# Erase or Fill? Deep Joint Recurrent Rain Removal and Reconstruction in Videos

Jiaying Liu, Wenhan Yang*, Shuai Yang, Zongming Guo,
Institute of Computer Science and Technology, Peking University, Beijing, P.R. China

## Abstract

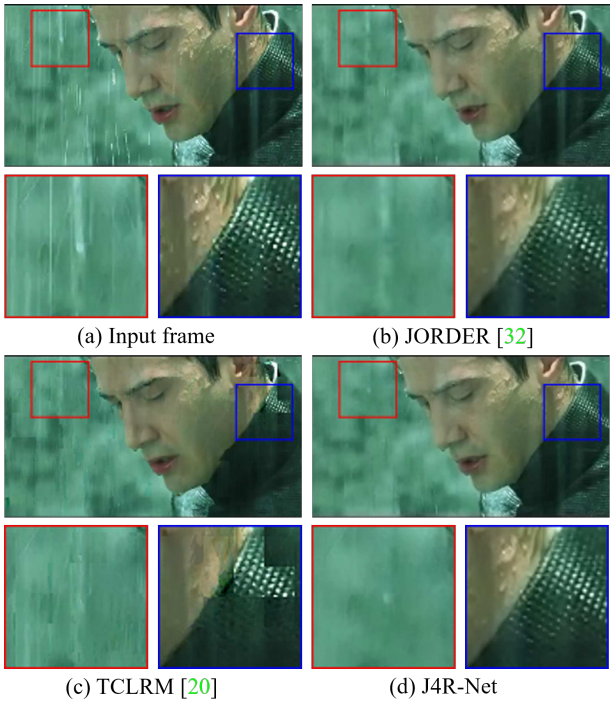
*In this paper, we address the problem of video rain removal by constructing deep recurrent convolutional networks. We visit the rain removal case by considering rain occlusion regions, i.e. the light transmittance of rain streaks is low. Different from additive rain streaks, in such rain occlusion regions, the details of background images are completely lost. Therefore, we propose a hybrid rain model to depict both rain streaks and occlusions. With the wealth of temporal redundancy, we build a **J**oint **R**ecurrent **R**ain **R**emoval and **R**econstruction **Net**work (**J4R-Net**) that seamlessly integrates rain degradation classification, spatial texture appearances based rain removal and temporal coherence based background details reconstruction. The rain degradation classification provides a binary map that reveals whether a location is degraded by linear additive streaks or occlusions. With this side information, the gate of the recurrent unit learns to make a trade-off between rain streak removal and background details reconstruction. Extensive experiments on a series of synthetic and real videos with rain streaks verify the superiority of the proposed method over previous state-of-the-art methods.*

## 1. Introduction

Bad weather conditions cause a series of visibility degradations and alter the content and color of images. Such signal distortion and detail loss lead to the failure of many outdoor computer vision applications, which generally rely on clean video frames as their input. Rain streaks, as one of the most common degradations in rain frames, make severe intensity fluctuations in small regions, and thus obstruct and blur the scene.

In the past decades, many researchers have been dedicated to rain image/video restoration. The rain removal from a single image [18, 14, 27, 23] solves this problem by signal separation between rain streaks and background images (non-rain images), based on their texture appearances. Frequency domain representation [18], sparse representation [23], Gaussian mixture model [21] and deep networks [32, 8] are adopted as basic models to differentiate rain streaks and background images. Furthermore, video-based methods [1, 2, 3, 5, 7, 9, 11, 12, 33] solve the problem based on both spatial and temporal redundancies. Some works [11, 9, 12] built on physical models, such as directional and chromatic properties of rains. Others [5, 4, 20, 17] further utilized temporal dynamics, including continuity of background motions, randomly appearing of streaks among frames, and explicit motion modeling, to facilitate video rain removal.

These methods achieve good effects in some cases.



(a) Input frame (b) JORDER [32]
(c) TCLRM [20] (d) J4R-Net

Figure 1. Demonstration for visual results of different methods on a practical rain video. Compared with JORDER [32] and TCLRM [20], our method successfully removes most rain streaks and enhances visibility significantly.

*Indicates corresponding author. This work was supported by National Natural Science Foundation of China under contract No. 61772043. We also gratefully acknowledge the support of NVIDIA Corporation with the GPU for this research. Email: yangwenhan@pku.edu.cn

However, they still neglect some important issues:

- In real-world scenarios, degradations generated by rain streaks are more complex. The additive rain model widely used in previous methods [18, 5] is insufficient to cover visual effects of some important degradations in practice. When the light transmittance of rain streaks is low, their corresponding background regions are totally occluded, and the whole occlusion regions only present the rain reliance.

- The spatial and temporal redundancies are considered separately. These two kinds of information are intrinsically correlated and complementary. The potential of jointly exploiting the information is not fully explored. Low rank based methods [20, 31] have made some attempts. However, they usually rely on the assumption of a static background. Therefore, their results will be ruined when large and violent motions are included.

- For learning-based video rain streak removal, training for recovery purposes remains challenging. The training relies on the video pairs synthesized from a large-scale high-quality video dataset with various scenes and objects. It is cost-heavy to collect such a dataset to synthesize rain frames.

Considering these limitations of existing works, our goal is to build a novel video rain model that can describe various rain streaks in practice, including both rain streaks and rain occlusions. Then, based on this model, we further develop a deep learning architecture to solve the corresponding inverse problem. We aim to develop a systematic approach to train the network with a rain video dataset synthesized from a medium-sized high-quality video set.

To achieve these goals, we explore possible rain models and deep learning architectures that can effectively restore clean frames even when rain occlusion regions appear.

First, we introduce a hybrid video rain model that is capable of describing rain occlusions. Starting from the simplest additive rain model, we add an additional pixel-wise map – to indicate whether a pixel is occluded or not. In non-rain occlusion regions, the streaks and backgrounds are combine linearly. In rain occlusion regions, the pixels are replaced by rain reliance.

Second, based on this refined model, we analyze its solving paradigm, and construct a deep network. To jointly utilize spatial and temporal redundancies, a recurrent neural network (RNN) is constructed. The rain streaks appear randomly among frames, whereas the motions of backgrounds are tractable. RNN is capable of encoding the information of adjacent background frames from their degraded observations, obtaining more representative features for deraining.

Third, following the solving paradigm, the proposed RNN – **J**oint **R**ecurrent **R**ain **R**emoval and **R**econstruction **Net**work (**J4R-Net**) – seamlessly integrates degradation classification, spatial texture appearances based rain removal and temporal coherence based background detail reconstruction. With the degradation classification map as side information, the gate of the recurrent unit learns to make a trade-off between rain streak removal and background detail reconstruction.

Finally, to train such an RNN network, besides the commonly used synthetic video pairs from natural videos, we also propose to use synthetic videos from natural images with artificially simulated motions to increase the diversity of training data in scenes and objects.

In summary, our contributions are as follows,

- We are the first to visit the rain removal case including rain occlusions. A novel hybrid video rain model is proposed to adapt to these cases.

- We are the first to solve the problem of video rain removal with deep networks. Specifically, a recurrent neural network is used in our work.

- Based on the proposed refined hybrid rain model, a **J**oint **R**ecurrent **R**ain **R**emoval and **R**econstruction **Net**work (**J4R-Net**) is constructed to seamlessly integrate degradation classification, spatial texture appearances based rain removal and temporal coherence based background detail reconstruction.

- We propose to use synthetic videos from natural images with artificially simulated motions to train deraining networks, offering better performance.

## 2. Related Work

### 2.1. Single Image Rain Removal

Single image deraining is a highly ill-posed problem and is addressed by a signal separation or texture classification route. Huang *et al.* [18] attempted to separate rain streaks from the high frequency layer by sparse coding. Then, a generalized low rank model [5] was proposed, where the rain streak layer is assumed to be low rank. Kim *et al.* [19] first detected rain streaks and then removed them with the nonlocal mean filter. Luo *et al.* [23] proposed a discriminative sparse coding method to separate rain streaks from background images. In [21], Li *et al.* exploited Gaussian mixture models to separate the rain streaks. The presence of deep learning promoted the development of single image deraining. In [8], a deep network that takes the image detail layer as its input and predicts the negative residues was constructed. It has a good capacity to keep texture details. But it cannot handle heavy rain cases where rain streaks are dense. In [32], a deep joint rain detection and removal

was proposed to recurrently remove rain streaks and accumulations, obtaining impressive results in heavy rain cases. However, rain streaks and textures of the background are intrinsically overlapped in the feature space. Thus, the remaining weak streaks or over-smooth background textures are usually presented in the results.

### 2.2. Video Rain Removal

Garg and Nayar were the first to focus on modeling rains, *e.g.* the photometric appearance of rain drops [11] and addressing rain detection and removal based on dynamic motion of rain drops and irradiance constraint [9, 12]. In their subsequent work [10], camera settings are explored to control the visibility of rain drops. These early attempts heavily rely on the linear space-time correlation of rain drops, and thus fail when rain streaks are diversified in scales and densities. Later works formulate rain streaks with more flexible and intrinsic models. In [33], the temporal and chromatic properties of rain are visited to differentiate rain, background and moving objects. In [22], a theory of chromatic property of rain is developed. Barnum *et al.* [1] utilized the features in Fourier domain for rain removal. Santhaseelan *et al.* [25] developed phase congruency features to detect and remove rain streaks. Successive works make their efforts in distinguishing fast moving edges and rain streaks. In [3, 2], the size, shape and orientation of rain streaks are used as discriminative features. In [5], the spatio-temporal correlation of local patches are encoded by a low-rank model to separate rain streaks and natural frame signals. Jiang *et al.* [17] further considered the overall directional tendency of rain streaks, and used two unidirectional total variation regularizers to constrain the separation of rain streaks and background. The presence of learning-based method, with improved modeling capacity, brings in new opportunities. Chen *et al.* [4] proposed to embed motion segmentation by Gaussian mixture model into rain detection and removal. Tripathi *et al.* [28, 29] trained Bayes rain detector based on spatial and temporal features. In [20], Kim *et al.* trained an SVM to refine the roughly detected rain maps. Wei *et al.* [31] encoded rain streaks as patch-based mixtures of Gaussian, which is capable of finely adapting a wider range of rain variations.

## 3. Hybrid Video Rain Model

### 3.1. Additive Rain Model

The widely used rain model [21, 23, 15] is expressed as:

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

where $\mathbf{B}$ is the background frame without rain streaks, and $\mathbf{S}$ is the rain streak frame. $\mathbf{O}$ is the captured image with rain streaks. Based on Eq. (1), rain removal is regarded as a signal separation problem [21, 23, 32]. Namely, given the observation $\mathbf{O}$, removing rain streaks is to estimate the background $\mathbf{B}$ and rain streak $\mathbf{S}$, based on the different characteristics of the rain-free images and rain streaks.

Figure 2. Left and middle panels: two adjacent rain frames. Right panel: the rain streaks in these rain frames, denoted in blue and red colors, respectively. The presented streaks have similar shapes and directions, and however, their distributions in spatial locations are uncorrelated.

This single-frame rain synthesis model in Eq. (1) can be extended to a multi-frame one by adding a time dimension as follows,

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

where $t$ and $N$ signify the current time-step and total number of the frames, respectively. Rain streaks $\mathbf{S}_t$ are assumed to be independent identically distributed random samples [26]. Their locations across frames are uncorrelated, as shown in Fig. 2.

Figure 3. Examples of rain occlusions in video frames. Compared with additive rain streaks, the rain occlusions (denoted in red color) contain little structural details of the background image.

However, in practice, degradations generated by rain streaks are very complex. For example, when the rain level is moderate or even heavy, the light transmittance of rain drop becomes low and the rain region of $\mathbf{O}_t$ presents identical intensities, as shown in Fig. 3. In this case, the signal superposition of rain frames includes rain streaks and rain occlusions. Based on Eq. (1), the deduced $\mathbf{S}_t = \mathbf{O}_t - \mathbf{B}_t$ deviates from its original distribution and contains more structure details. Rain removal in rain occlusion regions needs to remove the rain reliance and fill in the missing details. Thus, it is harder to learn a mixture mapping that restores signals in all regions without distinction. It is meaningful to build a unified hybrid model that describes both two kinds of degradation to guide solving the rain removal task.

### 3.2. Occlusion-Aware Hybrid Rain Model

To address this issue, we propose a hybrid rain model that is adaptive to model rain occlusions. In such a model, all pixels in rain frames are classified into two groups: 1) the ones following the additive rain model in Eq. (1); 2) the others whose pixel values are just equal to the rain reliance. The formulation of such a hybrid rain model is given as follows,

$$\mathbf{O}_t = (1 - \alpha_t)(\mathbf{B}_t + \mathbf{S}_t) + \alpha_t \mathbf{A}_t, \tag{3}$$

where $\mathbf{A}_t$ is the rain reliance map and $\alpha_t$ is an alpha matting map defined as follows,

$$\alpha_t(i,j) = \begin{cases} 1, & \text{if } (i,j) \in \Omega_{\mathbf{S}}, \\ 0, & \text{if } (i,j) \notin \Omega_{\mathbf{S}}, \end{cases} \tag{4}$$

where $\Omega_{\mathbf{S}}$ is the region where the light transmittance of rain drop is low, which is defined as *rain occlusion region*.

## 4. Joint Recurrent Rain Removal and Reconstruction Network

### 4.1. From Formulation to Network Design

Video rain removal is to recover the background sequence $\{\mathbf{B}_t\}$, given the input rain sequence $\{\mathbf{O}_t\}$. We rewrite all formulations into the pixel-wise form. Thus, Eq. (3) is rewritten into:

$$\begin{aligned}\mathbf{O}(i,j,t) = &(1 - \alpha(i,j,t))(\mathbf{B}(i,j,t) + \mathbf{S}(i,j,t)) \\ &+ \alpha(i,j,t)\mathbf{A}(i,j,t),\end{aligned} \tag{5}$$

where $(i,j)$ indexes the spatial location, and $t$ indexes the temporal location.

To solve $\mathbf{B}(i,j,t)$, $\alpha(i,j,t)$ is the first to be addressed. The mapping is signified as $F_\alpha(\cdot)$, which can be learned by a CNN/RNN network as follows,

$$\hat{\alpha}(i,j,t) = F_\alpha\left(\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon(i,j,t)\}\right), \tag{6}$$

where $\epsilon(i,j,t)$ is the neighboring pixels of $(i,j,t)$.

Then, $\mathbf{B}(i,j,t)$ is derived in two cases, respectively. When $\hat{\alpha}(i,j,t) = 0$, the rain streak can be estimated by a CNN/RNN network, denoted by $F_{\mathbf{S}}(\cdot)$ as follows,

$$\hat{\mathbf{S}}(i,j,t) = F_{\mathbf{S}}\left(\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon(i,j,t)\}\right). \tag{7}$$

Then, the background can be derived by

$$\hat{\mathbf{B}}(i,j,t) = \mathbf{O}(i,j,t) - \hat{\mathbf{S}}(i,j,t), \tag{8}$$

When $\hat{\alpha}(i,j,t) = 1$, $\mathbf{O}(i,j,t)$ contains no information related to $\mathbf{B}(i,j,t)$. We need to infer the missing information from its neighboring pixels. Thus, $\hat{\mathbf{A}}(i,j,t)$ can be learned by a CNN/RNN network, denoted by $F_{\mathbf{A}}(\cdot)$.

$$\hat{\mathbf{A}}(i,j,t) = F_{\mathbf{A}}\left(\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon(i,j,t)\}\right). \tag{9}$$

The recovery of $\hat{\mathbf{B}}(i,j,t)$ is an reconstruction process,

$$\begin{aligned}\hat{\mathbf{B}}(i,j,t) = F_{\mathbf{B}}\big(&\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon^{\alpha_0}(i,j,t)\}, \\ &\hat{\mathbf{A}}(i,j,t)\big),\end{aligned} \tag{10}$$

where $\epsilon^{\alpha_0}(i,j,t)$ is the neighboring pixels in non-occlusion regions whose $\hat{\alpha}$ value is zero. $F_{\mathbf{B}}(\cdot)$ can also be modeled by a CNN/RNN network. Note that, compared with Eqs. (7)-(9), Eq. (10) is quite different, because ideally one of its two input branches ($\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon^{\alpha_0}(i,j,t)\}$) only uses the input from the non-occlusion regions. With the information in $\epsilon^{\alpha_0}(i,j,t)$, which is more reliable, the lost details of background frames are better filled in.

Here, we argue that, the information inferred from $\epsilon^{\alpha_0}(i,j,t)$ can be well approximated by the temporally aggregated features from adjacent frames. In rain videos, based on the analysis in Sec. 3.1, the rain streaks or rain occlusions appear continuously in spatial locations and randomly among frames. Thus, the temporal redundancy among frames is usually more reliable than the spatial one. Based on this point, we will use the spatial features to perform rain streak removal and temporally aggregated features for background video reconstruction.

In summary, we need to build four mappings to estimate $\hat{\mathbf{A}}(i,j,t)$, $\hat{\alpha}(i,j,t)$, $\hat{\mathbf{S}}(i,j,t)$ (with $\hat{\mathbf{B}}(i,j,t)$ in the case that $\hat{\alpha}(i,j,t) = 0$) and $\hat{\mathbf{B}}(i,j,t)$ (in the case that $\hat{\alpha}(i,j,t) = 1$), respectively.

### 4.2. Network Architecture of J4R-Net

Based on the solution paradigm in the last subsection, we build the J4R-Net network. The network architecture is illustrated as Fig. 4. Briefly, we first extract the features of each frame by a residual CNN. Its output features are expected to estimate rain streaks $\hat{\mathbf{S}}_t$. Then, a small two-layer CNN is built to estimate $\hat{\alpha}_t$. Guided by $\hat{\alpha}_t$, the gated recurrent units are used to fuse the spatial and temporal information. At each time-step, with two inputs – spatial features from the current input frame $\mathbf{O}_t$, and the temporally aggregated memory from adjacent frames (approximated as the inferred information from $\epsilon_{t-1}^{\alpha_0}$) – the gated RNN outputs the fused features (expected to restore $\hat{\mathbf{B}}_t$) and the updated temporally aggregated memory (approximated as the inferred information from $\epsilon_t^{\alpha_0}$).The updated temporally aggregated memory is constrained to recover the details of the current background frame $E(\hat{\mathbf{B}}_t)$. The details of each component of our network are presented in the following.

**Single Frame CNN Extractor (CNN Extractor).** The residual learning architecture [13, 32] is used for single frame CNN feature extraction. As shown in Fig. 5, residual blocks are stacked to build a CNN network. In formulation, let $\mathbf{f}_{t,k,\text{in}}^{\text{c}}$ denote the input feature map of the $k$-th residual block. The output feature map of the $k$-th residual block,

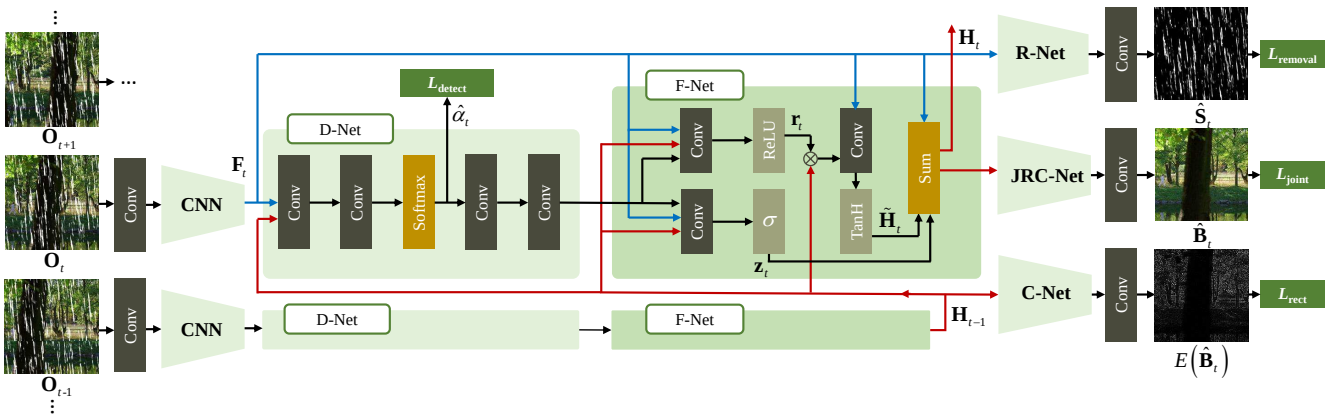



Figure 4. The framework of **J**oint **R**ecurrent **R**ain **R**emoval and **R**econstruction **Net**work (**J4R-Net**). We first employ a CNN to extract features of $t$-th frame $\mathbf{O}_t$. Then, in *degradation classification network* (*D-Net*), based on $\mathbf{F}_t$ and the aggregated feature $\mathbf{H}_{t-1}$ from previous frames, the degradation classification map $\alpha_t$ is detected. Then, in *Fusion Network* (*F-Net*), a gated recurrent neural network, based on $\mathbf{F}_t$, $\mathbf{H}_{t-1}$ and $\hat{\alpha}_t$, the new aggregated feature $\mathbf{H}_t$ is generated. $\mathbf{F}_t$ is inputted into *Removal Network* (*R-Net*) to estimate the rain streak $\hat{\mathbf{S}}_t$. This path makes $\mathbf{F}_t$ separate rain streaks based on spatial appearances. The aggregated feature $\mathbf{H}_{t-1}$ from previous frames is inputted into *reConstruction Network* (*C-Net*) to predict the details of the current frame $E(\hat{\mathbf{B}}_t)$, where $E(\cdot)$ is a high-pass filter. This path makes $\mathbf{H}_{t-1}$ capable of filling in structural details in rain occlusion regions of the current frame. The new aggregated feature $\mathbf{H}_t$ combines the information of two paths. It goes through *Joint Removal and reConstruction Network* (*JRC-Net*) to estimate the background image $\hat{\mathbf{B}}_t$, which is the final output of J4R-Net. The specific network configuration is provided in the supplementary material. (Best viewed in color)

$\mathbf{f}^c_{t,k,\text{out}}$, is progressively updated as follows:

$$\begin{aligned}\mathbf{f}^c_{t,k,\text{out}} &= \max\left(0, \mathbf{W}^c_{t,k,\text{mid}} * \mathbf{f}^c_{t,k,\text{mid}} + \mathbf{b}^c_{t,k,\text{mid}} + \mathbf{f}^c_{t,k,\text{in}}\right),\\ \mathbf{f}^c_{t,k,\text{mid}} &= \max\left(0, \mathbf{W}^c_{t,k,\text{in}} * \mathbf{f}^c_{t,k,\text{in}} + \mathbf{b}^c_{t,k,\text{in}}\right),\end{aligned} \tag{11}$$

where $*$ signifies the convolution operation. $\mathbf{W}$ and $\mathbf{b}$ with subscripts and superscripts denote the weight and bias of the corresponding convolution layers, respectively. $\mathbf{f}^c_{t,k,\text{in}} = \mathbf{f}^c_{t,k-1,\text{out}}$ is the output features of $(k-1)$-th residual block. There is a by-pass connection here between $\mathbf{f}^c_{t,k,\text{in}}$ and $\mathbf{f}^c_{t,k,\text{out}}$. This architecture is proven effective in increasing the network depth and improving network training. The output feature map is denoted as $\mathbf{F}_t$, where $t$ is the time-step of the frame. $\mathbf{F}_t$ encodes the spatial information of $\mathbf{O}_t$.

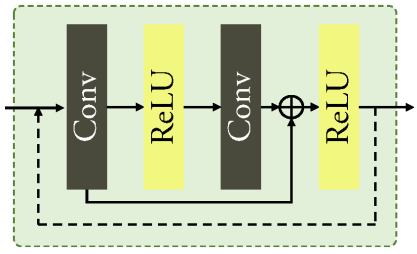



Figure 5. The CNN architecture for single frame CNN feature extraction. R-Net, C-Net and JRC-Net adopt this network architecture as well.

**Degradation Classification Network (D-Net).** Compared with rain removal in a single frame, video rain removal contains temporal sequential information. For rain occlusion regions, the temporal context makes it possible to regard the rain removal as a video reconstruction task, to restore the lost information from adjacent frames. Thus, we detect the degradation type of rain frames explicitly, providing useful side information for successive spatial and temporal redundancy fusion. D-Net takes $\mathbf{F}_t$ and $\mathbf{H}_{t-1}$ as its input, and predicts $\hat{\alpha}_t$ in the middle layer as follows,

$$\begin{aligned}\mathbf{f}^d_{t,0} &= [\mathbf{F}_t, \mathbf{H}_{t-1}],\\ \mathbf{f}^d_{t,1} &= \mathbf{W}^d_{t,1} * \mathbf{f}^d_{t,0} + \mathbf{b}^d_{t,1},\\ \mathbf{f}^d_{t,2} &= \mathbf{W}^d_{t,2} * \mathbf{f}^d_{t,1} + \mathbf{b}^d_{t,2},\\ \hat{\alpha}_t(k) &= \frac{\exp\left(\mathbf{f}^d_{t,2}(k)\right)}{\sum_{j=1,2}\exp\left(\mathbf{f}^d_{t,2}(j)\right)},\end{aligned} \tag{12}$$

where $\mathbf{H}_{t-1}$ is the aggregated memory from the last frame, which jointly encodes previous frames. Then, two layers of convolutions are used to transform $\hat{\alpha}_t(k)$ into the output feature map $\mathbf{f}^d_{t,4}$ as follows,

$$\begin{aligned}\mathbf{f}^d_{t,3} &= \mathbf{W}^d_{t,3} * \hat{\alpha}_t + \mathbf{b}^d_{t,3},\\ \mathbf{f}^d_{t,4} &= \mathbf{W}^d_{t,4} * \mathbf{f}^d_{t,3} + \mathbf{b}^d_{t,4}.\end{aligned} \tag{13}$$

**Fusion Network (F-Net).** After obtaining degradation-dependent features $\mathbf{f}^d_{t,4}$, spatial features $\mathbf{F}_t$, and temporally aggregated memory $\mathbf{H}_{t-1}$, we then consider to fuse them together. Gated recurrent unit (GRU) [6], an advanced RNN architecture, is used. With gate functions, the neuron chooses to read and reset at a time-step. This architecture updates and aggregates internal memory progressively, which facilitates its modeling of long-term temporal dynamics of se-

quential data. The formulations are presented as follows,

$$\begin{aligned}
\mathbf{H}_t^j &= \left(1-\mathbf{z}_t^j\right)\mathbf{H}_{t-1}^j + \mathbf{z}_t^j\widetilde{\mathbf{H}}_t^j,\\
\widetilde{\mathbf{H}}_t^j &= \tanh\left(\mathbf{W}_h\mathbf{F}_t + \mathbf{U}_h\left(\mathbf{r}_t^j\odot\mathbf{H}_{t-1}\right)\right)^j,\\
\mathbf{z}_t^j &= \sigma\left(\mathbf{W}_z\mathbf{F}_t + \mathbf{U}_z\mathbf{H}_{t-1}^j + \mathbf{V}_z\mathbf{f}_{t,4}^d\right)^j,\\
\mathbf{r}_t^j &= \mathrm{ReLU}\left(\mathbf{W}_r\mathbf{F}_t + \mathbf{U}_r\mathbf{H}_{t-1} + \mathbf{V}_r\mathbf{f}_{t,4}^d\right)^j,
\end{aligned} \tag{14}$$

where $j$ indexes the layer number of GRUs. For simplicity, we only show a one-layer GRU in Fig. 4. In fact, several layers of GRUs can be stacked. $\mathbf{H}_t^j$ is interpreted as the aggregated memory, representing the accumulated information at the $t$-th time-step from adjacent frames. $\mathbf{H}_t^j$ is also the output of the unit. $\mathbf{r}_t^j$ is the read gate, controlling the input information from adjacent frames to the current one. $\mathbf{z}_t^j$ is the update gate, deciding how much information of the current frame should be updated to the hidden state. $\widetilde{\mathbf{H}}_t^j$ is the new memory information generated at the $t$-th time-step.

**Rain Removal Network (R-Net).** R-Net aims to separate rain streaks based on spatial features, which makes $\mathbf{F}_t$ good at distinguishing rain streaks and normal textures.

**Reconstruction Network (C-Net).** C-Net aims to fill in missing rain occlusion regions based on temporal redundancy, which makes the network capable of modeling motions and temporal dynamics of background among frames.

**Joint Rain Removal and Reconstruction Network (JRC-Net).** JRC-Net aims to estimate the background frame with both kinds of information. Note that, R-Net, C-Net, and JRC-Net use the same architecture of single-frame CNN network.

**Loss Function.** Let $\hat{\mathbf{B}}_t$, $\hat{\alpha}_t$ and $\hat{\mathbf{S}}_t$ denote the estimated background layer, degradation type mask, and streak layer. Let $\mathbf{B}_t$, $\alpha_t$ and $\mathbf{S}_t$ denote the ground-truth background frame, degradation type mask, and streak layer. The loss function of the network includes four terms,

$$\begin{aligned}
l_{\text{all}} &= l_{\text{joint}} + \lambda_d l_{\text{detect}} + \lambda_c l_{\text{rect}} + \lambda_r l_{\text{removal}},\\
l_{\text{joint}} &= \left\|\hat{\mathbf{B}}_t - \mathbf{b}_t\right\|_2^2,\\
l_{\text{detect}} &= \log\left(\sum_{k=1,2}\exp\left(\mathbf{f}_{t,2}^d(k)\right)\right) - \alpha_t,\\
l_{\text{rect}} &= \left\|E\left(\hat{\mathbf{B}}_t\right) - E\left(\mathbf{b}_t\right)\right\|_2^2,\\
l_{\text{removal}} &= \left\|\hat{\mathbf{S}}_t - \mathbf{s}_t\right\|_2^2,
\end{aligned} \tag{15}$$

where $E(\cdot)$ is a high-pass filter. $\lambda_d$, $\lambda_c$, and $\lambda_r$ are set to 0.001, 0.0001, and 0.0001, respectively.

### 4.3. Removal or Reconstruction: An Intuitive Explanation

We take a closer look at our GRU-based F-Net. If $\mathbf{r}_t^j = 0$ and $\mathbf{z}_t^j = 1$, the network ignores accumulated memory from previous time-steps and just focuses on the current frame:

$$\mathbf{H}_t^j = \tanh\left(\mathbf{W}_h\mathbf{F}_t\right)^j. \tag{16}$$

In this case, the network works as a single frame rain removal network. If $\mathbf{r}_t^j$ is large and $\mathbf{z}_t^j = 0$, $\mathbf{U}_h\left(\mathbf{r}_t^j\odot\mathbf{H}_{t-1}\right)$ plays a dominant role in $\widetilde{\mathbf{H}}_t^j$, and then $\mathbf{H}_t^j$ is more depended on accumulated memory from adjacent frames:

$$\mathbf{H}_t^j = \tanh\left(\mathbf{U}_h\left(\mathbf{r}_t^j\odot\mathbf{H}_{t-1}\right)\right)^j. \tag{17}$$

In this case, the network performs multi-frame background reconstruction. Therefore, learned two gates $\mathbf{r}_t^j$ and $\mathbf{z}_t^j$ control practical functions of the network, and trade-off the benefits between them.

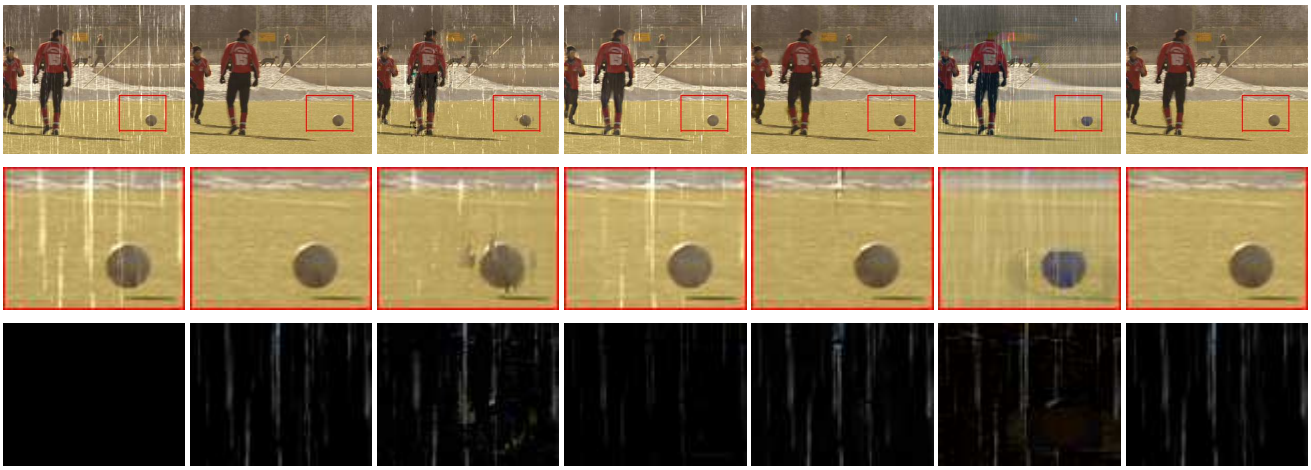
## 5. Experimental Results

We perform extensive experiments to demonstrate the superiority of J4R-Net, as well as effectiveness of its each component. Due to the space limit, some results are presented in the supplementary material.

**Datasets.** We compare J4R-Net with state-of-the-art methods on a few benchmark datasets:

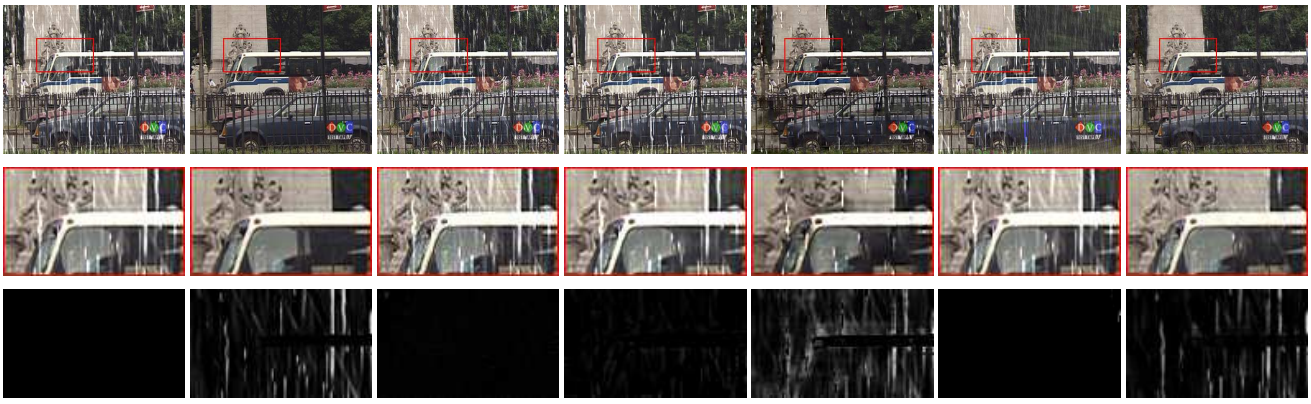
- *RainSynLight25*, which is synthesized by non-rain sequences with the rain streaks generated by the probabilistic model [11];
- *RainSynComplex25*, which is synthesized by non-rain sequences with the rain streak generated by the probabilistic model [11], sharp line streaks [32] and sparkle noises;

Figure 6. Top left panel: one example of *RainSynLight25*. Top right panel: one example of *RainSynComplex25*. Bottom panel: two examples of *RainPractical10*.



(a) Rain image (b) Ground truth (c) TCLRM (d) DetailNet (e) JORDER (f) SE (g) J4R-Net

Figure 7. Results of different methods on an example of *RainSynLight25*. From top to down: whole image, local regions of the estimated background layer, and local regions of the estimated rain streak layer.



(a) Rain image (b) Ground truth (c) TCLRM (d) DetailNet (e) JORDER (f) SE (g) J4R-Net

Figure 8. Results of different methods on an example of *RainSynComplex25*. From top to down: whole image, local regions of the estimated background layer, and local regions of the estimated rain streak layer.

- *RainPractical10*, ten rain video sequences we collected from practical scenes from Youtube website[1], GIPHY[2] and movie clips.

Some examples of *RainSynLight25*, *RainSynComplex25*, and *RainPractical10* are provided in Fig. 6. Our synthesized training and testing data is from CIF testing sequences, HDTV sequences[3] and HEVC standard testing sequences[4]. The augmented video clips are synthesized based on BSD500 [24], with the artificially simulated motions, including rescaling and displacement. More information about training data and training details are provided in the supplementary material.

**Comparison Methods.** We compare J4R-Net with six state-of-the-art methods: discriminative sparse coding (DSC) [23], layer priors (LP) [21], joint rain detection and removal (JORDER) [32], deep detail network (DetailNet) [8], stochastic encoding (SE) [31], temporal correlation and low-rank matrix completion (TCLRM) [20]. DSC, LP, JORDER and DetailNet are single frame deraining methods. SE and TCLRM are video derainig methods. JORDER and DetailNet are deep-learning based methods.

For the experiments on synthesized data, two metrics Peak Signal-to-Noise Ratio (PSNR) [16] and Structure Similarity Index (SSIM) [30] 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.

**Quantitative Evaluation.** Table 1 shows the results of different methods on *RainSynLight25* and *RainSynComplex25*. As observed, our method considerably outperforms

[1]https://www.youtube.com/
[2]https://giphy.com/
[3]https://media.xiph.org/video/derf/
[4]http://ftp.kw.bbc.co.uk/hevc/hm-10.0-anchors/bitstreams/

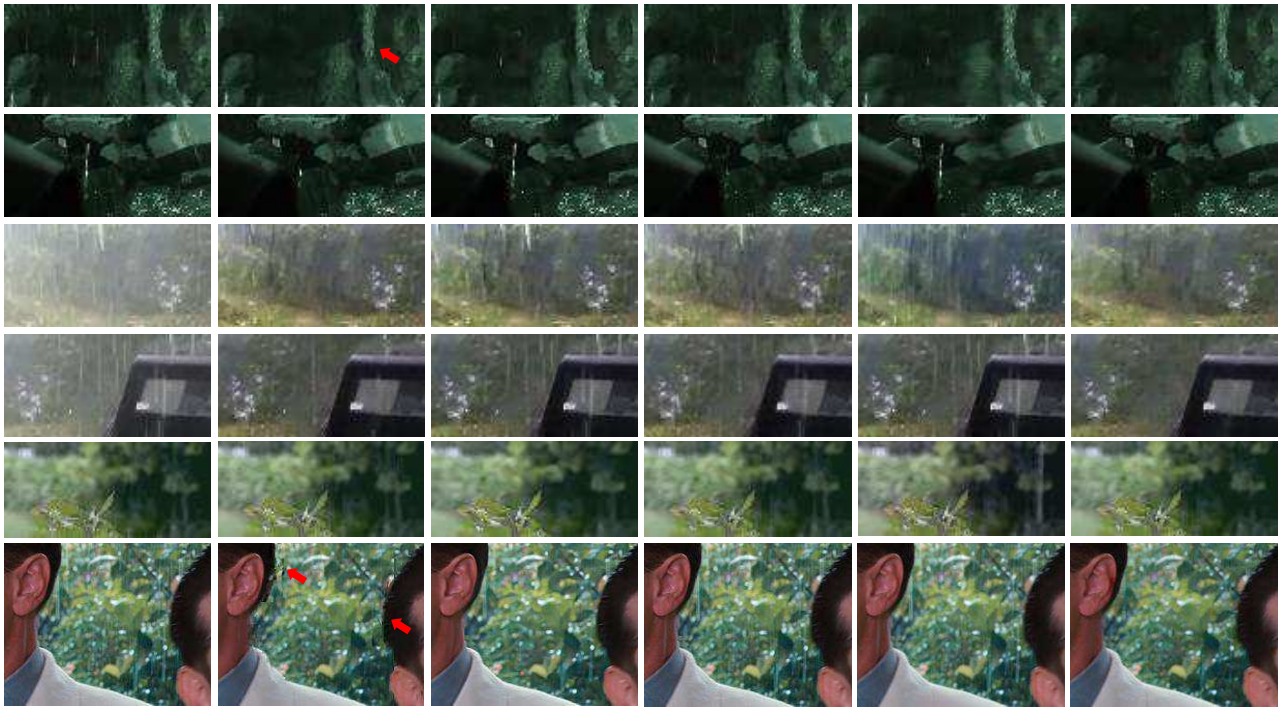



Figure 9. Results of different methods on practical images.

Table 1. PSNR and SSIM results among different rain streak removal methods on *RainSynLight25* and *RainSynComplex25*.

| Dataset | Rain Images | | DetailNet | | TCLRM | | JORDER | |
|---|---|---|---|---|---|---|---|---|
| Metrics | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| *Light* | 23.69 | 0.8058 | 25.72 | 0.8572 | 28.77 | 0.8693 | 30.37 | 0.9235 |
| *Heavy* | 14.67 | 0.4563 | 16.50 | 0.5441 | 17.31 | 0.4956 | 20.20 | 0.6335 |
| Dataset | LP | | DSC | | SE | | Ours | |
| Metrics | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| *Light* | 27.09 | 0.8566 | 25.63 | 0.8328 | 26.56 | 0.8006 | **32.96** | **0.9434** |
| *Heavy* | 17.65 | 0.5364 | 17.33 | 0.5036 | 16.76 | 0.5293 | **24.13** | **0.7163** |

other methods in terms of both PSNR and SSIM. The PSNR of J4R-Net is higher than that of JORDER, the state-of-the-art sinlge image rain removal method, with margins at more than 3dB and 5dB on *RainSynLight25* and *RainSynComplex25*, respectively. J4R-Net also obtains higher SSIM values than JORDER, with margins at about 0.03 and 0.14 on *RainSynLight25* and *RainSynComplex25*, respectively. Compared with SE and TCLRM, J4R-Net also achieves higher PSNR and SSIM. The gains of PSNR are about 5dB and 8dB on *RainSynLight25* and *RainSynComplex25*, respectively. The gains of SSIM are more than 0.08 and 0.25 on *RainSynLight25* and *RainSynComplex25*, respectively.

**Qualitative Evaluation.** Fig. 9 shows the results of practical images. Due to the space limit, we here only present the zooming-in local results. Their corresponding full results are provided in the supplementary material. TCLRM and J4R-Net remove the majority of rain streaks successfully. However, the result of TCLRM may contain artifacts in the area with large motions, as denoted by the red arrows. J4R-Net achieves superior performance in both removing rain streaks and avoiding artifacts.

## 6. Conclusion

In this paper, we proposed a hybrid rain model to depict both rain streaks and occlusions. Guided by this model, a **J**oint **R**ecurrent **R**ain **R**emoval and **R**econstruction **Net**work (**J4R-Net**) was built to seamlessly integrate rain degradation classification, spatial texture appearances based rain removal and temporal coherence based background details reconstruction. With a binary mask generated by rain degradation classification to denote the degradation type, the gate of the recurrent unit made a trade-off between rain streak removal and background details reconstruction. Extensive experiments on a series of synthetic and practical videos with rain streaks verified the superiority of the proposed method over previous state-of-the-art methods.

## 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. 1, 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. 1, 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. 1, 3

[4] 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. 1, 3

[5] 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. 1, 2, 3

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. *Empirical evaluation of gated recurrent neural networks on sequence modeling*. 2014. 5

[7] 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. 1

[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. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, July 2017. 1, 2, 7

[9] 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 I–528, 2004. 1, 3

[10] 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. 3

[11] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM Trans. Graphics*, volume 25, pages 996–1002, 2006. 1, 3, 6

[12] K. Garg and S. K. Nayar. Vision and rain. *Int'l Journal of Computer Vision*, 75(1):3–27, 2007. 1, 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, June 2016. 4

[14] 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

[15] 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

[16] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 7

[17] 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, 3

[18] 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

[19] 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

[20] 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. 1, 2, 3, 7

[21] 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. 1, 2, 3, 7

[22] 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

[23] 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, 7

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its

application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 7

[25] 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

[26] S. Starik and M. Werman. Simulation of rain in videos. In *Texture Workshop, ICCV*, June 2003. 3

[27] 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

[28] 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

[29] 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

[30] 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

[31] 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. 2, 3, 7

[32] 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. 1, 2, 3, 4, 6, 7

[33] 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. 1, 3