate forming visual effects similar to mist or fog, impair the background contrast, reducing the distant scenes' visibility significantly. When rainfall intensity in some period of time changes rapidly, rain accumulation can fluctuate over the period of time, which is visually like a flowing transparent veil covering the background. We call this phenomenon *accumulation flow*.

Many methods have been proposed to derain either images or videos. Single-image-based methods, e.g., [15], [20], [28], [34] employ some techniques, such as a frequency-domain representation [20], sparse representation [28], Gaussian mixture model [25] and deep networks [8], [42]. Video-based methods, e.g., [1], [2], [11], [46] make full use of both temporal and spatial information. Garg and Nayar [11] utilize the physics properties of rain, e.g., chromatic and direction. Kim *et al.* and Jiang *et al.* [19], [21] further exploit temporal dynamics, i.e., background motion's continuity, rain streaks' random occurrence, and motion cues.

Recently, deep-learning based methods have been proposed to tackle the video deraining problem. In [4], a rain image is first segmented into superpixels, then a consistency constraint is imposed on these aligned superpixels. Li *et al.* [23] propose a multiscale convolutional sparse coding-based video rain streak removal method. Liu *et al.* [26], [27] build a recurrent network to jointly integrate the tasks of rain degradation detection, background reconstruction and rain removal. In [18], [19], a tensor decomposition based deraining methods is proposed to fully consider the discriminative characteristics of clean backgrounds and rain streaks in the gradient domain. While these video deraining methods can be effective in some cases, they all are designed to handle only rain streak removal. Little attention is given to other factors of rain degradation, such as rain accumulation, despite their degradation in many cases is obviously visible. Moreover,

- *Wenhan Yang and Shiqi Wang are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China. E-mail: {wyang34, shiqwang}@cityu.edu.hk.*
- *Robby T. Tan is with the Yale-NUS College, Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077. E-mail: robby.tan@nus.edu.sg.*
- *Jiashi Feng is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077. E-mail: elefjia@nus.edu.sg.*
- *Bin Cheng is with the Machine Learning Group, Beijing Academy of Artificial Intelligence, Beijing 100081, China. E-mail: chengbin@baai.ac.cn.*
- *Jiaying Liu is with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China. E-mail: liujiaying@pku.edu.cn.*

Manuscript received 19 September 2020; revised 29 March 2021; accepted 16 May 2021. Date of publication 24 May 2021; date of current version 3 October 2022.
(Corresponding author: Jiaying Liu.)
Recommended for acceptance by C. C. Loy.
Digital Object Identifier no. 10.1109/TPAMI.2021.3083076

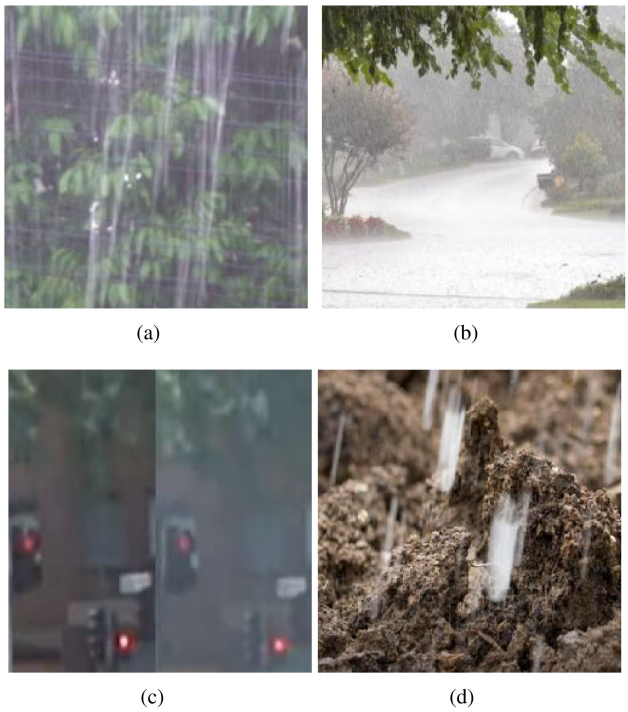


Fig. 1. Visibility degradation caused by rain. (a) Rain streaks. (b) Rain accumulation. (c) Rain accumulation flow. The atmosphere flow makes veiling layers' densities at the same pixel of two frames different. (d) Rain occlusion. There is an identical intensity in the occlusion regions.

how to make a full use of inter-frame and intra-frame contexts to promote joint estimation of multiple rain-related factors has not been fully explored.

Our goal in this paper is to handle video deraining in a more comprehensive way by fully considering the rain-related factors: rain streak, rain accumulation, accumulation flow, and rain occlusion as illustrated in Fig. 1. To achieve the goal, we introduce a new rain model to synthesize more visually realistic effects of various factors, i.e., rain streak, rain occlusion, rain accumulation, and accumulation flow. We also design a two-stage progressive network, which combines the rain model as well as both physics and natural video priors. In the first stage, a rain-free frame is recovered, which is followed by the inverse process based on our rain model. Subsequently, with the help of the previously recovered clean frames and the initial estimation, in the second stage, a more accurate estimation is inferred using adversarial learning.

Our contributions can be summarized as follows:

- A new rain model is proposed. Beyond existing video rain models, it captures rain degradation factors comprehensively, i.e., rain accumulation, accumulation flow, rain streaks, and rain occlusion, providing more realistic modeling of rain scenes. Based on the model, a novel rain video dataset is synthesized to support the development and evaluation of learning-based video rain removal methods in heavy rain.
- To make full use of the spatial and temporal contexts in rain scenes, a convolutional LSTM network is introduced to our deraining network. In the network, the inverse recovery module (physics network) is embedded. The rain-related variables are predicted. Then, the physics network estimates the rain-free frame based on the rain-related variables. This design takes advantages of the prior of the rain model and brings in a more effective architecture.
- Our proposed LSTM network has a two-stage design, which makes the first attempt to utilize the knowledge of both rain model and adversarial learning for video deraining. The first stage provides the physics accurate results and the second stage, where the results are further processed by the generator trained via the adversarial learning, adjusts the color and contrast distributions, correct details and remove artifacts.

This paper is an extension of [41], where we make further significant improvements: 1) In [41], for our synthetic data, the transmission used to generate the accumulation is constant within a frame. In this work, we change it to pixel-wise adaptive. The detail is illustrated in Section 5-Datasets. 2) We introduce the information of multiple frames in our two-stage progressive learning framework. In our current version, the models take five successive frames as their input in each stage, which is demonstrated to largely outperform the previous conference version. 3) To further utilize the prior knowledge of natural images, we apply the adversarial learning in the second stage of refinement network, to adjust the color and contrast as well as to correct the details and remove the artifacts generated in the physics recovery process. Extensive experiments demonstrate that, with the above-mentioned contributions, our model outperforms previous methods (including our conference version) quantitatively and qualitatively.

The rest of our paper is organized as follows. Section 2 illustrates the related work briefly. Section 3 presents our proposed comprehensive rain synthesis model. Section 4 proposes our recurrent video deraining network in details. In Section 5, experimental configurations and results are presented. The concluding remarks are provided in Section 6.

2 Related Work

2.1 Single-Image Rain Removal

Single image rain removal is an ill-posed task. To handle the ill-possessedness, different models and priors are utilized to separate the normal texture and rain signal from rain images. These models consist of sparse coding [20], Gaussian mixture model [25], discriminative sparse coding [28], rain direction prior [45], bilayer optimization [47], joint convolution analysis and synthesis sparse representation model [12]. The advent of deep networks promote the fast evolution of the rain removal from single images. In [7], [8], deep detail networks are constructed to infer the negative residue according to the information of the extracted high-frequency details of the rain images.

Yang *et al.* [42] developed deep networks to detect and remove rain streak in a joint manner, and to recurrent remove the rain streaks and accumulation. In [24], in order to handle the rain streaks having different sizes, Li *et al.* built several parallel sub-networks to generate the intermediate results, and after that, intermediate results are integrated into the final result. In [45], a new multi-stream density-aware densely connected CNN is built to estimate

rain density and remove rain streaks sequentially. In [30], a progressive recurrent network is constructed, which is incorporated with gate functions and recurrent units to capture the deep features' inter-stage dependencies to remove rain streak. In [9], inspired by Gaussian Laplacian pyramid decomposition, the network is designed to perform operations on the decomposition result, which makes the deraining process more efficiently and the model learning easier.

Yasarla *et al.* [44] proposed a network to extract rain-related content first at different scales and the corresponding confidence measure, which later on guides the successive rain removal process. In [13], Hu *et al.* developed a deep network for obtaining the depth-attentional features to estimate a residual signal and restore a clean background one. In [37], a spatial attentive network is built to remove rain streaks with the local-to-global attention guidance. Compared to the above-mentioned single-image rain removal work, only relying on exploiting spatial correlation, in our work, video rain removal is focused on, where we exploit temporal and spatial correlation jointly for removing rain from videos.

2.2 Multi-Frame Rain/Haze Removal

Video rain removal can exploit the temporal information and motion context additionally. Garg and Nayar make the first attempt to build the rain model [11] and deal with rain removal problem [10]. The later works address the problem with more flexible and intrinsic modeling of rain streaks and backgrounds, i.e., the shape, orientation, and size of rain streaks [2], Fourier domain feature [1], temporal and chromatic properties [46], phase congruency features [32], and rain streaks' directional tendency [19]. Later on, data-driven methods emerge and brings new progress as well as improved modeling capacity.

In [35], [36], with the help of the temporal and spatial features, a Bayesian rain detector is developed. Wei *et al.* [39] made attempt to encode rain streaks as mixtures of Gaussian. The model can finely adapt to a wide kind of rain variations. Kim *et al.* [21] trained an SVM. The SVM can be used to re-estimate the roughly detected rain streak maps. In [31], a matrix decomposition model is designed. The model is utilized to classify rain streaks into dense and sparse streaks. In [4], a rain image is first segmented into superpixels. Then, the aligned superpixels are enforced by the consistency constraints. After that, the aligned superpixels are compensated for the the lost details. In [23], a multi-scale convolutional sparse coding approach is designed for video deraining. In [26], Liu *et al.* built a recurrent network to seamlessly integrate the multi-task of rain degradation detection, rain removal and background reconstruction. However, all of these previous methods do not pay attention to dealing with rain accumulation.

A series of works that focus on video haze removal provide meaningful insights to handle rain accumulation. Zhang *et al.* [51] estimated the scene depth jointly with the clear latent image, where the formulation models the depth cues from stereo matching and fog information in a mutually beneficial way. Cai *et al.* [52] built a Markov random field injected with intensity value prior to improve spatial consistency and temporal coherence for video dehazing. In [22], Li *et al.* conducted a thorough study over a number of network structure choices for the temporal fusion in the end-to-end learning context. Besides, the video dehazing and object detection are optimized jointly. In [50], Ren *et al.* build an end-to-end learnable deep network to gather information among adjacent frames to estimate the transmission.

In our work, we target at handling more kinds of visibility degradation based on the rain synthesis model we propose, i.e., rain streaks, accumulation, accumulation flow, occlusion. To better utilize inter-frame correlation, a two-step RNN is designed to fully make use of the knowledge of physics guidance and adversarial learning. The first stage provides the physics accurate results and then in the second stage, the results are further processed by the generator trained via the adversarial learning, to adjust the color and contrast distributions as well as to correct details and remove artifacts.

3 Comprehensive Rain Model

To handle video deraining issue, we develop a new comprehensive rain model. Using the model, we synthesize rain images from clean ones with the four degradation factors: rain streaks, rain accumulation, accumulation flow and rain occlusion. Rain streaks are the falling raindrops that form whitish streaks due to raindrops' rapid speed relative to the camera's exposure time. Their appearance occludes the background, as illustrated in Fig. 1a. In our rain-streak rendering, rain streaks are fused linearly with the clean background frames [8], [25], [42]. Rain accumulation occurs when the distant rain streaks together with water particles interweave, generating an atmospheric veiling effect [24], [42] where individual rain-streaks cannot be seen individually any more, as illustrated in Fig. 1b. In our rendering, we follow the physics model commonly used to generate fog [42].

In videos, rain accumulation can be dynamic due to wind or other atmospheric conditions. This dynamic rain accumulation over time form accumulation flow, which shown in Fig. 1c. Its transparency is independent from the depth of the background. It takes any shape, and produces a semi-transparent covering veil effect. Its existence is continuous temporally. In the synthesis process, we sample a nature gray image, blur it, and then adjust its intensity range globally to simulate the accumulation flow. At a given temporal step, we will randomly generate the motion vector of the accumulation flow at the moment and the flow is then set based on the vector to move in different temporal steps. The light transmittance of raindrops turns to be low in the heavy rain case. In this case, the additive rain model is not obeyed anymore, and the rain region is identical in intensity [26]. As illustrated in Fig. 1d, the background information is totally occluded. The occlusion image is rendered through an alpha matting process. Its generation is guided a binary mask, with a rain-contaminated image as well as the given intensity map to fuse.

In its basic form, our rain model follows the commonly used rain model for a single image [16], [25], [28]:

$$\mathbf{O} = \mathbf{B} + \mathbf{S}, \tag{1}$$

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

Fig. 2. Several example data based on our synthesis data produced by Eq. (5).

where $\mathbf{S}$ represents rain streaks, $\mathbf{B}$ represent the rain-free frame, and $\mathbf{O}$ is the image degraded by rain streaks. For video, we add a temporal indicator t:

$$\mathbf{O}_t = \mathbf{B}_t + \mathbf{S}_t,\ t = 1, 2, \ldots, N, \tag{2}$$

where N denotes the number of the video frames. $\mathbf{S}_t$, the rain streaks, are assumed to be independent and identically distributed [33]. Taking into account rain accumulation and accumulation flow, our model is expressed as:

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

where $\mathbf{A}_t$ represents the global atmospheric light, β_t represents atmospheric transmission that dependent on the depth of scene , and $\mathbf{U}_t$ denotes the rain accumulation flow layer that dependent on the atmospheric flow and local raindrop density. All these factors are temporally continuous. For a given fixed scene, $\{\mathbf{A}_t\}$ and $\{\alpha_t\}$ are affected only by the camera motions. $\{\mathbf{U}_t\}$ has its motion trajectory. Finally, similar to modeling rain occlusions in [26], [27]:

$$\mathbf{O}'_t = (1 - \boldsymbol{\alpha}_t)(\mathbf{B}_t + \mathbf{S}_t) + \boldsymbol{\alpha}_t \mathbf{M}_t, \tag{4}$$

where $\boldsymbol{\alpha}_t$ signifies an alpha matting map, and $\mathbf{M}_t$ is the rain reliance map, the rain model starting from Eq. (3) to describe rain occlusions is expressed as:

$$\widetilde{\mathbf{O}}_t = (1 - \boldsymbol{\alpha}_t)\mathbf{O}_t + \boldsymbol{\alpha}_t \mathbf{M}_t. \tag{5}$$

Therefore, we obtain a rain model that captures rain streaks, accumulation, accumulation flow, and occlusions in a comprehensive way.

With the guidance of our rain model (5), we can synthesize more realistic-looking rain videos compared to existing methods. Two rendered examples by our model are shown in Fig. 2. Based on the rain synthesis model, we build a novel video rain dataset. More details are discussed in Section 5. A summary of commonly used datasets in recent video rain removal works, including their included degradation factors, rain models, main features, and code links, are provided in Table 1. Most of previously adopted datasets (*TCLRM*, *Stochastic*, *MS-CSC*, *DIP*, *FastDeRain*, *MRF*, and *NTURain*) only consider rain streak degradation and takes the rain model in Eq. (2). *RainSynLight25* and *RainSynComplex25* additionally model rain occlusions and their related rain models turn Eq. (4). Our dataset is build based on Eq. (5) and four kinds of degradation factors are included.

4 Recurrent Video Deraining Network

Our method is based on a two-stage network that utilizes multi-frames to derain the input video progressively. The initially derained estimations are used as guidance on the refined deraining network, which extracts more effective features. The first stage of our method follows the inverse recovery process in Eq. (3) and Eq. (5). Our method makes use of both the prior knowledge of rain model via injecting physics network and nature image distributions by employing the adversarial learning. As rain models cannot totally simulate complex rain scenes, i.e., the complex real rain accumulation and illumination change after the degradation, therefore, we introduce an enhancement network that applies adversarial learning to adjust the derained results generated by the inverse recovery of a rain model.

4.1 Network Architecture

Our method consists of 3 main networks: the initial deraining network (Initial-DerainNet), inverse recovery network (PhysicsNet) refined deraining network (Refined-DerainNet), as illustrated in Fig. 3. In the first stage, Initial-DerainNet takes successive rain frames $\mathbf{O}_{t-2}, \mathbf{O}_{t-1}, \ldots, \mathbf{O}_{t+2}$ as its input and estimates the rain-related variables of the frame t ($\hat{\mathbf{h}}_t, \hat{\boldsymbol{\beta}}_t^i$, and $\hat{A}_t$), where $\hat{\mathbf{h}}_t$ aims to regress $\mathbf{O}_t - \mathbf{U}_t - \mathbf{S}_t$. PhysicsNet utilizes the predicted rain-related variables to estimate the initial background (rain-free) frame $\widehat{\mathbf{B}}_t^i$ with the help of Eqs. (3) and (5).

In the second stage, Refined-DerainNet takes the existing clean frames ($\widehat{\mathbf{B}}_{t-2}^f$, $\widehat{\mathbf{B}}_{t-1}^f$ and the initially estimated clean frame $\mathbf{B}_t^i$ from the first stage) as well as their corresponding rain frames $\mathbf{O}_{t+1}$ and $\mathbf{O}_{t+2}$ as input to predict the refined background frame $\widehat{\mathbf{B}}_t^f$. The adversarial learning is used to constrain the training of the Refined-DerainNet, i.e., the generation process of $\widehat{\mathbf{B}}_t^f$. We utilize multiples losses to jointly regularize the recovery of $\widehat{\mathbf{B}}_t^f$ to accurately predict background frames while keeping the generalization capacity of the models. Note that, compared to [41] we do not align the frames. We observed that, alignment cannot lead to performance gains in our new framework, which takes successive five frames as input. The exclusion of the alignment process reduces the network's complexity.

4.2 Initial Deraining Network

Initial-DerainNet's architecture is a U-Net [49] like network, as illustrated in Fig. 3. A few frames are fed into convolutional layers concurrently and transformed into features via multiple convolutional layers. For the intermediate layers, we down-sample the spatial resolutions of features (at the encoder side) and then up-sample them (at the decoder side). In the encoder part, 3D convolutions are used to change the resolution sizes and shrink the temporal step of the tensor stacked by the features of the input frames. The specific structure of the encoder is depicted in Fig. 4a. The input frames are rearranged into two sequences:

TABLE 1
Summary of Rain Synthetic Models in the Literature

Name	# Sequence	Degradation	Model	Main Features	Publication
TCLRM[1]	9 (Synthetic Test) 6 (Real Test)	Streak	Eq. (2)	The real testing sequences include 1 captured one and 5 movie clips. 3 of 9 synthetic sequences are captured with moving cameras, whereas 6 of 9 are captured with stationary ones.	Kim *et al.* 2015 [21]
Stochastic[2]	4 (Synthetic Test) 2 (Real Test)	Streak	Eq. (2)	Rain streaks, varied from tiny drizzling to heavy rainstorms, are added to four videos with static backgrounds.	Wei *et al.* 2017 [39]
MS-CSC[3]	3 (Synthetic Test) 3 (Real Test)	Streak	Eq. (2)	Different types of rain streaks are added to these videos, varying from tiny drizzling to heavy rainstorms and vertical rain to slash lines.	Li *et al.* 2018 [23]
DIP	6 (Synthetic Test) 2 (Real Test)	Streak	Eq. (2)	The synthesized rain videos include heavy and light synthetic rain.	Jiang *et al.* 2017 [19]
FastDeRain[4]	12 (Synthetic Test) 4 (Real Test)	Streak	Eq. (2)	12 video sequences are synthesized with 4 clean videos and 3 types of rain streaks.	Jiang *et al.* 2019 [18]
MRF	5 (Synthetic Test) 1 (Real Test)	Streak	Eq. (2)	Various rain and snow video sequences include illumination variations, camera motions, moving objects, *etc*.	Ren *et al.* 2017 [31]
NTURain[5]	25 (Synthetic Train) 8 (Synthetic Test) 7 (Real Test)	Streak	Eq. (2)	Three to four different rain appearances are synthesized over each video clip to provide us 25 rainy scenes. 8 testing scenes can be divided into two groups: one shot from a panning and unstable camera and the other from a fast-moving camera.	Liu *et al.* 2018 [4]
RainSynLight25[6]	190 (Synthetic Train) 25 (Synthetic Test)	Streak, Occlusion	Eq. (4)	The dataset is synthesized by non-rain sequences with the rain streaks generated by the probabilistic model [11].	Liu *et al.* 2018 [26]
RainSynComplex25[6]	190 (Synthetic Train) 25 (Synthetic Test)	Streak, Occlusion	Eq. (4)	The dataset is synthesized by non-rain sequences with the rain streaks generated by the probabilistic model [11], sharp line streaks [42] and sparkle noises.	Liu *et al.* 2018 [26]
RainSynAll100	900 (Synthetic Train) 100 (Synthetic Test)	Streak, Occlusion, Rain Accumulation, Accumulation Flow	Eq. (5)	The dataset is generated by 1,000 clean sequences from the Vimeo-90K dataset [40] with the mentioned four kinds of degradations.	Our work

[1] *http://mcl.korea.ac.kr/deraining*
[2] *https://github.com/wwzjer/RainRemoval_ICCV2017*
[3] *https://github.com/MinghanLi/MS-CSC-Rain-Streak-Removal*
[4] *https://github.com/TaiXiangJiang/FastDeRain/blob/local/Data/data.md*
[5] *https://github.com/hotndy/SPAC-SupplementaryMaterials*
[6] *https://github.com/flyywh/J4RNet-Deep-Video-Deraining-CVPR-2018*

$$\mathbf{s}_1 = \mathbf{s}_2 = [\mathbf{O}_{t-2}, \mathbf{O}_{t-1}, \mathbf{O}_t, \mathbf{O}_{t+1}, \mathbf{O}_{t+2}], \tag{6}$$

which are processed by two sub-encoder as illustrated in Fig. 4b and the extracted features are summed together after the process as illustrated in Fig. 4a.

As shown in Fig. 3, we use the skip connections (red lines), which help the features produced by the shallow layers reach the decoder's counterpart layers. Initial-DerainNet generates three rain-related variables:

$$\hat{\mathbf{v}}_t^i = \left[\hat{\mathbf{h}}_t^i, \hat{\mathbf{A}}_t^i, \hat{\boldsymbol{\beta}}_t^i\right] = \mathbf{G}_{\mathrm{I}}(\mathbf{s}_1, \mathbf{s}_2), \tag{7}$$

where $\hat{\mathbf{h}}_t^i$, $\hat{\mathbf{A}}_t^i$, and $\hat{\boldsymbol{\beta}}_t^i$ are rain streak-free image (might including rain accumulation), atmospheric air light and transmission of the rain accumulation estimated by the first stage Initial-DerainNet. $\mathbf{G}_{\mathrm{I}}(\cdot)$ denotes the Initial-DerainNet process.

There are three decoders to decode the feature generated by the encoder to output $\hat{\mathbf{h}}_t^i$, $\hat{\mathbf{A}}_t^i$, and $\hat{\boldsymbol{\beta}}_t^i$, respectively. Note that, the estimations of $\hat{\mathbf{A}}_t^i$ and $\hat{\boldsymbol{\beta}}_t^i$ will influence each other, hence we make them share the same encoder. Due to $\hat{\mathbf{A}}_t^i$ is a global variable, there are no skip connections that bypassing the features from the encoder to the corresponding decoder side. A convolutional LSTM is used to feed-forward the information at the feature level across frames at the end of the convolutional layers at the beginning of the decoder side, as denoted in Fig. 3.

4.3 Physics Module

Given $\hat{\mathbf{h}}_t^i$, $\hat{\boldsymbol{\beta}}_t^i$, and $\hat{\mathbf{A}}_t^i$, we employ Eq. (3) to estimate the clean background frame $\hat{\mathbf{B}}_t^i$ with the guidance of a single frame rain input:

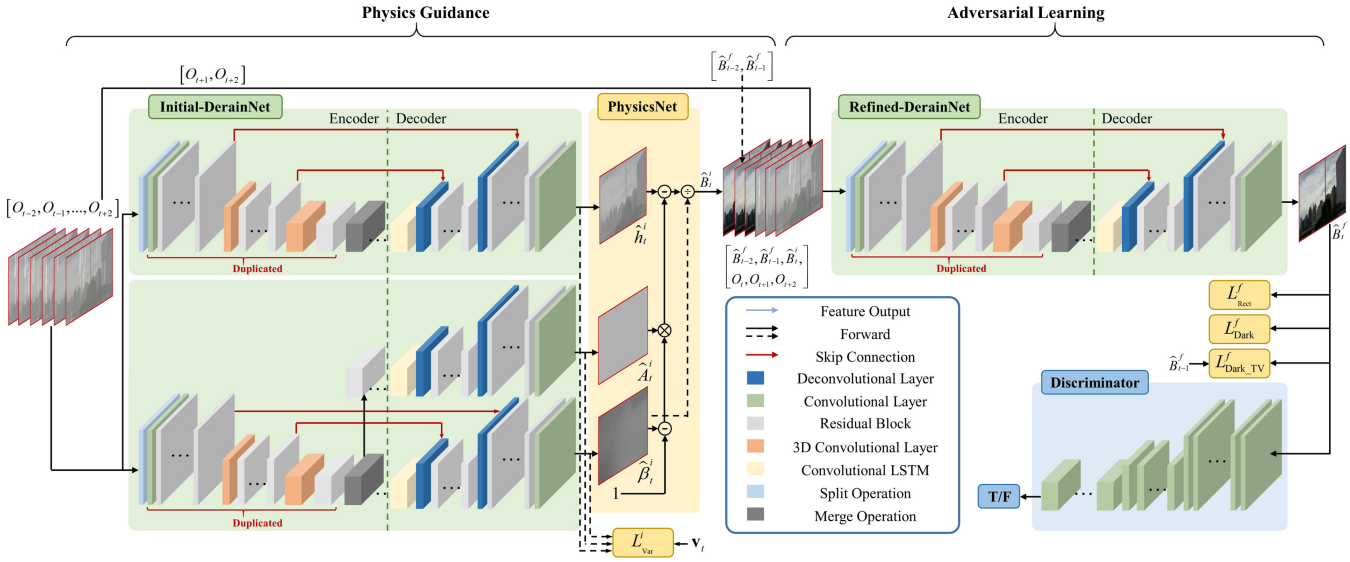


Fig. 3. Our two-stage progressive network framework for video rain removal. In the first stage, *Initial-DerainNet* uses successive rain frames $\mathbf{O}_{t-2}, \mathbf{O}_{t-1}, \ldots, \mathbf{O}_{t+2}$ as its input and outputs the estimation of the rain-related variables of the frame t. *Physics recovery module* translates these predicted rain-related variables into the initially estimated background frame $\mathbf{B}_t^i$ with the guidance of the inverse recovery in Eq. (3) and Eq. (5). In the second stage, *Refined-DerainNet* takes the existing clean frames ($\widehat{\mathbf{B}}_{t-2}^f$, $\widehat{\mathbf{B}}_{t-1}^f$ and the initially estimated clean frame $\widehat{\mathbf{B}}_t^i$ from the first stage) as well as their corresponding rainy frames $\mathbf{O}_{t+1}$, $\mathbf{O}_{t+2}$ as the network's input to directly predict the rain-free frames. We train the whole model in an end-to-end manner with the loss functions for variable estimation L_{Var}^f, and background frame refinement (reconstruction constraint L_{Rect}^f, adversarial learning L_{Dis}^f, and dark channel prior constraint L_{Dark}^f, $L_{\text{Dark\_TV}}^f$).

$$\hat{\mathbf{B}}_t^i = \frac{\hat{\mathbf{h}}_t - \left(1 - \hat{\boldsymbol{\beta}}_t^i\right) \times \hat{A}_t}{\max\left(\hat{\boldsymbol{\beta}}_t^i, \epsilon\right)}, \tag{8}$$

where ϵ signifies the threshold that helps guarantee the numerical stability, which in our experiments is set to 0.1. $\hat{\mathbf{h}}_t$ aims to regress $\mathbf{O}_t - \mathbf{U}_t - \mathbf{S}_t$. This module, which is injected to the whole network for an end-to-end training, makes full use of the prior of the physics model and brings in a more effective architecture.

4.4 Refined Deraining Network

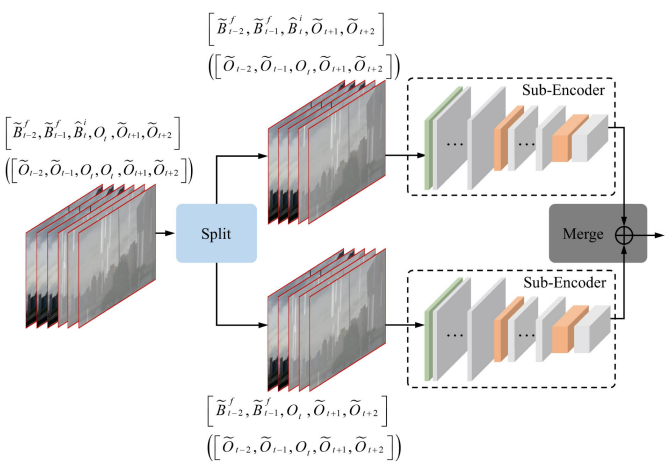
Having estimated the $(t-1)$th and $(t-2)$th rain-free background frames $\hat{\mathbf{B}}_{t-1}^f$ and $\hat{\mathbf{B}}_{t-2}^f$ as well as the initially estimated background of $\hat{\mathbf{B}}_t^i$ at time-step t, Refined-DerainNet takes them as input:

$$\mathbf{s}_1^{'} = \left[\hat{\mathbf{B}}_{t-2}^f, \hat{\mathbf{B}}_{t-1}^f, \hat{\mathbf{B}}_t^i, \mathbf{O}_{t-1}, \mathbf{O}_{t+1}\right] \tag{9}$$

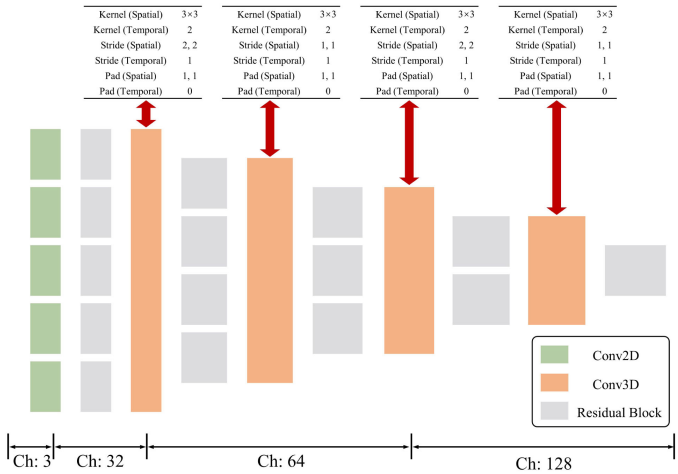
$$\mathbf{s}_2^{'} = \left[\hat{\mathbf{B}}_{t-2}^f, \hat{\mathbf{B}}_{t-1}^f, \mathbf{O}_t, \mathbf{O}_{t+1}, \mathbf{O}_{t+2},\right], \tag{10}$$

and directly predicts more refined rain-free background frames.

Refined-DerainNet has the same architecture as Initial-DerainNet. In the network, features are extracted from $\mathbf{s}_1^{'}$ and $\mathbf{s}_2^{'}$, respectively, and summed together at the bottleneck



(a) Encoder Architecture



(b) Sub-Encoder Architecture

Fig. 4. (a) The architecture of our encoder in Fig. 3. (b) The sub-encoder architecture that constitutes the encoder in (a).

of the encoder and decoder, which bridges the encoder and decoder and has the smallest spatial size. The network employs the skip connections as well as a convolutional LSTM. The convolutional LSTM is used to propagate the information at the feature level across frames, as denoted in Fig. 3, at the beginning of the decoder. We re-estimate the rain-free frames with a refinement network:

$$\hat{\mathbf{B}}_t^f = \mathbf{G}_{\mathrm{R}}\Big(\mathbf{s}_1^{'}, \mathbf{s}_2^{'}\Big), \tag{11}$$

where $\hat{\mathbf{B}}_t^f$ is the refined rain-free frame. $\mathbf{G}_{\mathrm{R}}(\cdot)$ denote the process of Refined-DerainNet.

4.4.1 Adversarial Learning

To check whether the output (the clean background) looks realistic and indeed clean, we employ a discriminator, using the following loss functions:

$$L_{\mathrm{Rect}}^f = -\mathrm{SSIM}\Big(\hat{\mathbf{B}}_t^f, \mathbf{B}_t\Big), \tag{12}$$

$$L_{\mathrm{Dis}}^f = -\log(\mathbf{G}_{\mathrm{D}}(\mathbf{B}_t)) - \log\Big(1 - \mathbf{G}_{\mathrm{D}}\Big(\hat{\mathbf{B}}_t^f\Big)\Big), \tag{13}$$

where $\mathbf{G}_{\mathrm{D}}(\cdot)$ is the discriminator, L_{Dis}^f is the adversarial loss, and $\mathbf{B}_t$ is the corresponding ground-truth. L_{Rect}^f is the reconstruction loss to keep the fidelity, and $\mathrm{SSIM}(\cdot)$ is the SSIM function.

We also intend to keep the refined images free from rain accumulation and continuous temporally. For this, we apply constraints based on the dark channels [48] of the refined frames:

$$L_{\mathrm{Dark}}^f = \Big\|\mathrm{Dark}\Big(\hat{\mathbf{B}}_t^f\Big)\Big\|_2^2, \tag{14}$$

$$L_{\mathrm{Dark\_TV}}^f = \Big\|\mathrm{Dark}\Big(\hat{\mathbf{B}}_t^f\Big) - \mathrm{Dark}\Big(\hat{\mathbf{B}}_{t-1}^f\Big)\Big\|_2^2, \tag{15}$$

where $\mathrm{Dark}(\cdot)$ is the function to calculate the dark channel of the input. The first term, the dark channel minimization, makes the refined frames more rain-accumulation free while the second term, the dark channel temporal total variation, makes the refined frame more temporally continuous.

4.4.2 Loss Functions

Our network is trained in an end-to-end manner. The loss functions are expressed as:

$$L_{\mathrm{all}} = L_{\mathrm{Var}}^i + \lambda_{\mathrm{Rect}} L_{\mathrm{Rect}}^f + \lambda_{\mathrm{Dis}} L_{\mathrm{Dis}}^f + \lambda_{\mathrm{Dark}} L_{\mathrm{Dark}}^f + \lambda_{\mathrm{Dark\_TV}} L_{\mathrm{Dark\_TV}}^f, \tag{16}$$

$$L_{\mathrm{Var}}^i = -\mathrm{SSIM}\Big(\hat{\mathbf{h}}_t^i, \mathbf{h}_t\Big) - \mathrm{SSIM}\big(\hat{\beta}_t^i, \beta_t\big) - \mathrm{SSIM}\big(\hat{\mathbf{A}}_t^i, \mathbf{A}_t\big), \tag{17}$$

where $\mathbf{h}_t$, β_t and $\mathbf{A}_t$ are the corresponding ground truths, and λ_{Rect}, λ_{Dis}, λ_{Dark} and $\lambda_{\mathrm{Dark\_TV}}$ are the weighting parameters.

5 Experimental Results

5.1 Datasets

Our Dataset Synthesis. Our dataset is synthesized based on Eq. (5). The synthesized videos come from two kinds of resources: 1) single images from OTS [54] with the depth information and sampled uniform motions; 2) real videos without the depth information.

In the first case, the transmission β_t is generated by:

$$\beta_t = \exp\Big(-s_t^\beta d_t / C_t^\beta\Big), \tag{18}$$

where s_t^β is sampled from a uniform distribution between [2.6, 4.6], d_t is the scene depth, and C_t^β is set to 10 empirically. The global atmospheric light $\mathbf{A}_t$ is sampled from a uniform distribution between [0.25, 0.95]. The motion vector of the frame is sampled from a uniform distribution between $\big[-\mathrm{B}_t^{\mathrm{MV}}, \mathrm{B}_t^{\mathrm{MV}}\big]$, where $\mathrm{B}_t^{\mathrm{MV}}$ is a quarter of the minimum of the frame's weight and height. The synthesized motions will be applied to crop different image regions to form a simulated video. The accumulation flow U_t is generated via the following process: we sample a nature gray image from BSD500 dataset [56], resize it to a very large one, crop and blur it, and then adjust its intensity range (with a random value sampled from the distribution between [0.1, 0.5]) globally to simulate the accumulation flow; the motion of the accumulation flow is selected from KITTI dataset [55] with a guided blurring operation; after that, this motion guides the accumulation flow to move among different time-steps. The rain streak $\mathbf{S}_t$ and occlusion-related variables $\alpha_t, \mathbf{M}_t$ are generated following the same procedure in [26], [27].

In the second case, the procedures are the same in most aspects. The differences lie in: 1) β_t is global variable sampled from [0.5, 1] instead of a pixel-wise one; 2) the background sequences are selected from the Vimeo-90K Dataset [40]; 3) the videos are ready-made, which are not synthesized from single images. More details about our dataset synthesis can be found on our website.

Finally, our dataset includes 900 training sequences and 100 validation sequences, where the sequence lengths of half of these sequences are 7 while those of the other half are 9.

Evaluation Datasets. The proposed method is compared with SOTA on widely used datasets. In [26], Liu *et al.* propose two datasets *RainSynComplex25* and *RainSynLight25* with respective heavy and light rain streaks. In [4], Chen *et al.* propose *NTURain*, which consists of two groups. One is captured by panning and unstable camera that has slow movements, while the other is taken from a car-mount fast moving camera. In this paper, we additionally propose *RainSynAll100*, which is generated by 500 clean sequences and 500 clean images with the mentioned four kinds of degradation factors. The whole dataset consists of testing and training datasets, including the respective 100 and 900 video sequences. We use *practical rain video sequences* selected from of movie clips and videos of Youtube website.

5.2 Evaluations

Baselines. Our MFGAN is compared with the following state-of-the-art (SOTA) methods: DetailNet [8], Discriminative Sparse Coding (DSC) [28], Joint Rain Detection and Removal

TABLE 2
Quantitative Evaluation of Different Rain Streak Removal Methods on *RainSynLight25*, *RainSynComplex25*, and *NTURain*

Metric	Dataset	TCLRM	DetailNet	MS-CSC	JORDER-E	SE	DSC	J4RNet	SpacCNN	FastDerain	CVPR-2019	Proposed
PSNR	*RainSynLight25*	28.77	26.00	25.58	30.37	26.56	25.63	32.96	32.78	29.42	35.80	36.99
SSIM		0.8693	0.9035	0.8089	0.9235	0.8006	0.8328	0.9434	0.9239	0.8683	0.9622	0.9760
PSNR	*RainSynComplex25*	17.31	23.53	16.96	20.20	16.76	17.33	24.13	21.21	19.25	27.72	32.70
SSIM		0.4956	0.7969	0.5049	0.6335	0.5293	0.5036	0.7163	0.5854	0.5385	0.8239	0.9357
PSNR	*NTURain*	29.98	-	27.31	32.61	25.73	29.20	32.14	33.11	30.32	36.05	38.92
SSIM		0.9199	-	0.7870	0.9482	0.7614	0.9137	0.9480	0.9474	0.9262	0.9676	0.9764

Best results are denoted in red and the second best results are denoted in blue.

(JORDER) [42], Progressive Recurrent Network (PReNet) [30], Uncertainty guided Multi-scale Residual Learning (UMRL) [44], Stochastic Encoding (SE) [39], Temporal Correlation and Low-Rank Matrix completion (TCLRM) [21], Discriminative Intrinsic Priors (DIP) [19], Joint Recurrent Rain Removal and Reconstruction (J4RNet) [26], FastDeRain [18], MultiScale Convolutional Sparse Coding (MS-CSC) [23], SuperPixel Alignment and Compensation CNN (SpacCNN) [4]. The code links of all compared methods are provided in Table 4. J4RNet, DetailNet, JORDER, MS-CSC, PReNet, and SpacCNN are built based on deep-learning. SE, FastDerain, TCLRM, MS-CSC, J4RNet, DIP, and SpacCNN are video rain removal methods. DSC, DetailNet, PReNet, JORDER, and UMRL are single image deraining methods. When we conduct evaluations on *RainSynAll100*, for the methods that do not handle rain accumulation, spatio-temporal MRF dehazing (MRF) [3] and End-to-end united Video Dehazing and detection Network (EVD-Net) [22] are taken for pre-processing or post-processing.

Implementation Details. For quantitative evaluation, we use *RainSynLight25*, *RainSynComplex25*, *NTURain*, our proposed *RainSynAll100*, and our collected real rain videos for evaluation. As demonstrated in Table 1, *NTURain* includes 25 paired videos for training and 8 for testing. Both *RainSynLight25* and *RainSynHeavy25* include 190 paired videos for training and 25 for testing, respectively. Our proposed *RainSynAll100* includes 900 paired videos for training and 100 for testing. For qualitative evaluation, we use the collected real rain videos that do not have the paired clean version. Our MFGAN is trained via two steps. In the first step, our model is first trained without using adversarial loss L^f_{Dis}, as well as dark channel prior related losses L^f_{Dark} and $L^f_{\text{Dark\_TV}}$. In the second step, all losses are used for training. The weighting parameters are set as follows: $\lambda_{\text{Rect}} = 1$, $\lambda_{\text{Dis}} = 0.001$, $\lambda_{\text{Dark}} = 0.01$ and $\lambda_{\text{Dark\_TV}} = 0.1$. Adam optimizer is used in the whole training process with the learning rate 1e-4 for both generator and discriminator of our MFGAN. The model is initialized by Kaiming Initialization [53]. Empirically, the initialized weight of the convolutional layer takes only 0.5 of the default value. All training videos are sampled and cropped into $128 \times 128 \times 5$ cubics with a batch size of 2. For the non-deep learning-based methods, including TCLRM, SE, DSC, UMRL, FastDeRain, JCAS, and DIP, the evaluation is directly performed based on the codes released by the authors. For DetailNet and SpacCNN, we use their released models. JORDER and J4RNet are retrained based on the gray version of the respective training set, following their original settings. J4RNet-E, J4RNet-P, CVPR-2019, and our proposed method are retrained with the respective training set when the evaluation is performed on different datasets. MS-CSC does not need training, as it is an optimization-based deep-learning method. In the quantitative evaluation, Structure Similarity Index (SSIM) [38] and Peak Signal-to-Noise Ratio (PSNR) [17] are used as the quality measures. We follow previous methods to compare the quantitative results in the luminance channel only, since the human visual system is more sensitive to the luminance channel compared to the chrominance ones.

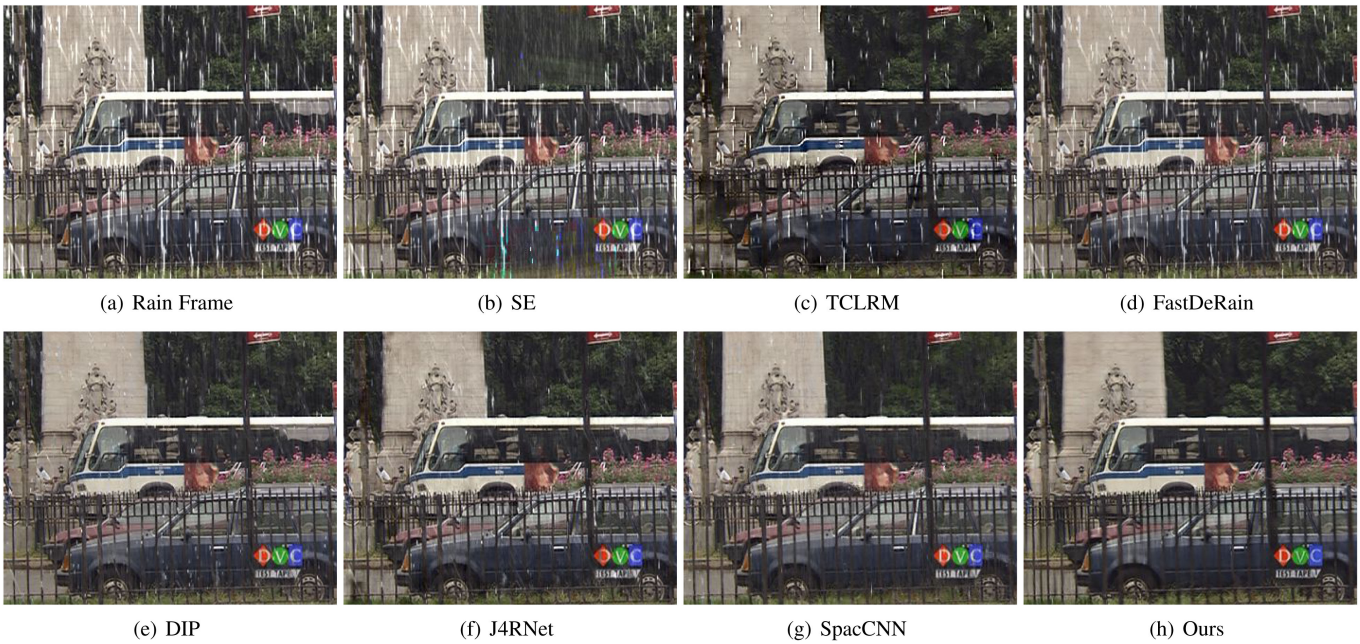
Quantitative Evaluation. In Table 2, our method is compared on the datasets with rain streak degradation only. Our method achieves better performance compared with previous methods. The proposed method obtains more than 7.5 dB and 4.0 dB PSNR gains on *RainSynComplex25* and *RainSynLight25* compared with J4RNet and SpacCNN. Compared to our CVPR-2019 results [41], Our method obtains almost 5.0 dB and 1.1dB PSNR gains on *RainSynComplex25* and *RainSynLight25*.

All methods are also evaluated on the proposed synthesized rain dataset *RainSynAll100*. SE, FastDerain, DIP,

TABLE 3
Quantitative Evaluation on *RainSynAll100*

Metric	FastDeRain+EVDNet	EVDNet+FastDeRain	DIP+EVDNet	EVDNet+DIP	SpacCNN+EVDNet	EVDNet+SpacCNN
PSNR	17.01	16.76	18.28	17.78	17.87	17.80
SSIM	0.5824	0.5794	0.6804	0.6485	0.6423	0.6379
Metric	FastDeRain+MRF	MRF+FastDeRain	DIP+MRF	MRF+DIP	SpacCNN+MRF	MRF+SpacCNN
PSNR	17.09	16.95	18.79	18.52	18.39	18.16
SSIM	0.5772	0.568	0.6914	0.6733	0.6469	0.6298
Metric	MS-CSC+EVDNet	EVDNet+MS-CSC	SE+EVDNet	EVDNet+SE	J4RNet-E	CVPR-2019
PSNR	16.27	16.02	15.43	16.90	18.93	21.06
SSIM	0.5188	0.5170	0.5186	0.5735	0.6749	0.7405
Metric	MS-CSC+MRF	MRF+MS-CSC	SE+MRF	MRF+SE	J4RNet-P	Ours
PSNR	16.19	16.06	15.2961	16.94	19.26	25.14
SSIM	0.5078	0.4990	0.5053	0.5344	0.6238	0.9172

ST-MRF and EVD-Net as used as pre/post-processing. Best results are denoted in red and the second best results are denoted in blue.



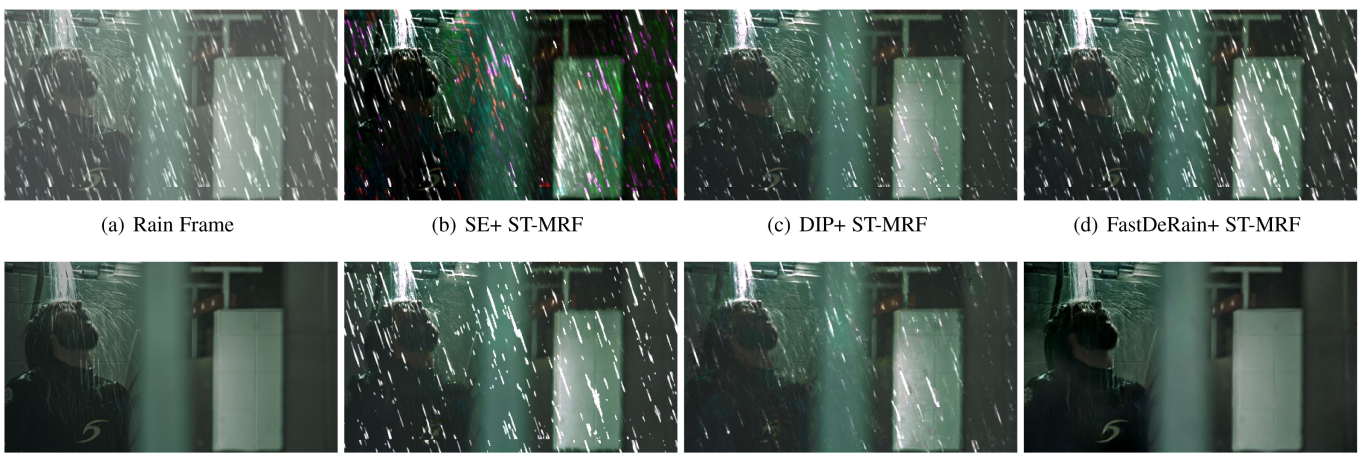
(a) Rain Frame (b) SE (c) TCLRM (d) FastDeRain

(e) DIP (f) J4RNet (g) SpacCNN (h) Ours

Fig. 5. Results of rain removal methods on a video frame from the synthesized dataset *RainSynComplex25*.

SpacCNN and MS-CSC are combined with two SOTA video defogging methods: EVD-Net [22] and ST-MRF [3] (post/pre-processing of rain accumulation removal) for a fair comparison. In Table 3, MRF + DIP uses the sequential combination of ST-MRF and DIP. DIP + MRF takes ST-MRF as post-processing. EVD-Net + DIP and DIP+ EVD-Net employ EVD-Net as pre/post-processing, respectively. The same applies to other SOTA methods. J4RNet-E uses the method in [26] to predict the background frame based on the input rain frame directly. J4RNet-P injects the inverse recovery module to predict the rain-related variables first and then estimate the background frame accordingly based on the predicted variables. As is illustrated in Table 3, our method rank the first among all methods. The performance gain is almost 6.0 dB in PSNR and 0.3 in SSIM. Compared to our previous CVPR-2019 results [41], our method obtains more than 0.170 and 4dB gains in SSIM and PSNR, respectively.

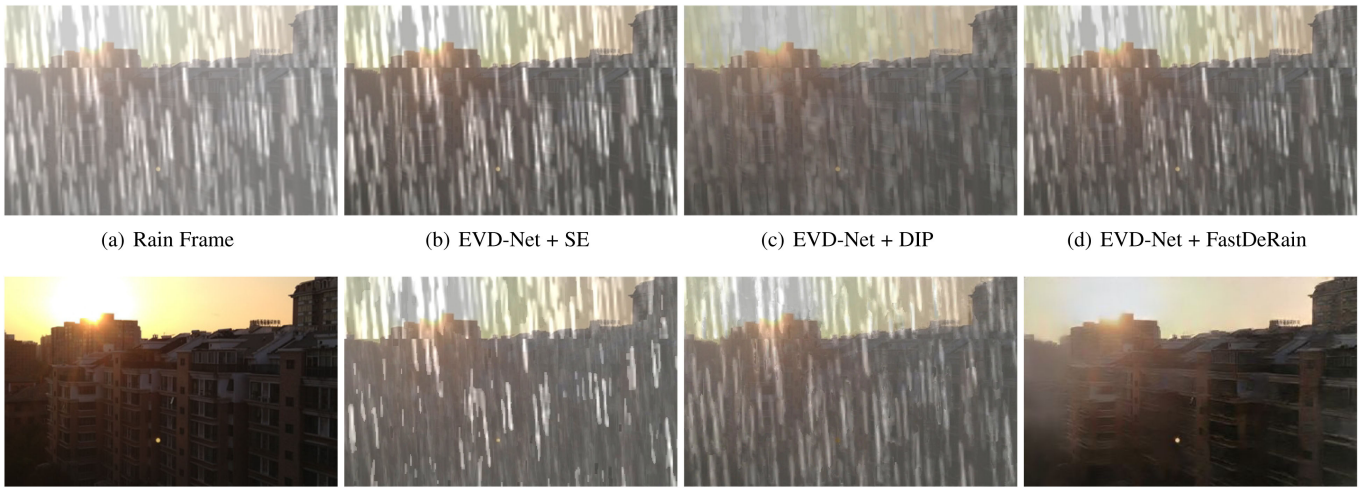
Qualitative Evaluation. Visual results of different deraining methods are also compared in Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16. Figs. 5, 6, and 7 show the results on synthetic rain videos. It is illustrated that, our method performs better than other SOTA methods on the synthetic data. Our method also obtains better results than other SOTA methods on real images (Figs. 9-7). For Figs. 9, 10, 11, 12, 13, and 14, all methods only apply rain streak removal. It is shown that, the proposed method method is better to remove most large (Figs. 8 and 9) and small rain streaks (Fig. 10). For accumulation, our method restores the best results in Figs. 14, 15, and 16. Note that, For Fig. 15, other methods apply EVD-Net as post-processing. Comparatively, our method is more successful to remove rain accumulation. For Fig. 16, other methods apply ST-MRF as pre-processing. The results of other methods are over-exposed. Comparatively, our method obtains naturally looking results.



(a) Rain Frame (b) SE+ ST-MRF (c) DIP+ ST-MRF (d) FastDeRain+ ST-MRF

(e) GT (f) MS-CSC+ ST-MRF (g) SpacCNN+ ST-MRF (h) Ours

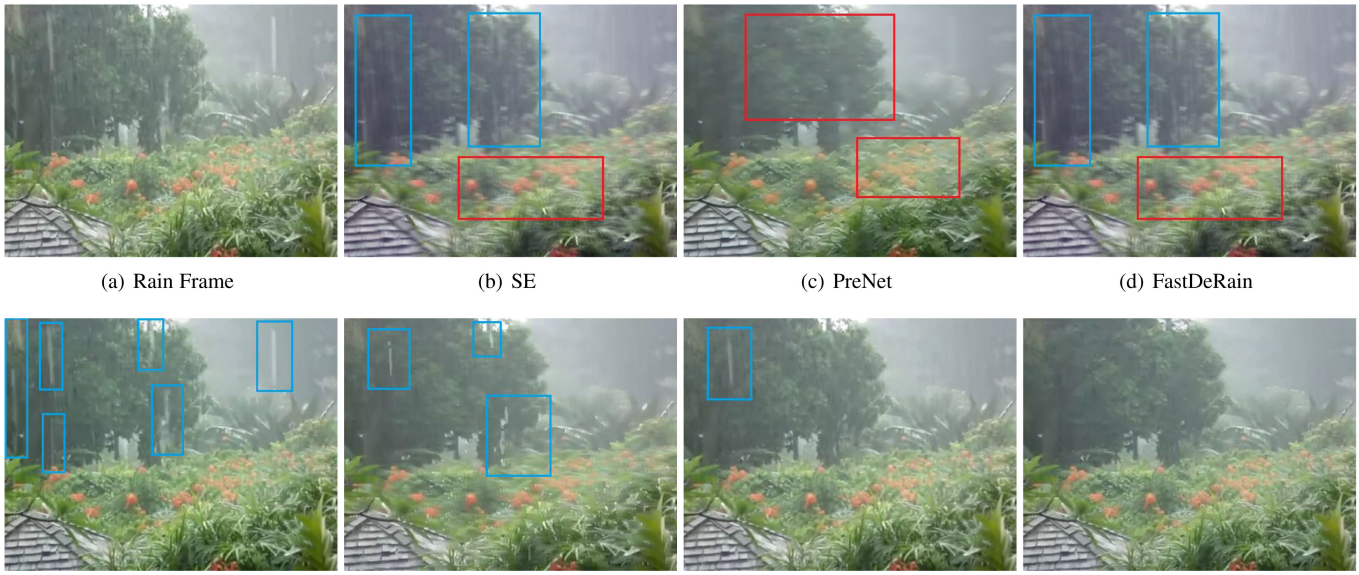
Fig. 6. Results of rain removal methods on a video frame from the synthesized dataset *RainSynAll100*. Except for our method, other methods apply ST-MRF as post-processing.



(a) Rain Frame (b) EVD-Net + SE (c) EVD-Net + DIP (d) EVD-Net + FastDeRain

(e) GT (f) EVD-Net + MS-CSC (g) EVD-Net + SpacCNN (h) Ours

Fig. 7. Results of rain removal methods on a video frame from the synthesized dataset *RainSynAll100*. Except for our method, other methods apply EVD-Net as pre-processing.



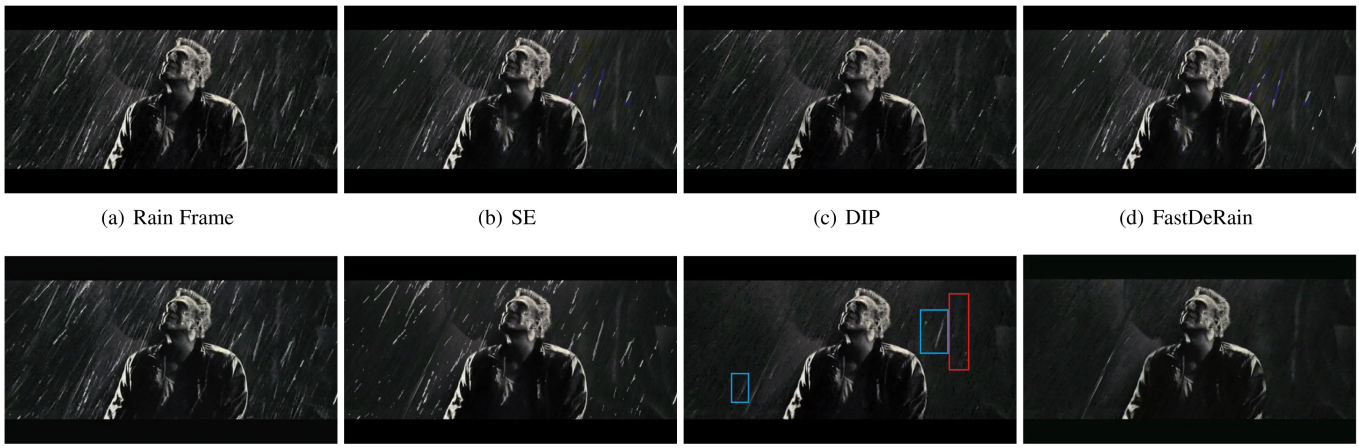
(a) Rain Frame (b) SE (c) PreNet (d) FastDeRain

(e) UMRL (f) MS-CSC (g) SpacCNN (h) Ours

Fig. 8. Results of rain streak removal by different methods on a real video frame. The results of SE, PreNet, FastDeRain, UMRL, and MS-CSC have remaining rain streaks, as denoted in blue boxes. Meanwhile, SE, PreNet and FastDeRain also falsely remove some details, as denoted by red boxes. Comparatively, our method can well handle the rain streaks.

Complexity Comparison. In Table 5, we compare the runtime of several SOTA methods. The testing video's resolution is 832×512. The proposed method, J4RNet-E, and J4RNet-P method are implemented by Pytorch. Other SOTA methods are implemented by MATLAB. DetailNet and SpacCNN are built based on MatConvNet.[1] JORDER is built on the Caffe's Matlab wrapper.[2] TCLRM is built based on CPU while other methods are GPU-based methods. Generally, the running speed of the proposed method is on par with other SOTA methods. Note that, compared with our previous work [41], our this work only needs almost a half running time. We also compare the parameters of different deraining methods in Table 6. In general, the parameters of our light-weighted model (Ours-S used for only rain streaks in Table 2) are on par with those of JORDER, CVPR-2019 and SpacCNN. However, Ours-S achieves much better quantitative results on the three datasets in Table 2. Our full model (Ours-L) includes many more parameters and leads to significantly superior performance as demonstrated in the quantitative results of Table 3 and the visual results in Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, and 17. It is noted that, Ours-S and Ours-L have more parameters as they introduce 3D convolutions to aggregate temporal information at the feature level. The

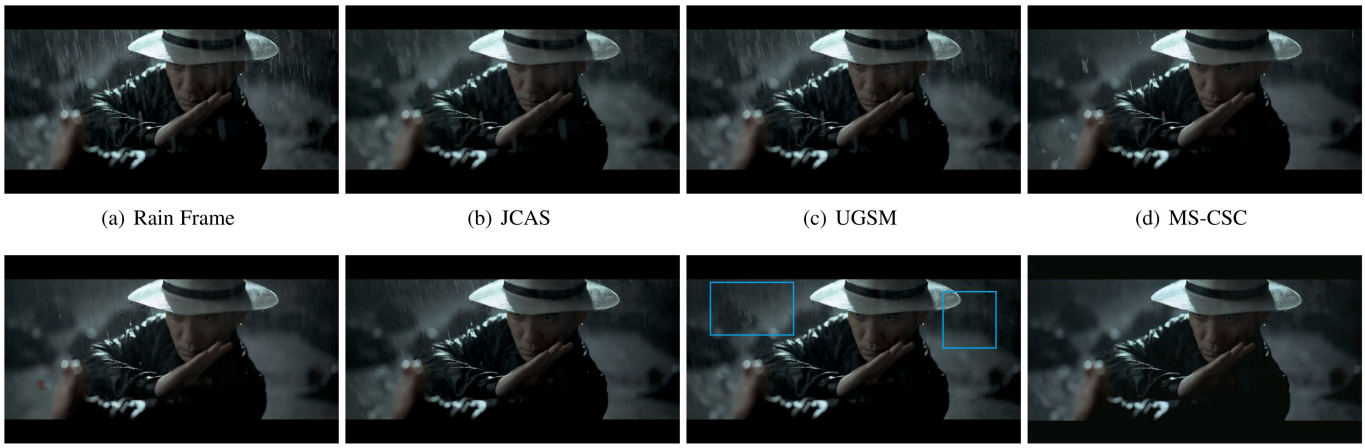
1. http://www.vlfeat.org/matconvnet/
2. http://caffe.berkeleyvision.org/



(a) Rain Frame (b) SE (c) DIP (d) FastDeRain

(e) UMRL (f) MS-CSC (g) SpacCNN (h) Ours

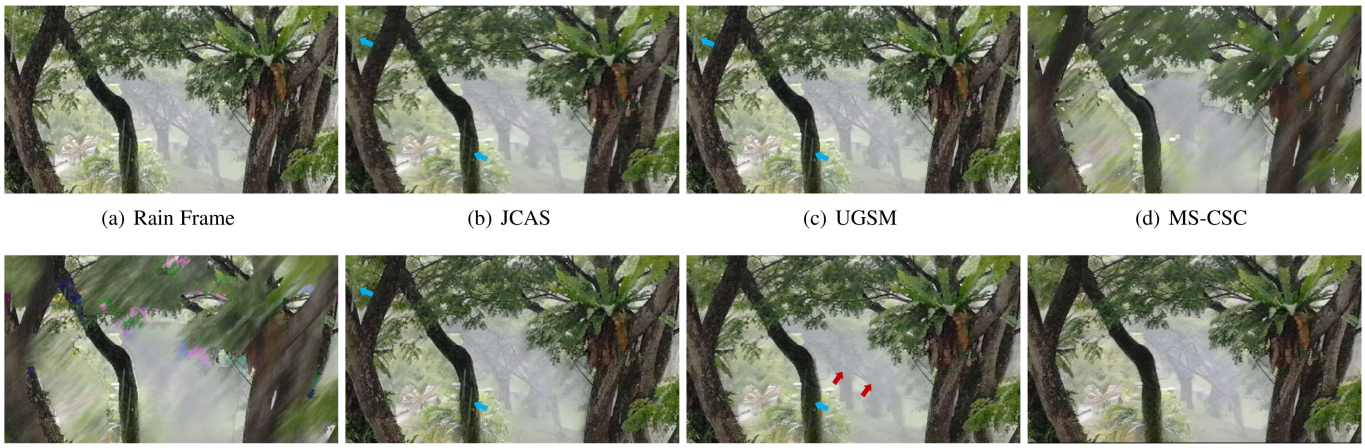
Fig. 9. Results of rain streak removal by different methods on a real video frame. The results of SE, DIP, FastDeRain, UMRL, and MS-CSC have obvious remaining rain streaks. For SpacCNN, there are small rain streaks in the regions denoted by blue boxes and details are falsely removed as denoted in the red box. Comparatively, our method can well handle the rain streaks.



(a) Rain Frame (b) JCAS (c) UGSM (d) MS-CSC

(e) SE (f) FastDeRain (g) SpacCNN (h) Ours

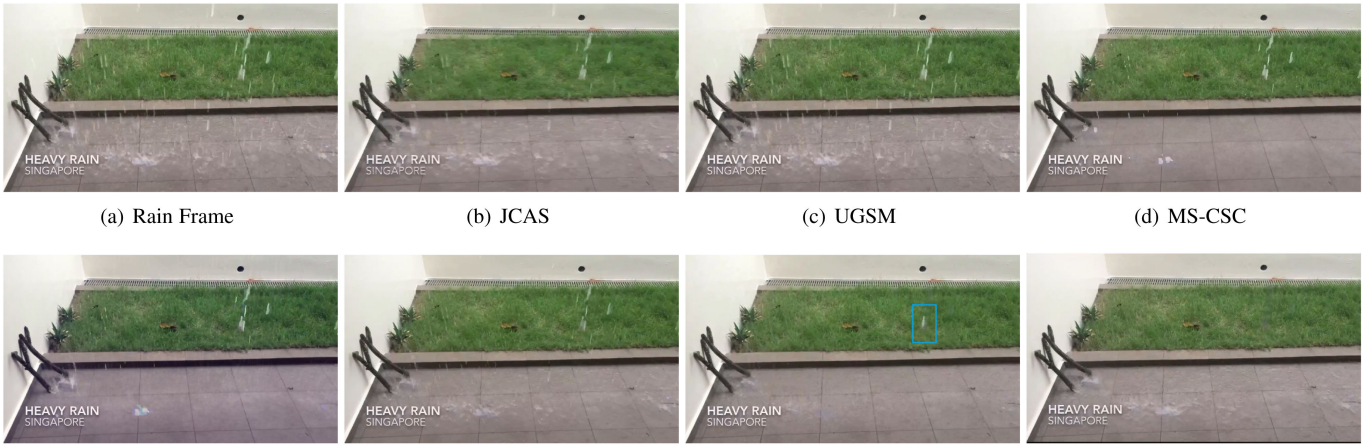
Fig. 10. Results of rain streak removal by different methods on a real video frame. The results of JCAS, UGSM, MS-CSC, SE, and FastDeRain have obvious remaining rain streaks. For SpacCNN, there are small rain streaks in the regions denoted by blue boxes. Comparatively, our method can well handle the rain streaks.



(a) Rain Frame (b) JCAS (c) UGSM (d) MS-CSC

(e) SE (f) FastDeRain (g) DIP (h) Ours

Fig. 11. Results of rain streak removal by different methods on a real video frame. The results of JCAS, UGSM, and FastDeRain have obvious remaining rain streaks, as denoted by blue arrows. As the video clip includes large camera motions, the results of MS-CSC and SE are totally damaged. For DIP, there are remaining rain streaks denoted by blue arrows and the details are blurred in the regions denoted by red arrows. Comparatively, our method can well handle the rain streaks and preserve structure details.



(a) Rain Frame (b) JCAS (c) UGSM (d) MS-CSC

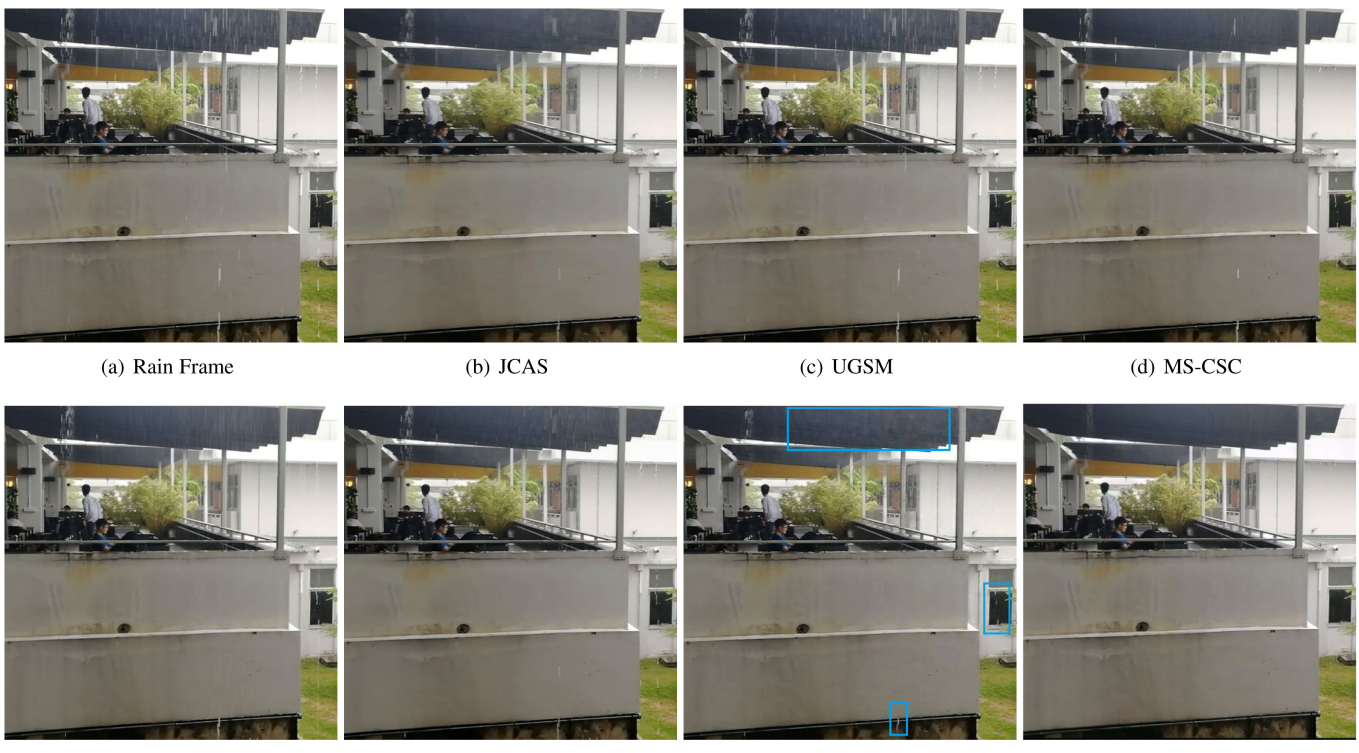
(e) SE (f) FastDeRain (g) SpacCNN (h) Ours

Fig. 12. Results of rain streak removal by different methods on a real video frame. The results of JCAS, UGSM, MS-CSC, SE, and FastDeRain have obvious remaining rain streaks as the rain streaks are too dense in the frame. SE additionally suffers from the color cast. SpacCNN still has remaining rain streaks as denoted by the blue box. Comparatively, our method can successfully remove most rain streaks.

increased parameters do not certainly lead to an increased computational complexity as the 2D convolutions in other models will be reused many times when dealing with multiple input frames at a certain time-step.

5.3 Ablation Studies

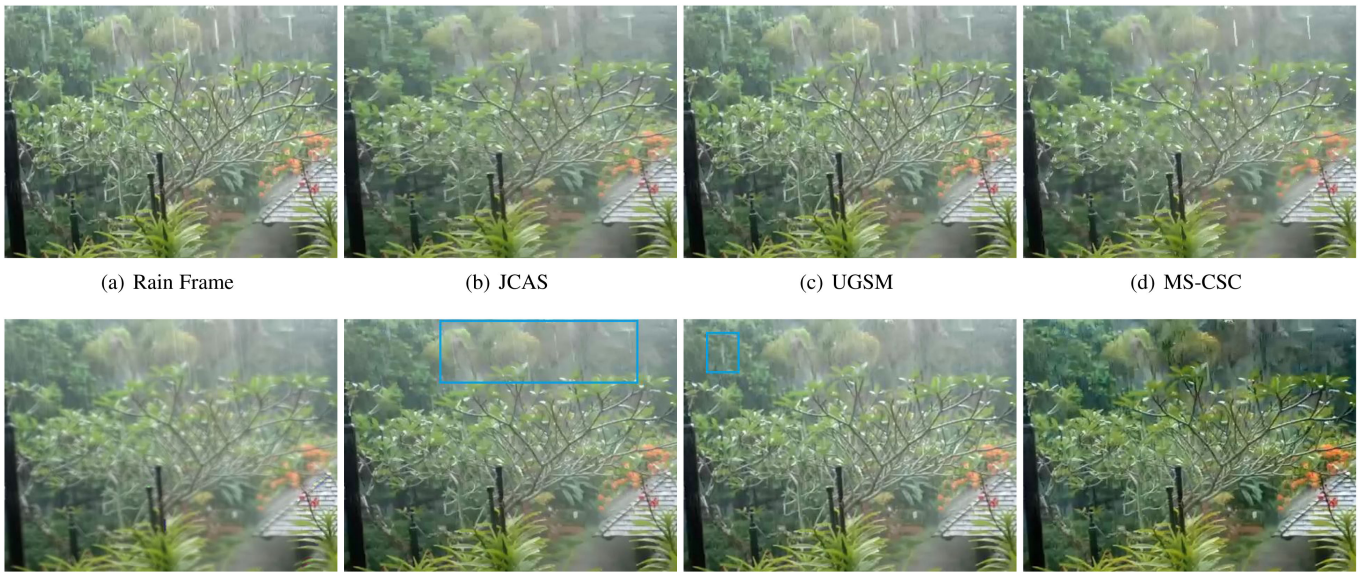
Ablation Study on Network Architecture. In Table 7, our methods with different components are evaluated. Table 7 shows that, the LSTM and SF-DerainNet improve the quantitative results significantly (v_1 versus v_4 and v_2 versus v_4). The rain streak removal can be benefited from the physics network, resulting a higher SSIM (v_3 versus v_4). Adding flow estimation and alignment cannot further improve the performance in our case (v_4 versus v_5). Therefore, v_4 is selected as our final version.



(a) Rain Frame (b) JCAS (c) UGSM (d) MS-CSC

(e) SE (f) FastDeRain (g) J4RNet (h) Ours

Fig. 13. Results of rain streak removal by different methods on a real video frame. It is clearly observed that, the results of JCAS, UGSM, MS-CSC, SE, and FastDeRain fail to remove the intensive rain streaks. J4RNet still has remaining rain streaks as denoted by blue boxes. Comparatively, our method can successfully remove most rain streaks.



(a) Rain Frame (b) JCAS (c) UGSM (d) MS-CSC

(e) SE (f) FastDeRain (g) SpacCNN (h) Ours

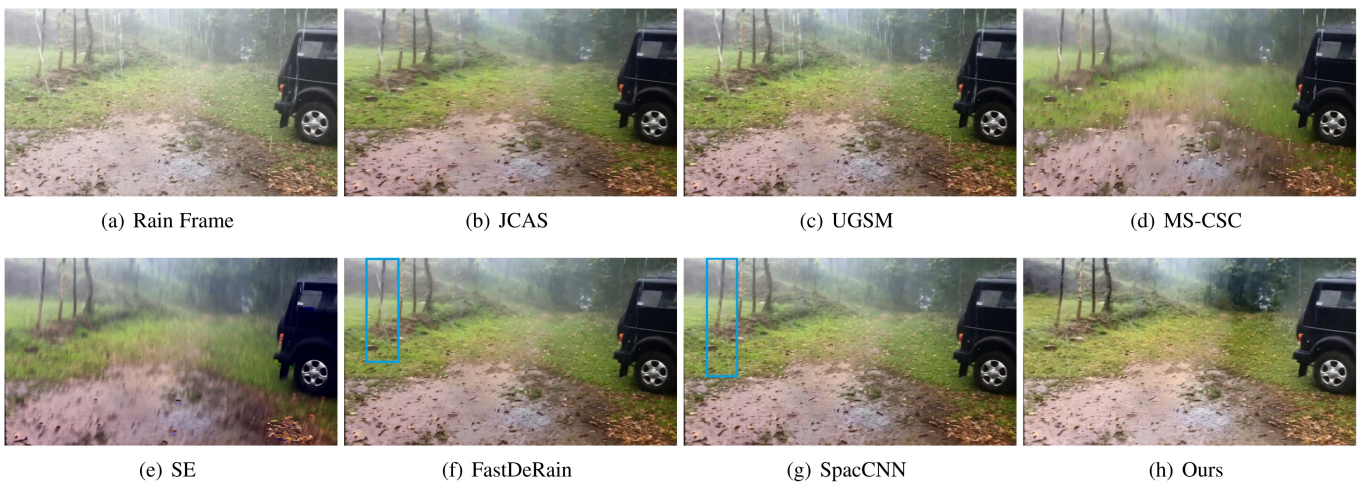
Fig. 14. Results of rain removal (rain streak and accumulation removal) by different methods on a real video frame. No method applies pre-processing or post-processing. The remaining rain streaks are denoted in blue boxes.

Visual Comparisons of Different Versions. We also compare our results with our previous results (CVPR-2019) [41] in Fig. 17. The figure shows our method is more successful in removing the rain streaks, as illustrated in the blue boxes of the top two panels in Fig. 17, and in removing rain accumulation, as illustrated in the blue boxed of the bottom two panels in Fig. 17.

Result of Versions with Different Parameters. We also show the performance of our methods with different parameters in Fig. 18. The figure shows that more parameters lead to higher performance, and with more parameters, the marginal performance gain is small.

6 Conclusion

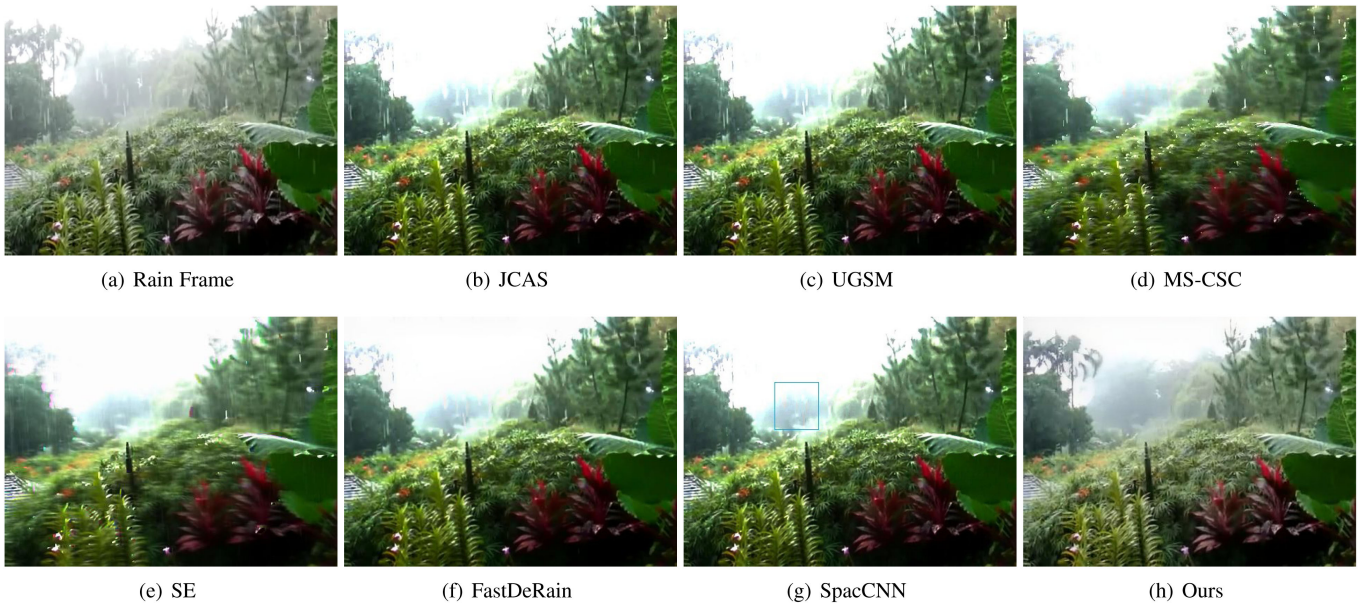
We introduced a video deraining method that consider more comprehensive degradation factors, i.e., accumulation, rain streak, accumulation flow and occlusion. To accomplish this, a new rain model is proposed, which can capture the factors, i.e., rain accumulation, accumulation flow, rain streaks, and rain occlusion. Based on the model, a novel rain video dataset is synthesized to support the development and evaluation of our deraining method. A recurrent neural network (RNN) is constructed, where the inverse recovery module can be injected. Our proposed two-stage RNN exploits the



(a) Rain Frame (b) JCAS (c) UGSM (d) MS-CSC

(e) SE (f) FastDeRain (g) SpacCNN (h) Ours

Fig. 15. Results of rain removal (rain streak and accumulation removal) by different methods on a real video frame. Except for our method, other methods apply EVD-Net as post-processing. The remaining rain streaks are denoted in blue boxes.



(a) Rain Frame (b) JCAS (c) UGSM (d) MS-CSC

(e) SE (f) FastDeRain (g) SpacCNN (h) Ours

Fig. 16. Results of rain removal (rain streak and accumulation removal) by different methods on a real video frame. Except for our method, other methods apply ST-MRF as pre-processing. The remaining rain streaks are denoted in blue boxes.

TABLE 4
Summary of Code Links for All Methods

Methods	Project Page
DetailNet	https://xueyangfu.github.io/projects/tip2017.html
DSC	http://www.math.nus.edu.sg/ matih/download/imaee_derainine/rain_removal_v.l.l.zin
PReNet	https://github.com/csdwren/PReNet
UMRL	https://github.com/raieevyasarla/UMRL–using-Cycle-Spinning
SE	https://github.com/wwzier/RainRemoval_ICCV2017
TCLRM	http://mcl.korea.ac.kr/derainina/
DIP	https://github.com/TaiXiangJiang/FastDeRain
FastDeRain	-
J4RNet	https://github.com/flyywh/J4RNet-Deep-Video-Derainina-CVPR-2018
MS-CSC	https://github.com/MinehanLi/MS-CSC-Rain-Streak-Removal
SpacCNN	https://github.com/hotndv/SPAC-SupplementarvMaterials
MRF	https://caibolun.aithub.io/st-mrf/
EVD-Net	https://github.com/Boyiliee/EVD-Net

TABLE 5
Running Time Comparison (in Section) of Different Rain Removal Methods on a Video with the Spatial Resolution 832×512

Methods	JORDER	DetailNet	FastDeRain	SpacCNN	TCLRM
Time	0.6329	1.4698	0.3962	9.5075	192.7007
Methods	MS-CSC	SE	CVPR-2019	J4RNet	Proposed
Time	15.7957	19.8516	0.8974	0.8401	0.5146

TABLE 6
Parameter Comparison of Different Deep-Learning Based Rain Removal Methods

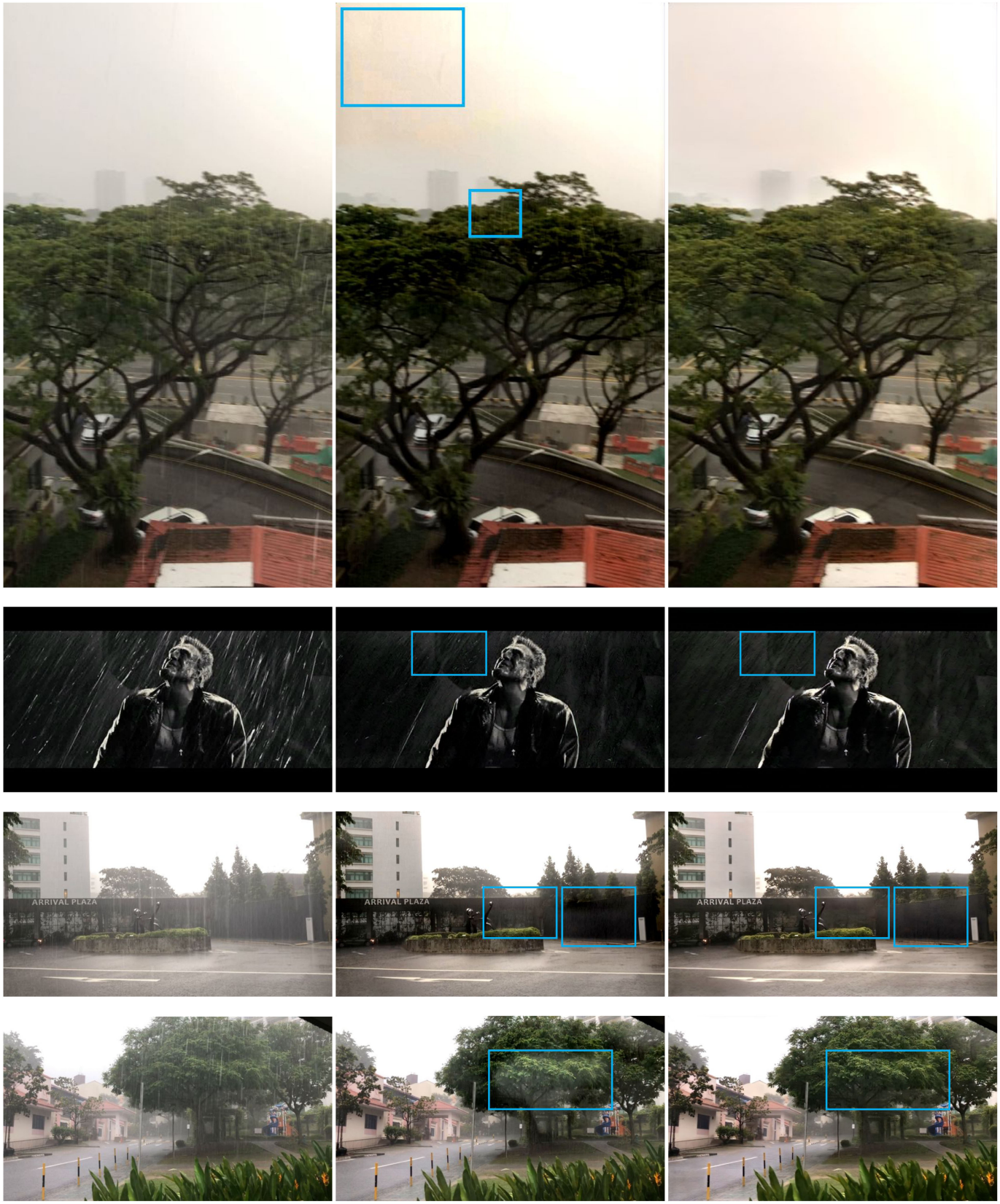
Methods	DetailNet	PreNet	JORDER	SpacCNN
#Para.	58,175	168,963	4,169,024	1,430,403
Method	-	CVPR-2019	Ours-S	Ours-L
#Para.	-	4,466,694	6,964,803	29,472,018

knowledge of adversarial learning and physics model. The first stage provides the physics accurate results and then in the second stage, the results are further processed by the generator trained via the adversarial learning, to adjust the color and contrast distributions as well as to correct details.

TABLE 7
Ablation Analysis for Network Architecture

Baseline	v_1	v_2	v_3	v_4	v_5
Initial-DerainNet	×	✓	✓	✓	✓
Inverse Recovery	✓	✓	×	✓	✓
LSTM	✓	×	✓	✓	✓
Alignment	×	×	×	×	✓
PSNR	23.79	24.82	24.93	25.14	24.76
SSIM	0.9029	0.9166	0.9090	0.9172	0.9160

Best results are denoted in red and the second best results are denoted in blue.



(a) Rain Frame (b) Our previous version (w/o adversarial learning) (c) Our new version (w adversarial learning)

Fig. 17. Visual comparisons of our method and our previous version (CVPR-2019).

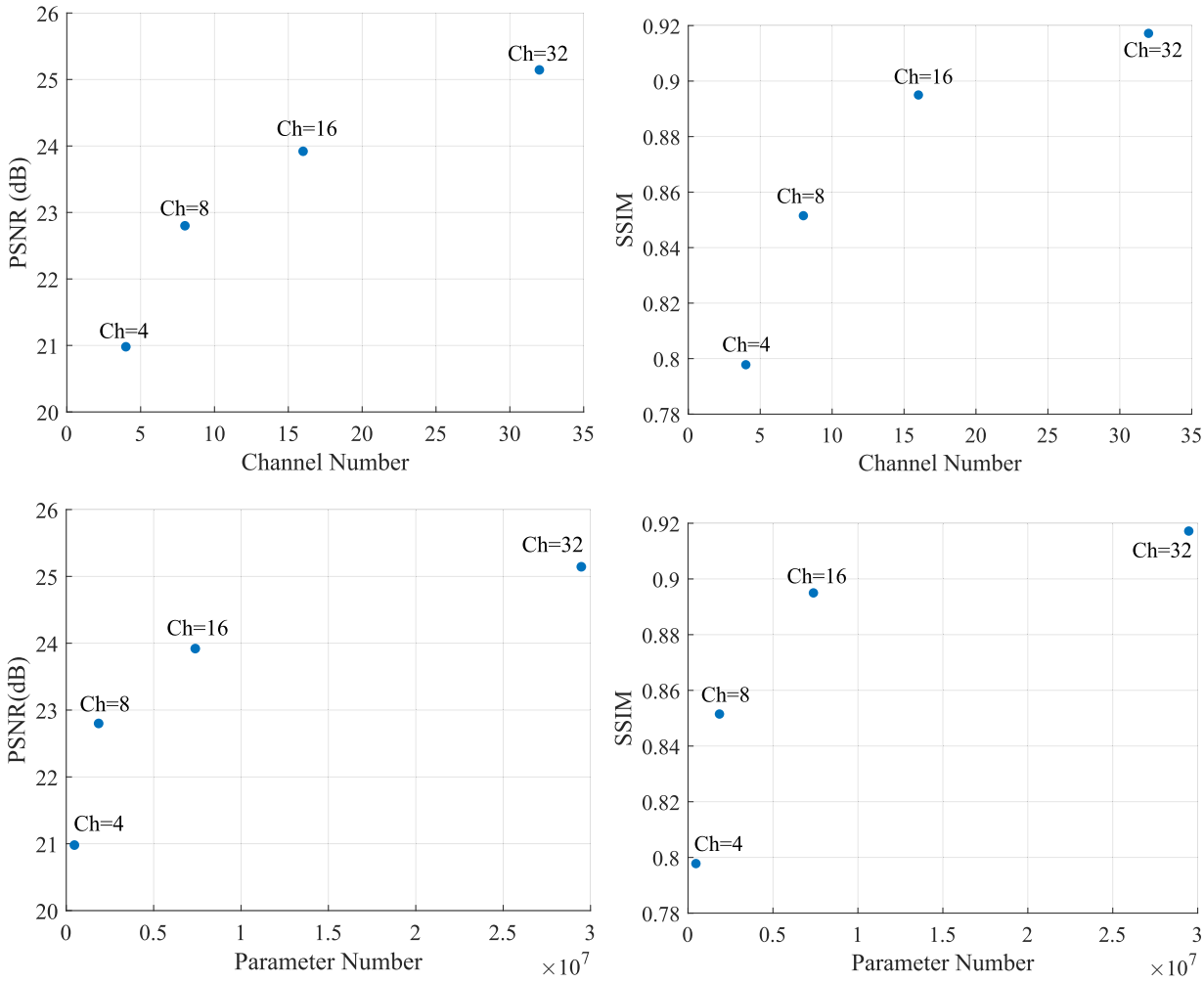


Fig. 18. Performance of versions having different channels and parameters.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China under Grant 2018AAA 0102702, the Fundamental Research Funds for the Central Universities, the National Natural Science Foundation of China under Contract No. 61772043, No. 62022038, and No. 62022002, the National Research Foundation Singapore under its AI Singapore Programme (Award Number: [AISG-100E-2019-035]), the Hong Kong RGC ECS under Grant 21211018, GRF under Grant 11203220. This is a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). The work of Robby T. Tan was supported by MOE2019-T2-1-130.

REFERENCES

[1] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, no. 2/3, 2010, Art. no. 256.

[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," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 348–367, 2011.

[3] B. Cai, X. Xu, and D. Tao, "Real-time video dehazing based on spatio-temporal MRF," in *Proc. Pacific Rim Conf. Multimedia*, 2016, pp. 315–325.

[4] 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. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6286–6295.

[5] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 576–584.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[7] 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. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.

[8] 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. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3855–3863.

[9] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 1794–1807, Jun. 2020.

[10] K. Garg and S. K. Nayar, "When does a camera see rain?," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, 1067–1074.

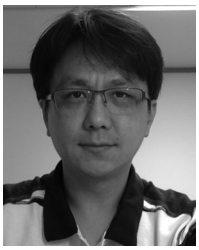
[11] K. Garg and S. K. Nayar, "Photorealistic rendering of rain streaks," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 996–1002, 2006.

[12] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1708–1716.

[13] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, "Depth-attentional features for single-image rain removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8022–8031.

[14] 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 Conf.*, 2018, pp. 413–413.
[15] 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 Trans. Multimedia*, vol. 16, no. 1, pp. 83–93, Jan. 2014.
[16] 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. Conf. Multimedia Expo*, 2012, pp. 164–169.
[17] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, 2008.
[18] T. Jiang, T. Huang, X. Zhao, L. Deng, and Y. Wang, "Fastderain: A novel video rain streak removal method using directional gradient priors," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2089–2102, Apr. 2019.
[19] 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 Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4057–4066.
[20] L. W. Kang, C. W. Lin, and Y. H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.
[21] J. H. Kim, J. Y. Sim, and C. S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015.
[22] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "End-to-end united video dehazing and detection," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7016–7023.
[23] M. Li *et al.*, "Video rain streak removal by multiscale convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6644–6653.
[24] R. Li, L.-F. Cheong, and R. T. Tan, "Single Image deraining using scale-aware multi-stage recurrent network," 2017, *arXiv: 1712.06830*.
[25] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2736–2744.
[26] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? Deep joint recurrent rain removal and reconstruction in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3233–3242.
[27] J. Liu, W. Yang, S. Yang, and Z. Guo, "D3R-Net: Dynamic routing residue recurrent network for video rain removal," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 699–712, Feb. 2019.
[28] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3397–3405.
[29] 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 Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2482–2491.
[30] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3937–3946.
[31] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4210–4219.
[32] V. Santhaseelan and V. K. Asari, "Utilizing local phase information to remove rain from video," *Int. J. Comput. Vis.*, vol. 112, no. 1, pp. 71–89, 2015.
[33] S. Starik and M. Werman, "Simulation of rain in videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2003, pp. 406–409.
[34] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang, "Exploiting image structural similarity for single image rain removal," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 4482–4486.
[35] A. K. Tripathi and S. Mukhopadhyay, "A probabilistic approach for detection and removal of rain from videos," *IETE J. Res.*, vol. 57, no. 1, pp. 82–91, 2011.
[36] A. K. Tripathi and S. Mukhopadhyay, "Video post-processing: Low-latency spatio-temporal approach for detection and removal of rain," *IET Image Process.*, vol. 6, no. 2, pp. 181–196, 2012.
[37] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12270–12279.
[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[39] 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. Conf. Comput. Vis.*, 2017, pp. 2516–2525.
[40] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, 2019.
[41] W. Yang, J. Liu, and J. Feng, "Frame-consistent recurrent video deraining with dual-level flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1661–1670.
[42] 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 Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1357–1366.
[43] 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. Pattern Anal. Mach. Intell.*, Vol. 42, no. 6, pp. 1377–1393, Jun. 2020.
[44] R. Yasarla and V. M. Patel, "Uncertainty guided multi-scale residual learning-using a cycle spinning CNN for single image de-raining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8405–8414.
[45] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 695–704.
[46] 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. Conf. Multimedia Expo*, 2006, pp. 461–464.
[47] L. Zhu, C. Fu, D. Lischinski, and P. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2526–2534.
[48] K. He, J. Sun and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.
[49] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.
[50] W. Ren *et al.*, "Deep video dehazing with semantic segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1895–1908, Apr. 2019.
[51] Z. Li, P. Tan, R. Tan, D. Zou, S. Z. Zhou, and L. Cheong, "Simultaneous video defogging and stereo reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4988–4997.
[52] B. Cai, X. Xu, D. Tao, "Real-time video dehazing based on spatio-temporal MRF," in *Proc. Pacific Rim Conf. Multimedia*, 2016, pp. 315–325.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.
[54] B. Li *et al.*, "Benchmarking single-image dehazing and beyond," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 492–505, Jan. 2019.
[55] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
[56] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, 898–916, May 2011.

Wenhan Yang (Member, IEEE) received the BS and PhD degrees (Hons.) in computer science from Peking University, Beijing, China, in 2012 and 2018. He is currently a postdoctoral research fellow at the Department of Computer Science, City University of Hong Kong. His current research interests include image/video processing/restoration, bad weather restoration and human-machine collaborative coding. He has authored more than 100 technical articles in refereed journals and proceedings, and holds 9 granted patents. He received the IEEE ICME-2020 Best Paper Award, the IFTC 2017 Best Paper Award, and the IEEE CVPR-2018 UG2 Challenge First Runner-up Award. He was the Candidate of CSIG Best Doctoral Dissertation Award in 2019. He served as the area and session chair of IEEE ICME-2021, and the organizer of the IEEE CVPR-2019/2020/2021 UG2+ Challenge and Workshop.

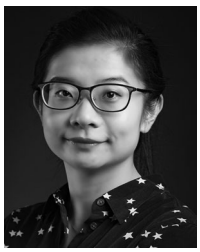


Robby T. Tan (Member, IEEE) received the PhD degree in computer science from the University of Tokyo. He is currently an associate professor at Yale-NUS College and ECE (Electrical and Computing Engineering), National University of Singapore. Previously, he was an assistant professor with Utrecht University. His research interests include machine learning and computer vision, particularly in dealing with bad weather, physics-based, and motion analysis.

Jiashi Feng (Member, IEEE) received the PhD degree from the National University of Singapore, in 2014. He is currently an assistant professor at the Department of Electrical and Computer Engineering, National University of Singapore. Before joining NUS as a faculty, he was a postdoc research follow at UC Berkeley. His research interests include computer vision and machine learning. In particular, he is interested in object recognition, detection, segmentation, robust learning and deep learning.

Shiqi Wang (Member, IEEE) received the BS degree in computer science from the Harbin Institute of Technology, in 2008, and the PhD degree in computer application technology from Peking University, in 2014. From 2014 to 2016, he was a post-doctoral fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. From 2016 to 2017, he was with the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore, as a research fellow. He is currently an assistant professor with the Department of Computer Science, City University of Hong Kong. He has proposed more than 40 technical proposals to ISO/MPEG, ITU-T, and AVS standards, and authored/coauthored more than 150 refereed journal/conference papers. His research interests include video compression, image/video quality assessment, and image/video search and analysis. He received the Best Paper Award from IEEE ICME 2019, IEEE Multimedia 2018, PCM 2017, and is the coauthor of a paper that received the Best Student Paper Award in the IEEE ICIP 2018.

Bin Cheng received the BE degree from the University of Science and Technology of China, and the PhD degree from the National University of Singapore. He is currently a research director at the Beijing Academy of Artificial Intelligence (BAAI). His research interests include computer vision, machine learning and the related applications. And he also successfully shipped a dozen of technologies to incubate industry products, including smart devices, search, live stream, short video, and financial risk-control.



Jiaying Liu (Senior Member, IEEE) received the PhD degree (Hons.) in computer science from Peking University, Beijing China, in 2010. She is currently an associate professor with Peking University Boya Young fellow with the Wangxuan Institute of Computer Technology, Peking University. She has authored more than 100 technical articles in refereed journals and proceedings, and holds 50 granted patents. Her current research interests include multimedia signal processing, compression, and computer vision. She is a senior member of CSIG and CCF. She was a visiting scholar with the University of Southern California, Los Angeles, from 2007 to 2008. She was a visiting researcher with the Microsoft Research Asia in 2015 supported by the Star Track Young Faculties Award. She has served as a member of Multimedia Systems and Applications Technical Committee (MSA TC), and Visual Signal Processing and Communications Technical Committee (VSPC TC) in IEEE Circuits and Systems Society. She received the IEEE ICME-2020 Best Paper Award and IEEE MMSP-2015 Top10 percent Paper Award. She has also served as the associate editor of *the IEEE Trans. on Image Processing*, *the IEEE Trans. on Circuit System for Video Technology* and Elsevier JVCI, the technical program chair of IEEE ICME-2021/ACM ICMR-2021, the publicity chair of IEEE ICME-2020/ICIP-2019, and the area chair of CVPR-2021/ECCV-2020/ICCV-2019. She was the APSIPA distinguished lecturer (2016-2017).