

# A directional global sparse model for single image rain removal

Liang-Jian Deng\*, Ting-Zhu Huang<sup>†</sup>, Xi-Le Zhao<sup>‡</sup>, Tai-Xiang Jiang<sup>§</sup>

*School of Mathematical Sciences/Research Center for Image and Vision Computing,  
University of Electronic Science and Technology of China,  
Chengdu, Sichuan, 611731, P. R. China*

## Abstract

Rain removal from a single image is an important issue in the fields of outdoor vision. Rain, a kind of bad weather that is often seen, usually causes complex local intensity changes in images and has negative impact on vision performance. Many existing rain removal approaches have been proposed recently, such as some dictionary learning-based methods and layer decomposition-based methods. Although these methods can improve the visibility of rain images, they fail to consider the intrinsic directional and structural information of rain streaks, thus usually leave undesired rain streaks or change the background intensity of rain-free region significantly. In the paper, we propose a simple but efficient method to remove rain streaks from a single rainy image. The proposed method formulates a global sparse model that involves three sparse terms by considering the intrinsic directional and structural knowledge of rain streaks, as well as the property of image background information. We employ alternating direction method of multipliers (ADMM) to solve the proposed convex model which guarantees the global optimal solution. Results on a variety of synthetic and real rainy images demonstrate that the proposed method outperforms two recent state-of-the-art rain removal methods. Moreover, the proposed method needs no training and requires much less computation significantly.

*Key words:* Single image rain removal; Directional sparse model; Alternating direction method of multipliers.

## 1 Introduction

Many outdoor vision problems are caused by bad weather, in which the local (or global) intensities and color contrast of outdoor images are often degraded so that human can not see the clear and visible scenes. In general, the severe weather can be classified into two categories [1, 2], one is generated by microscopic particles such as haze, smoke and fog, the other is caused by large particles including rain, hail and snow. The two categories of severe weather usually lead to different impacts on outdoors images. For instance, the bad weather caused by microscopic particles generally results in the globally information loss of image intensities, *e.g.*, haze [3], while the severe weather caused by large particles often affects the local image intensities and distributes randomly in the image, *e.g.*, rain [2, 4].

There are many rain removal approaches in the last few decades from different perspectives of modeling. These methods generally fall into two categories. One is video based methods that realize the

---

\*Corresponding author. E-mail: liangjian.deng@uestc.edu.cn

<sup>†</sup>Corresponding author. E-mail: tingzhuhuang@126.com

<sup>‡</sup>E-mail: xlzhao122003@163.com

<sup>§</sup>E-mail: taixiangjiang@gmail.com

rain streaks removal by utilizing the rich properties of multiple relative images, the other is single image based methods that remove the rain streaks only using one input rainy image. Comparing with video rain removal, single image rain removal is more applicable when there is only one rainy image available. Obviously, it is also more challenging. In this paper, we focus on how to remove rain streaks from a single rainy image.

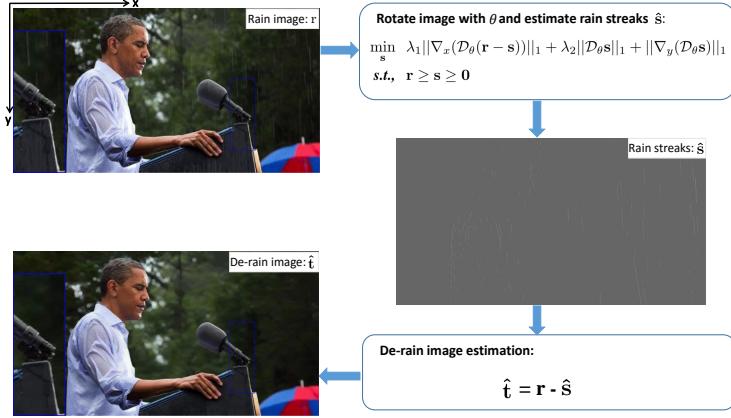


Figure 1: The flow chart of the proposed method. The intensity range of the real rainy image is [0, 255], here we add 102 to the intensity of rain streaks  $\hat{s}$  for better visibility. Note that for this  $464 \times 886 \times 3$  color image, the proposed method only takes about 1.9 second on our computer.

For video based methods, there are very rich literatures about this type of methods [1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], since the redundant temporal information in videos would greatly help detect rain streaks and recover desired de-rain images by using the similarity between image sequences. Garg and Nayar [1] introduce a correlation model and a physics-based motion blur model to detect and remove the dynamics of rain in videos. Once the rain streaks are detected, the pixel values of the corresponding rain locations are obtained by averaging their rain-free temporal neighboring pixels. Garg and Nayar [5] propose a novel approach to control a video camera's parameters when taking a rainy image. In their paper, they reveal the truth that rain visibility significantly depends on the exposure time and the depth of field. Based on this truth, they design an approach which automatically adjusts parameters when capturing the video, aiming to reduce the rain streaks effectively. On the basis of their works [1, 5], Garg and Nayar [6] further reveals the relation between vision and rain, then develops a more efficient method for video rain removal. Moreover, Zhang *et al.* [7] utilize a temporal prior which assumes rain streaks do not appear everywhere in the video, and a chromatic changes prior that rain streaks are almost the same across RGB channels.

For single image rain removal, only less works have been proposed as a comparison with video based methods [16, 17, 18, 19, 2, 20, 4, 21, 22, 23]. The single image rain removal is actually more difficult than video cases due to the only available input image. Recently, Luo *et al.* [2] assume that a rainy image is consisted of a rain layer and a background layer, and develop a dictionary learning method based on a nonlinear screen blend model for single image rain removal. This method sparsely approximates the patches of two layers by highly discriminative sparse coding and obtains accurate layer decomposition. Different from Luo *et al.*, Li *et al.* [4] take a linear superimposition of the desired background layer and the rain streak layer into consideration. Their approach can be formulated as an energy minimization model, in which the corresponding patch-based priors of rain layer and background layer are learned

by Gaussian mixture models. The method can describe different orientations and scales of the rain streaks, and obtains state-of-the-art results on several examples. Although these single image rain removal methods can get excellent results visually and quantitatively, there still exist significant rain streaks for some examples, in the meanwhile the true scene of background sometimes appears to be over-smoothed. In general, existing rain streak removal approaches consider various properties and priors but the directional property of rain streaks, which is helpful for the separation of rain streaks and the desired background. Motivated by unidirectional total variation (UTV)[24] that is used for removing the stripe noise of remote sensing images, here we will consider utilizing the UTV to characterize the important directional property.

Recently, deep learning based methods have become a new trend in image/video rain removal, see *e.g.*, [25, 26, 27, 28]. Tools of deep learning are very interesting, and they are able to get state-of-the-art performance for rain removal. Deep learning uses a training phase where the training of the net is crucial and is the most time demanding phase. With the development of deep learning, we believe that it can obtain better performance in the future, not only in the fields of image processing but also in the other computer vision tasks.

In this paper, to remove rain streaks of a single image, we enforce three sparse priors on rain streaks and the desired rain-free image, which are from some statistical analysis (see Fig. 2), including two directional sparse priors. The first sparse prior is imposed on rain streaks that can be viewed as sparse components (partly sparse when rain is heavy). Moreover, we enforce a sparse prior of the variation along the vertical direction ( $y$ -direction) of rainy image (see the direction defined in Fig. 1), since rain streaks generally come down from top to bottom. For the rain streaks far away from the vertical direction, we take a rotational strategy to easily deal with it. Furthermore, the sparse prior of the variation of rain-free image along the horizontal direction ( $x$ -direction) is utilized to constrain the rain-free image. Combining the three priors, the final convex minimization model is formed and then solved by ADMM method efficiently. In the meanwhile, the global optimal of the proposed method is guaranteed. Results on several synthetic and real images demonstrate that the proposed method outperforms recent state-of-the-art rain removal methods, *i.e.*, [2, 4].

#### **Advantages of the proposed method:**

- The proposed method is to found a concise and efficient optimization model that utilizes three simple sparse priors to describe the latent and intrinsic properties of rain streaks. Experimental results on several examples show that the proposed method performs better than two recent state-of-the-art methods not only on removing rain streaks, but also on preserving the content of background. Furthermore, the proposed directional model is quite novel since many existing state-of-the-art methods fail to consider the key directional property of rain streaks in their models.
- Since the proposed model is convex and we employ ADMM to solve the model, therefore, the global optimal of the proposed method is guaranteed.
- The proposed method only involves three cheap soft-thresholding operators and few times of fast Fourier transformation (FFT) for one iteration of ADMM, thus requires significantly less computation than the two compared state-of-the-art rain removal methods (our computation complexity  $\mathcal{O}(n \cdot \log n)$ ).

The organization of this paper is as follows. In Section 2, the related work will be introduced briefly. In Section 3, we detailedly describe the proposed method that includes the proposed minimization model and the corresponding algorithm. In Section 4, we mainly compare the proposed method with two state-of-the-art single image rain removal methods, and discuss the results under different conditions. Finally, we will draw conclusions in Section 5.

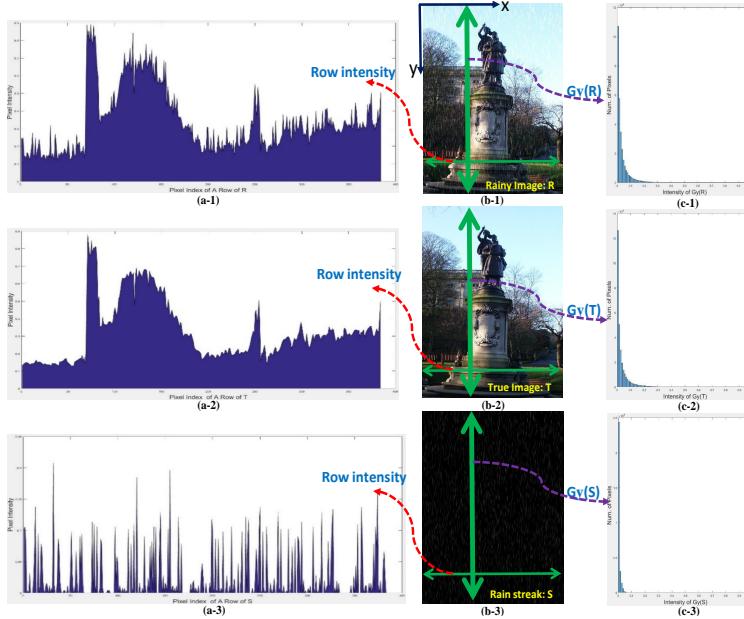


Figure 2: Motivation of the proposed framework. (a-1)-(a-3) represent the pixel intensity along one row of rainy image (horizontal direction), true image and rain streaks, respectively. (b-1)-(b-3) are the rainy image, true image and rain streaks, respectively. In addition, (c-1)-(c-3) respectively stand for the intensity distribution along the vertical direction in the gradient domain. Note that, (c-1)-(c-3) are obtained by the statistical operating on 38 images of UCID dataset that will be introduced in the section of results.

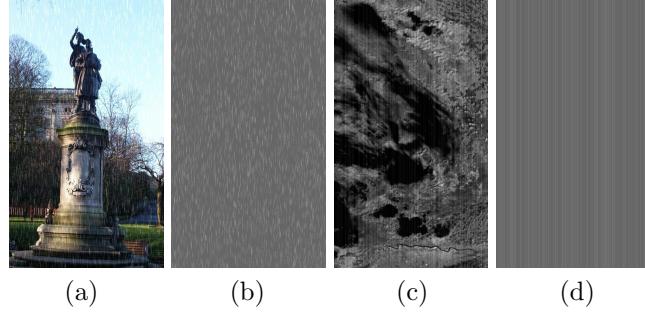


Figure 3: The similarity of derain problem and destriping problem. (a) Rainy image; (b) The corresponding rain streaks of (a); (c) The remote sensing image with stripe noise; (d) The stripe noise of (c). Note that the rain streaks (b) and stripe noise (d) both show significant directional and structural property.

## 2 Related work

In general, the rain model can be described as follows

$$\mathbf{R} = \mathbf{T} + \mathbf{S}, \quad (1)$$

where  $\mathbf{R} \in \mathbb{R}^{M \times N}$  is the observed rainy image,  $\mathbf{T} \in \mathbb{R}^{M \times N}$  is the unknown rain-free image and  $\mathbf{S} \in \mathbb{R}^{M \times N}$  represents the rain streaks.

For the sake of simplicity, we rewrite Eq. (1) as follows

$$\mathbf{r} = \mathbf{t} + \mathbf{s}, \quad (2)$$

where  $\mathbf{r} \in \mathbb{R}^{MN \times 1}$ ,  $\mathbf{t} \in \mathbb{R}^{MN \times 1}$  and  $\mathbf{s} \in \mathbb{R}^{MN \times 1}$  lexicographically represent the vector versions of  $\mathbf{R}$ ,  $\mathbf{T}$  and  $\mathbf{S}$ , respectively.

The rain model (2) is actually an additive model which is exactly the same with the additive image noise model, and the rain streaks  $\mathbf{s}$  can be viewed as the noise as well. To illustrate the motivation of our work, here we exhibit the related work from two aspects. 1) As we known, total variation (TV) model [29] is a quite powerful tool for additive noise removal. Therefore, how to use TV-based model to remove the additive noise (or rain streaks) is a key issue. 2) In addition, in the application of destriping of remote sensing images, the stripe noise also has the same formulation as (2). Besides, rain streaks and stripe noise all keep the similar directional property (see Fig. 3). These two points motivate us to present the following excellent work on remote sensing image destriping via unidirectional TV (UTV) model [24].

### 2.1 UTV for the destriping of remote sensing images

Similarly, the stripe noise model of remote sensing images can be formulated from the variational point as follows

$$\mathbf{I}(x, y) = \mathbf{I}_u(x, y) + \mathbf{I}_s(x, y), \quad (3)$$

where  $\mathbf{I}(x, y)$  is the known observed image with stripe noise,  $\mathbf{I}_u(x, y)$  and  $\mathbf{I}_s(x, y)$  represent the unknown true stripe-free image and stripe noise at the location  $(x, y)$ , respectively. Since most stripe noise can be assumed as constant over a given scan line, which indicates that the stripe noise has the directional property, Bouali and Ladjal [24] constructed a directional model for the destriping problem. For instance,

from Fig. 3(d), it is clear that the stripe noise contains directional and structural information along the vertical direction. In other words, the following relation holds for the stripe noise:

$$\left| \frac{\partial \mathbf{I}_s(x, y)}{\partial y} \right| \leq \left| \frac{\partial \mathbf{I}_s(x, y)}{\partial x} \right|, \quad (4)$$

Nonlinear approaches have been applied to many image applications, e.g., image denoising [30, 31, 32], image super-resolution [33, 34], etc. Thereinto, the TV model has shown a powerful ability for image denoising. This model was first proposed by Rudin, Osher and Fatemi (ROF) [29] and extended to various of image applications. It minimizes the following energy functional,

$$E(\mathbf{I}_u) = \int_{\Omega} \|\mathbf{I}_u - \mathbf{I}\| + \lambda TV(\mathbf{I}_u), \quad (5)$$

where  $\lambda$  is a positive regularization parameter to balance the two terms, and the  $TV(\mathbf{I}_u)$  represents the total variation of the estimated solution  $\mathbf{I}_u$ , see the following expression,

$$TV(\mathbf{I}_u) = \int_{\Omega} |\nabla \mathbf{I}_u| = \int_{\Omega} \left| \frac{\partial \mathbf{I}_u}{\partial x} \right| + \left| \frac{\partial \mathbf{I}_u}{\partial y} \right| dx dy. \quad (6)$$

In particular, the ROF model has excellent ability to preserve discontinuous information, such as image edges. The property is quite crucial to removing the artifacts of remote sensing images and simultaneously keeping the high-frequency image details. By the relation (4), we have

$$\int_{\Omega} \left| \frac{\partial \mathbf{I}_s(x, y)}{\partial y} \right| dx dy \leq \int_{\Omega} \left| \frac{\partial \mathbf{I}_s(x, y)}{\partial x} \right| dx dy, \quad (7)$$

which indicates that:

$$TV_y(\mathbf{I}_s) \leq TV_x(\mathbf{I}_s), \quad (8)$$

where  $TV_x$  and  $TV_y$  represent the horizontal and vertical variations, respectively. To remove the stripe noise of remote sensing images, the authors in [24] proposed the following minimization energy functional that involves two unidirectional variations,

$$\min_{\mathbf{I}_u} E(\mathbf{I}_u) = TV_y(\mathbf{I}_u - \mathbf{I}) + \tilde{\lambda} TV_x(\mathbf{I}_u), \quad (9)$$

where  $\tilde{\lambda}$  is a positive regularization parameter, and it is also rewritten as follows

$$\min_{\mathbf{I}_u} E(\mathbf{I}_u) = \int_{\Omega} \left| \frac{\partial(\mathbf{I}_u - \mathbf{I})}{\partial y} \right| + \left| \frac{\partial \mathbf{I}_u}{\partial x} \right| dx dy. \quad (10)$$

For the solution of variational model (10), authors in [24] gave an iterative scheme based on Euler-Lagrange equation and gradient descent method. Here, we only focus on the variational model. More details of the algorithm can be found from the reference [24]. In summary, the success of the unidirectional destriping model (10) indicates the following two key points, one is that the directional (vertical and horizontal) information separated into the fidelity term and the regularization term is significantly helpful for destriping, the other is both fidelity and regularization terms should use an edge-preserving norm, e.g.,  $TV$  norm and  $\ell_1$  norm, to avoid blurring artifacts [24].

## 2.2 From UTV to the motivation of image rain streaks removal

As stated before, the natural image rain removal is similar to the destriping of remote sensing images. For instance, from Fig. 3, the rain streaks and the stripe noise of remote sensing images both contain directional information (rain streaks and stripe noise both comes from top to bottom) and structural information (the intensity within rain streaks and stripe noise remain almost constant). This motivates us to adopt the UTV based model to characterize the directional property of rain streaks for natural image rain removal problem.

In particular, the rain streaks and the stripe noise also have some significant differences. 1) The stripe noise of remote sensing images generally distributes through the whole image, while the rain streaks only cover a local region. 2) The stripe noise of remote sensing images is generally only along the vertical direction or the horizontal direction, which makes the destriping problem easier. However, the rain streaks usually are not along the exact vertical direction, which indicates that there commonly exists a small angle between the direction of rain streaks and the vertical direction. 3) The intensities of remote sensing images normally exceed the range of [0, 255], *e.g.*, it can become 1024 for 10 bit remote sensing data, while the intensities of natural rainy images generally fall into the range of [0, 255]. In spite of these difference, we can still utilize the directional and structural properties of rain streaks and extend UTV model to the rain streaks removal task.

Moreover, many existing image rain streaks removal approaches for the additive rain model fail to consider the directional and structural information. It may result in the leaving of rain streaks or the loss of background image details. In our work, we focus on the intrinsic directional and structural property of rain streaks. We incorporate the ideology of UTV model to finally formulate the directional image de-rain model. We will detailedly present the proposed model for image rain removal in the next section.

## 3 The proposed method

Recall the rain model (1), we will give the proposed model by extending the UTV model and considering some other latent priors which are motivated by some statistical analysis in Fig. 2. The proposed convex model whose global optimal is guaranteed mainly contains three simple sparse regularizers, including two directional sparse priors on rain streaks and background image respectively and a generally sparse prior on rain streaks.

### 3.1 The proposed optimization model

**1) The Sparsity of Rain Streaks:** In the problem of rain streaks removal, we actually can approximately consider the rain streaks being sparse when the rain is not heavy. Naturally, using  $\ell_0$  norm of vectors to describe the sparsity is an ideal way, since the  $\ell_0$  norm indicates the number of nonzero elements. However, due to the non-convexity of  $\ell_0$  norm, we settle for using  $\ell_1$  norm to depict the sparsity. One advantage of adopting  $\ell_1$  norm is its convexity that can promise the global optimal in some optimal algorithms, *e.g.*, ADMM. In addition, the undesired effects on the rain-free regions can be avoided when enforcing  $\ell_1$  norm term of rain streaks. Thus  $\ell_1$  norm is directly employed as one of the regularizers here, see as follows

$$\mathbf{Reg}^{(1)}(\mathbf{s}) = \|\mathbf{s}\|_1. \quad (11)$$

**2) The Sparsity of Rain Streaks along the vertical direction:** In the real scenes of rainfall, the rain streaks generally come down from top to bottom, that is to say along the vertical direction of the rainy image (*i.e.*, the  $y$ -direction). Because of the smoothness within the rain streaks, the difference between adjacent pixels is generally quite small and can be approximately viewed as zero. Thus we also consider the sparse prior for the variation along the vertical direction of the rainy image. In particular, the rain

streaks sometimes do not fall off strictly along the vertical direction, *e.g.*, there exists an angle between the vertical direction and the real rain falling direction, but the sparsity of the variation along the vertical direction still holds if the angle is small. Actually, detecting the real direction of falling rain is a very difficult task. Thus for the large angle case we may utilize rotation strategy to deal with it easily (see details in Section 3.3).

The final regularizer for the variation within the rain streaks is assumed as follows

$$\mathbf{Reg}^{(2)}(\mathbf{s}) = \|\nabla_y \mathbf{s}\|_1, \quad (12)$$

where  $\nabla_y \mathbf{s}$  stands for the vector form of  $\nabla_y \mathbf{S}$ , and  $\nabla_y$  is the difference operator in terms of the vertical direction. Actually, this term can be viewed as a transformation of the first regularization term of UTV model (9).

**3) The Sparsity of Rain-Free Image along the horizontal direction:** Inspired by the work of unidirectional Total Variation (UTV) [24], to obtain a robust rain removal, the variational information across rain streak direction is used for determining the discontinuity of rain streaks. Here, the across-rain streak direction is simply viewed as horizontal direction. Moreover, due to the relation of  $\mathbf{t} = \mathbf{r} - \mathbf{s}$ , thus the final regularizer about the rain-free image across the rain streak direction is given as follows

$$\mathbf{Reg}^{(3)}(\mathbf{s}) = \|\nabla_x(\mathbf{r} - \mathbf{s})\|_1, \quad (13)$$

where  $\nabla_x(\mathbf{r} - \mathbf{s})$  represents the vector form of  $\nabla_x(\mathbf{R} - \mathbf{S})$ ,  $\nabla_x$  is the difference operator in terms of the horizontal direction. Similarly, this term can also be viewed as the transformation of the second regularization term of UTV model (9).

**4) Nonnegative constraint:** In natural image de-rain problem, the rain streaks  $\mathbf{s}$  are nonnegative and generally have the brightest intensity in a rainy image. Thus for the rainy image  $\mathbf{r}$  and the rain streaks  $\mathbf{s}$ , the following constraint holds:

$$\mathbf{r} \geq \mathbf{s} \geq \mathbf{0}. \quad (14)$$

After analyzing the properties of rain streaks and giving the corresponding sparse regularizers, we may find the naive sparse model strictly along the horizontal ( $x$ -) and vertical ( $y$ -) direction for rain removal problem. However, sometimes the direction of rain streaks is far away from the vertical direction (*i.e.*,  $90^\circ$ ), and the mentioned regularizers seem to be unreasonable. Therefore, here we employ a rotation operator  $\mathcal{D}_\theta$  with the rotation angle  $\theta$  for the three regularizers, *i.e.*,  $\|\mathcal{D}_\theta \mathbf{s}\|_1$ ,  $\|\nabla_y(\mathcal{D}_\theta \mathbf{s})\|_1$  and  $\|\nabla_x(\mathcal{D}_\theta(\mathbf{r} - \mathbf{s}))\|_1$ .

Therefore, the final minimization model for solving the rain removal problem is summarized as follows

$$\begin{aligned} & \min_{\mathbf{s}} \lambda_1 \|\nabla_x(\mathcal{D}_\theta(\mathbf{r} - \mathbf{s}))\|_1 + \lambda_2 \|\mathcal{D}_\theta \mathbf{s}\|_1 + \|\nabla_y(\mathcal{D}_\theta \mathbf{s})\|_1, \\ & \text{s.t., } \mathbf{r} \geq \mathbf{s} \geq \mathbf{0}, \end{aligned} \quad (15)$$

where  $\lambda_1$  and  $\lambda_2$  are two positive regularization parameters. In particular, the proposed convex model (15) is similar to the model in [35] which is utilized for the destriping problems of remote sensing images. However, the latter one employs  $\ell_0$ -norm regularizer to generate a non-convex model which fails to guarantee the global optimal.

### 3.2 The algorithm for the proposed model

Since the proposed  $\ell_1$ -norm model (15) is not differentiable, thus we make variable substitutions and solve the following equivalent problem:

$$\begin{aligned} & \min \lambda_1 \|\mathbf{u}\|_1 + \lambda_2 \|\mathbf{v}\|_1 + \|\mathbf{w}\|_1 \\ & \text{s.t., } \mathbf{u} = \nabla_x(\mathcal{D}_\theta(\mathbf{r} - \mathbf{s})), \mathbf{v} = \mathcal{D}_\theta \mathbf{s}, \mathbf{w} = \nabla_y(\mathcal{D}_\theta \mathbf{s}), \end{aligned} \quad (16)$$

where the nonnegative constraint in model (15) is simply implemented via a projection strategy (see details from (26)).

For convenience, we denote the  $\mathcal{D}_\theta(\mathbf{r} - \mathbf{s})$ ,  $\mathcal{D}_\theta\mathbf{r}$  and  $\mathcal{D}_\theta\mathbf{s}$  as  $\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}$ ,  $\mathbf{r}_{\mathcal{D}_\theta}$  and  $\mathbf{s}_{\mathcal{D}_\theta}$ , respectively. Thus the augmented Lagrangian function of problem (16) is given as follows

$$\begin{aligned}\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{s}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = & \\ \lambda_1 \|\mathbf{u}\|_1 + \langle \mathbf{p}_1, \nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u} \rangle + \frac{\beta_1}{2} \|\nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u}\|_2^2 + \lambda_2 \|\mathbf{v}\|_1 + \langle \mathbf{p}_2, \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v} \rangle & (17) \\ + \frac{\beta_2}{2} \|\mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v}\|_2^2 + \|\mathbf{w}\|_1 + \langle \mathbf{p}_3, \nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w} \rangle + \frac{\beta_3}{2} \|\nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w}\|_2^2,\end{aligned}$$

where  $\beta_1$ ,  $\beta_2$  and  $\beta_3$  are regularization parameters, and  $\mathbf{p}_1$ ,  $\mathbf{p}_2$  and  $\mathbf{p}_3$  represent three Lagrange multipliers. ADMM is a quite popular approach for solving  $\ell_1$  problem [36], here we use it to solve the problem (17) that can be decomposed into four simple subproblems. Note that, the four subproblems all have closed-form solutions according to the ADMM scheme.

**a)** The  $\mathbf{u}$ -subproblem is given as follows

$$\begin{aligned}\hat{\mathbf{u}} &= \arg \min_{\mathbf{u}} \lambda_1 \|\mathbf{u}\|_1 + \langle \mathbf{p}_1, \nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u} \rangle + \frac{\beta_1}{2} \|\nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u}\|_2^2, \\ &= \arg \min_{\mathbf{u}} \lambda_1 \|\mathbf{u}\|_1 + \frac{\beta_1}{2} \|\nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u} + \frac{\mathbf{p}_1}{\beta_1}\|_2^2,\end{aligned}\quad (18)$$

which can be solved accurately by soft-thresholding strategy [37] that is described as follows

$$\mathbf{u}^{k+1} = \text{shrink} \left( \nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}^k) + \frac{\mathbf{p}_1^k}{\beta_1}, \frac{\lambda_1}{\beta_1} \right), \quad (19)$$

where  $\text{shrink}(a, b) = \text{sign}(a) \max(|a| - b, 0)$  and

$$\text{sign}(a) = \begin{cases} 1, & a > 0, \\ 0, & a = 0, \\ -1, & a < 0. \end{cases}$$

**b)** The  $\mathbf{v}$ -subproblem is given by minimizing the following function

$$\begin{aligned}\hat{\mathbf{v}} &= \arg \min_{\mathbf{v}} \lambda_2 \|\mathbf{v}\|_1 + \langle \mathbf{p}_2, \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v} \rangle + \frac{\beta_2}{2} \|\mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v}\|_2^2 \\ &= \arg \min_{\mathbf{v}} \lambda_2 \|\mathbf{v}\|_1 + \frac{\beta_2}{2} \|\mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v} + \frac{\mathbf{p}_2}{\beta_2}\|_2^2,\end{aligned}\quad (20)$$

where the closed-form solution of subproblem (20) can be obtained by soft-thresholding strategy that is mentioned in the  $\mathbf{u}$ -subproblem. Its solution has the following form:

$$\mathbf{v}^{k+1} = \text{shrink} \left( \mathbf{s}_{\mathcal{D}_\theta}^k + \frac{\mathbf{p}_2^k}{\beta_2}, \frac{\lambda_2}{\beta_2} \right). \quad (21)$$

**c)** Similarly, the  $\mathbf{w}$ -subproblem is shown as follows

$$\begin{aligned}\hat{\mathbf{w}} &= \arg \min_{\mathbf{w}} \|\mathbf{w}\|_1 + \langle \mathbf{p}_3, \nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w} \rangle + \frac{\beta_3}{2} \|\nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w}\|_2^2 \\ &= \arg \min_{\mathbf{w}} \|\mathbf{w}\|_1 + \frac{\beta_3}{2} \|\nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w} + \frac{\mathbf{p}_3}{\beta_3}\|_2^2,\end{aligned}\quad (22)$$

and the closed-form solution of subproblem (22) is given as follows

$$\mathbf{w}^{k+1} = \text{shrink} \left( \nabla_y \mathbf{s}_{\mathcal{D}_\theta}^k + \frac{\mathbf{p}_3^k}{\beta_3}, \frac{1}{\beta_3} \right). \quad (23)$$

**d)** The  $\mathbf{s}_{\mathcal{D}_\theta}$ -subproblem has the following form,

$$\begin{aligned} \hat{\mathbf{s}}_{\mathcal{D}_\theta} &= \arg \min_{\mathbf{s}_{\mathcal{D}_\theta}} \langle \mathbf{p}_1, \nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u} \rangle + \langle \mathbf{p}_2, \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v} \rangle + \langle \mathbf{p}_3, \nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w} \rangle \\ &\quad + \frac{\beta_1}{2} \|\nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u}\|_2^2 + \frac{\beta_2}{2} \|\mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v}\|_2^2 + \frac{\beta_3}{2} \|\nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w}\|_2^2 \\ &= \arg \min_{\mathbf{s}_{\mathcal{D}_\theta}} \frac{\beta_1}{2} \|\nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}) - \mathbf{u} + \frac{\mathbf{p}_1}{\beta_1}\|_2^2 + \frac{\beta_2}{2} \|\mathbf{s}_{\mathcal{D}_\theta} - \mathbf{v} + \frac{\mathbf{p}_2}{\beta_2}\|_2^2 \\ &\quad + \frac{\beta_3}{2} \|\nabla_y \mathbf{s}_{\mathcal{D}_\theta} - \mathbf{w} + \frac{\mathbf{p}_3}{\beta_3}\|_2^2. \end{aligned} \quad (24)$$

Since the quadratic  $\mathbf{s}_{\mathcal{D}_\theta}$ -subproblem (24) is differentiable, the closed-form solution can be computed easily by the following formula:

$$(\beta_1 \nabla_x^T \nabla_x + \beta_2 \mathbf{I} + \beta_3 \nabla_y^T \nabla_y) \tilde{\mathbf{s}}_{\mathcal{D}_\theta}^{k+1} = \nabla_y^T (\beta_3 \mathbf{w}^{k+1} - \mathbf{p}_3^k) + \nabla_x^T (\beta_1 \nabla_x \mathbf{r}_{\mathcal{D}_\theta} - \beta_1 \mathbf{u}^{k+1} + \mathbf{p}_1^k) + \beta_2 \mathbf{v}^{k+1} - \mathbf{p}_2^k, \quad (25)$$

where  $\mathbf{I}$  is an identity matrix, and Eq. (25) can be solved efficiently by fast Fourier transform (FFT). Due to the nonnegative constraint (14), the resulting  $\tilde{\mathbf{s}}_{\mathcal{D}_\theta}^{k+1}$  is projected by the following formula:

$$\mathbf{s}_{\mathcal{D}_\theta}^{k+1} = \min(\mathbf{r}_{\mathcal{D}_\theta}, \max(\tilde{\mathbf{s}}_{\mathcal{D}_\theta}^{k+1}, \mathbf{0})). \quad (26)$$

**e)** According to ADMM method, we finally update the Lagrange multipliers  $\mathbf{p}_1$ ,  $\mathbf{p}_2$  and  $\mathbf{p}_3$  by the following schemes:

$$\begin{aligned} \mathbf{p}_1^{k+1} &= \mathbf{p}_1^k + \beta_1 (\nabla_x(\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}^{k+1}) - \mathbf{u}^{k+1}), \\ \mathbf{p}_2^{k+1} &= \mathbf{p}_2^k + \beta_2 (\mathbf{s}_{\mathcal{D}_\theta}^{k+1} - \mathbf{v}^{k+1}), \\ \mathbf{p}_3^{k+1} &= \mathbf{p}_3^k + \beta_3 (\nabla_y \mathbf{s}_{\mathcal{D}_\theta}^{k+1} - \mathbf{w}^{k+1}). \end{aligned} \quad (27)$$

The steps **a) - e)** represent one iteration of ADMM which decomposes the difficult minimization problem (15) into four simpler subproblems (*i.e.*,  $\mathbf{u}$ -,  $\mathbf{v}$ -,  $\mathbf{w}$ - and  $\mathbf{s}_{\mathcal{D}_\theta}$ -subproblems). In particular, the four subproblems all have closed-form solutions that promise the fast speed and the accuracy of the algorithm. Thereinto, the  $\mathbf{u}$ -,  $\mathbf{v}$ - and  $\mathbf{w}$ -subproblems are all solved by the soft-thresholding method, and  $\mathbf{s}_{\mathcal{D}_\theta}$ -subproblem is computed by the efficient FFT. We summarize the following Algorithm 1 for the proposed rain removal model (15).

---

**Algorithm 1:** The summarized algorithm for the proposed rain removal model (15)

---

**Input:** Rain image  $\mathbf{r}$ ,  $\lambda_1, \lambda_2, \beta_1, \beta_2, \beta_3$  and pre-defined  $\theta$

**Output:** De-rain image  $\mathbf{t}$

**Initialize:**  $\mathbf{p}_1^0 = \mathbf{p}_2^0 = \mathbf{p}_3^0 = \mathbf{0}, \mathbf{s}^0 = \mathbf{0}$

**While:**  $\|\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}^k\| / \|\mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}^{k-1}\| > tol$  and  $k < M_{iter}$

1) Solve  $\mathbf{u}^{k+1}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}$  by Eqs. (19), (21) and (23)

2) Solve  $\mathbf{s}_{\mathcal{D}_\theta}^{k+1}$  using FFT by Eq. (26)

3) Update the Lagrange multipliers  $\mathbf{p}_1^{k+1}, \mathbf{p}_2^{k+1}$  and  $\mathbf{p}_3^{k+1}$  by Eq. (27)

**EndWhile.**

4) Compute the resulting de-rain image by:

$$\mathbf{t}_{\mathcal{D}_\theta} = \mathbf{r}_{\mathcal{D}_\theta} - \mathbf{s}_{\mathcal{D}_\theta}$$

5) Rotate back  $\mathbf{t}_{\mathcal{D}_\theta}$  to obtain the final de-rain image:  $\mathbf{t}$

---

In Algorithm 1,  $\lambda_1, \lambda_2, \beta_1, \beta_2$  and  $\beta_3$  are some parameters involved in the proposed method.  $tol$  and  $M_{iter}$  represent a positive tolerance value and the maximum number of iterations, respectively. Actually, these parameters are not sensitive to different images, which will be exhibited detailedly in the section of results. Moreover, the ADMM-based Algorithm 1 for the separable convex model (15) guarantees the global optimal (see [38]).

### 3.3 Discussion on the strategy of rotation

Our original de-rain model is formulated by the directional priors that include the horizontal and vertical directions. However, rain streaks are not always along the vertical direction, and they generally have an angle with the vertical direction. Therefore, in the final model (15), we employ a rotational operator  $\mathcal{D}_\theta$  for the issue of rotation. Actually, even though the direction of rain streaks is not along the vertical direction, the proposed method can still work for the rain removal problem and obtain competitive results. However, to get better results, we divide the model into two categories, one is that rain streaks are approximately along the vertical direction, and the other is that rain streaks are obviously far away from the vertical direction.

- The directions of rain streaks are generally within  $90^\circ \pm 15^\circ$ <sup>1</sup>. Empirically, it is not necessary to rotate the rainy image for this case, because the proposed framework can still obtain state-of-the-art results, such as the first five examples in Fig. 6.
- When the directions of rain streaks are far away from the range of  $90^\circ \pm 15^\circ$ <sup>2</sup>, the rainy image needs to be rotated such that the rain streaks are roughly along the vertical direction. A flowchart of the strategy of rotation is introduced in Fig. 4. The rain streaks only need to be roughly in the vertical direction, which is actually easy to achieve<sup>3</sup>.

## 4 Results

In this section, we compare the proposed method with two recent state-of-the-art rain removal approaches. One uses a dictionary learning based algorithm for single image rain removal [2] (denoted as

<sup>1</sup>The vertical direction is viewed as  $90^\circ$

<sup>2</sup>In most of cases, the directions of rain streaks fall into the range  $90^\circ \pm 15^\circ$ , except that when heavy wind coming. Therefore, our method works well for most of real rainy cases.

<sup>3</sup>The rotation can be easily implemented by the command of “imrotate” in MATLAB.

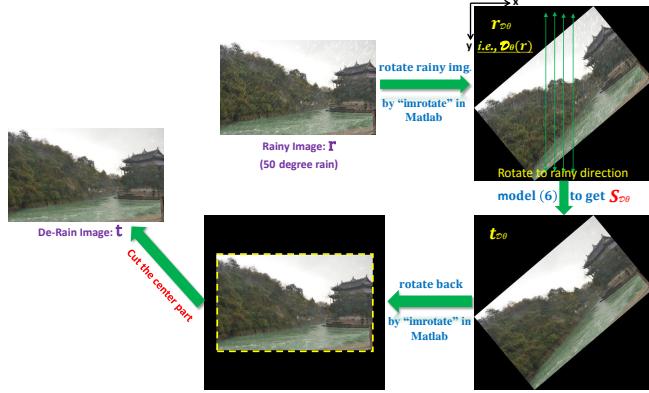


Figure 4: The flowchart of our rotation strategy.

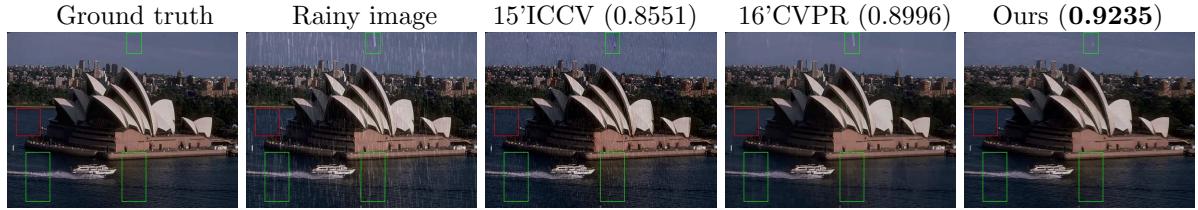


Figure 5: Visual results and the corresponding SSIM of different compared methods. Our method almost removes all rain streaks while “15’ICCV” and “16’CVPR” leave significant rain streaks (see green boxes). Moreover, our method preserves more details of background image, while “16’CVPR” smoothens the details (see red boxes). Readers are recommended to zoom in all figures for better visibility.

“15’ICCV”)<sup>4</sup>, the other employs a minimization model with the learned rain layer prior for single image rain removal [4] (denoted as “16’CVPR”)<sup>5</sup>. Our experiments are implemented in MATLAB(R2016a) on a desktop of 16Gb RAM and Intel(R) Core(TM) CPU i5-4590: @3.30GHz. Readers can find the Matlab code (p-code) to test the performance of our method<sup>6</sup>

Since humans are more sensitive to the changes of luminance, we first convert RGB rainy images to YUV space which is very popular in image/video processing, and then only conduct our algorithm on the luminance channel (Y) to remove rain streaks. For fair comparisons, the two compared methods are also implemented on the luminance channel. In addition, two kinds of data, *i.e.*, synthetic data and real data, are utilized for the experiments. For the synthetic data, we use root-mean-square error (RMSE), peak signal-noise ration (PSNR) and structural similarity (SSIM)<sup>7</sup> [39] on illuminance channel to estimate the performance of different methods. RMSE is defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (h_i - \hat{h}_i)^2}, \quad (28)$$

where  $h, \hat{h}$  are the vector-form of ground-truth image and the estimated de-rain image, respectively.  $N$  represents the total number of image pixels.

For the parameters in the proposed Algorithm 1, we empirically set  $\lambda_1 = 0.95$ ,  $\lambda_2 = 0.08$ ,  $\beta_1 = \beta_2 = \beta_3 = 200$ ,  $\text{tol} = 10^{-3}$ , and  $M_{\text{iter}} = 400$  for synthetic data and only change  $\beta_1 = \beta_2 = \beta_3 = 100$  for real data. Note that, if tuning parameters finely for different images may get better results, however, we unify parameters here to exhibit the stability of the proposed method. For the parameters of “15’ICCV” and “16’CVPR”, we keep the default settings of the provided code.

**Rain streaks generation:** The rain model (1) is an additive one which is a popular type used in many works, *e.g.*, [16, 4]. For the simulation of rain streaks, the literature [16] mentions that the rain streaks are usually generated by Photoshop<sup>89</sup>. Referring to this literature, the rain streaks are generated by the following steps: 1) Add salt&pepper noise to a zero matrix with the same size as the ground-truth image. The added noise is with random density  $den^{10}$ . The bigger  $den$  is, the heavier the synthetic rain will be. 2) Convolute a motion blur kernel with the added noise in 1) to generate the rain streaks. The motion blur kernel is with two parameters, *i.e.*,  $len$  and  $theta^{11}$  which respectively control the magnitude and the direction of the generated rain streaks. 3) Add the rain streaks to the ground-truth image to obtain the final simulated rainy image.

## 4.1 Results on synthetic data

The visual and quantitative results for the synthetic image “Sydney” are presented in Fig. 5, from which, we can see that the proposed method almost removes all rain streaks (see green boxes) and preserves the details of background image well (see red boxes). Nevertheless, the method “15’ICCV” fails to remove rain streaks completely and obtains unsatisfying SSIM. Although the results of “16’CVPR” indicate that the method “16’CVPR” performs better than the method “15’ICCV”, it still can not outperform

---

<sup>4</sup>Code available on the site: <http://www.math.nus.edu.sg/~matjh/research/research.htm>

<sup>5</sup>The authors provide the corresponding de-rain code.

<sup>6</sup><http://www.escience.cn/people/dengliangjian/codes.html>. Upon acceptance of this paper we will make the Matlab source code available to provide more details of the implementation.

<sup>7</sup><https://ece.uwaterloo.ca/~z70wang/research/ssim/>

<sup>8</sup><http://photo-tutorial.ws/photoshop-tutorials/photo-effects/rain.html>

<sup>9</sup><https://www.photoshopessentials.com/photo-effects/rain/>

<sup>10</sup>Matlab command: `imnoise('img', 'salt&pepper', den)` where ‘img’ is the input image.

<sup>11</sup>Matlab command: `fspecial('motion', len, theta)`.

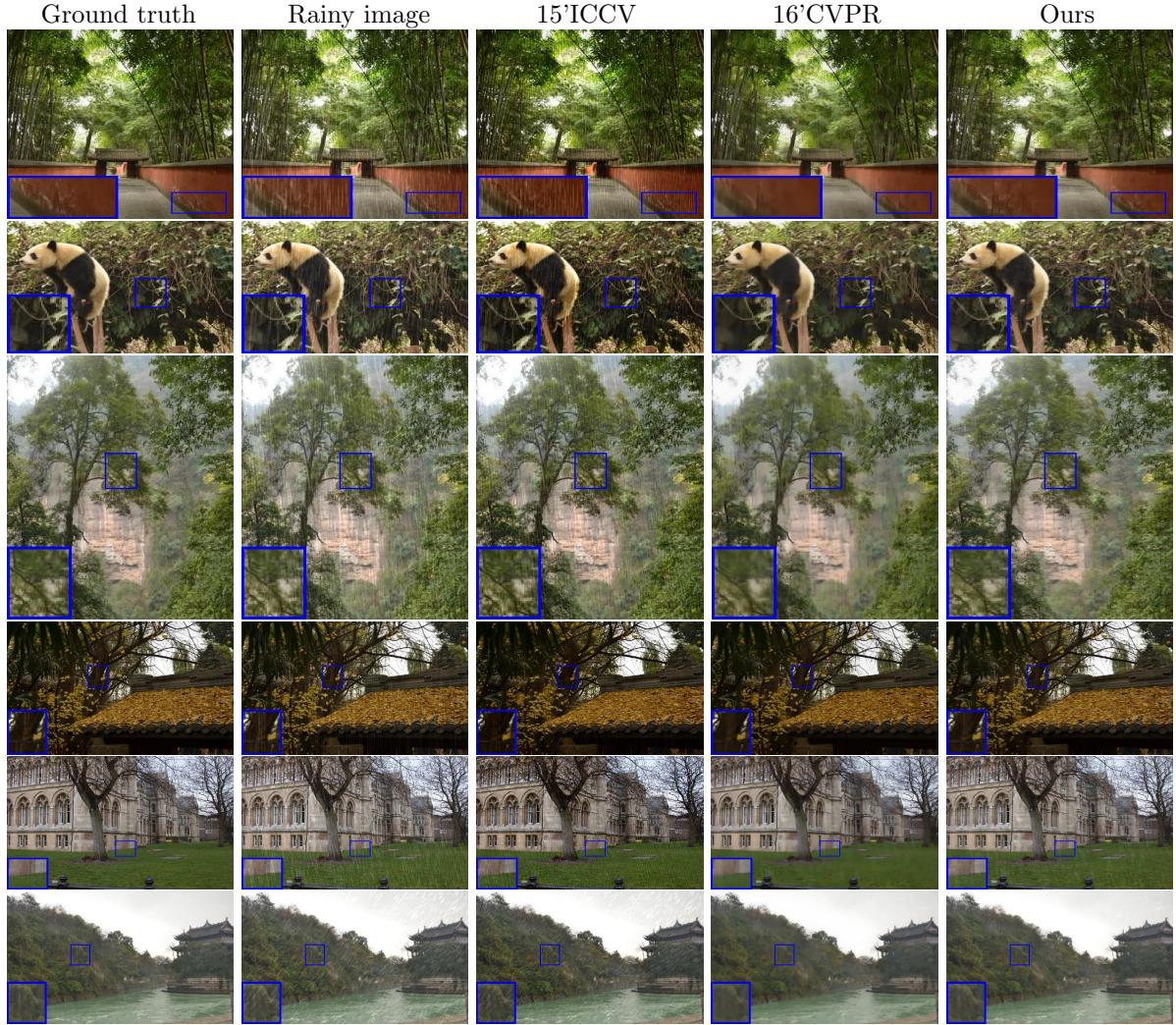


Figure 6: Visual results on synthetic images (rain's orientation): “bamboo” ( $90^\circ$ ), “panda” ( $85^\circ$ ), “tree” ( $95^\circ$ ), “roof” ( $90^\circ$ ), “building” ( $99^\circ$ ) and “river” ( $50^\circ$ ). The method “15'ICCV” can not remove rain streaks completely. “16'CVPR” removes rain streaks well, but over-smooth background details significantly. The proposed method not only removes rain streaks better, but also preserves the details of background images well. Quantitative results can be found in Tab. 1.

the proposed method, both visually and quantitatively. Note that the experimental image in Fig. 5 is obtained from Li *et al.*'s paper [4].

We also list visual results for more synthetic images in Fig. 6. The rain streaks in these synthetic images are with different orientations, *e.g.*,  $90^\circ$ ,  $85^\circ$  and  $50^\circ$ , etc. From this figure, we know that the method “15'ICCV” remains rain streaks significantly (see boxes in the figure). Although “16'CVPR” removes rain streaks well, it may over-smooth background images. The proposed method not only removes rain streaks completely, but also preserves more details of background images. In particular, the rain streaks of image “bamboo” estimated by different compared methods are presented in Fig.

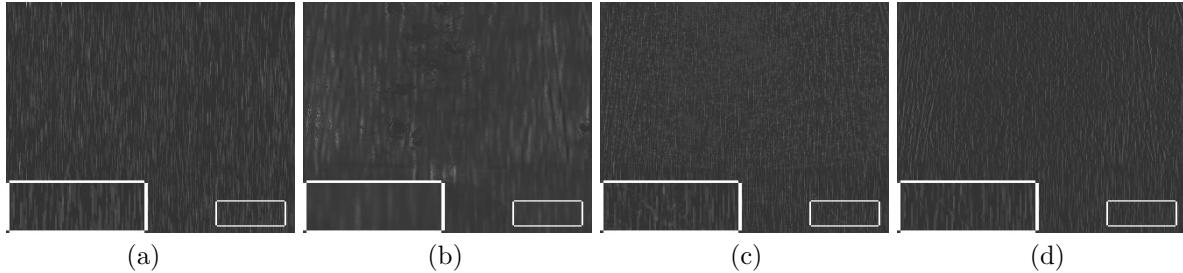


Figure 7: Comparison on rain streaks for the example “bamboo”. (a) True rain streaks; (b)-(d) Estimated rain streaks by “15’ICCV”, “16’CVPR” and the proposed method. Here we add intensity value 51 to rain streaks for better visibility (intensity range [0, 255]).

7. From the figure, it is clear that our method gets more consistent rain streaks than “15’ICCV” and “16’CVPR” comparing with the ground-truth rain streaks Fig. 7(a). Moreover, our method also obtains more excellent quantitative results, *i.e.*, PSNR, SSIM and RMSE, both for background images and the corresponding rain streaks (see Tab. 1). In Tab. 1, we present quantitative results for two cases, *i.e.*, heavy rain case and light rain case, and Fig. 6 exhibits the visual results for the case where rain streaks are heavy.

Especially, our model is based on the assumption that rain streaks generally come down from top to bottom (*i.e.*, almost along the vertical direction of rainy images). When rain streaks are not along the vertical direction obviously (see the sixth example in Fig. 6), we should rotate the rainy image to make rain streaks to be approximately along the vertical direction, and then apply the proposed model to the rotated rainy image, see the discussion on rotation in Section 3.3. Since the image rotation has to involve pixels interpolation, our quantitative performance after image rotation is worse than the method “16’CVPR”, but the visual quality of our method still outperforms other methods (see boxes for better visibility).

Furthermore, we also employ the “UCID” dataset<sup>12</sup> to evaluate the performance of the three compared methods. Tab. 2 shows the average quantitative performance of 38 images from UCID dataset<sup>13</sup>, including average PSNR, SSIM, RMSE and the corresponding standard derivations. The compared methods are respectively implemented on background image and rain streaks when adding different rain streaks. From Tab. 2, the proposed method outperforms other two state-of-the-art methods on different rainy images. It not only obtains the best PSNR, SSIM and RMSE, but also generates the smallest standard derivations for almost all examples, which indicates that our method is more stable than “15’ICCV” and “16’CVPR”.

## 4.2 Results on real data

We also employ some real rainy images to test the performance of different methods. In Fig. 8, the first and third examples are from the literatures of “16’CVPR” and “15’ICCV”, respectively, and the second example is obtained from internet.

Fig. 8 presents visual results of the three compared methods on real images. Once again, the proposed method shows the best ability of removing rain streaks, in the meanwhile, it also retains the details of background images better than the other two compared methods.

<sup>12</sup><http://homepages.lboro.ac.uk/~cogs/datasets/ucid/ucid.html>

<sup>13</sup>Here, we only select the first 38 images of UCID dataset for evaluation, since too many images will result in significant computation increasing due to the low speed of “15’ICCV” and “16’CVPR”.

Table 1: Quantitative comparisons of Fig. 6 (Bold: the best one).

Rain type	Heavy									Light								
	Image	Method	Background (T)			Streak (S)			Time (s)	Background (T)			Streak (S)			Time (s)		
			PSNR	SSIM	RMSE	PSNR	SSIM	RMSE		PSNR	SSIM	RMSE	PSNR	SSIM	RMSE			
<b>bamboo</b>	15'ICCV	27.74 0.8289 10.460	27.73 0.4227 10.472	69.88	28.81 0.9110 9.251	28.80 0.3903 9.256	78.10											
	16'CVPR	30.52 0.8950 7.593	30.64 0.6706 7.488	960.86	30.44 0.9223 7.661	30.49 0.4984 7.620	973.74											
	Proposed	<b>30.95</b> <b>0.9176</b> <b>7.227</b>	<b>30.96</b> <b>0.6989</b> <b>7.221</b>	<b>0.873</b>	<b>32.90</b> <b>0.9476</b> <b>5.773</b>	<b>32.90</b> <b>0.6603</b> <b>5.775</b>	<b>0.64</b>											
<b>panda</b>	15'ICCV	27.84 0.8931 10.339	27.78 0.4541 10.410	58.23	28.02 0.9406 10.118	28.01 0.3583 10.134	44.27											
	16'CVPR	30.57 0.9128 7.555	30.66 0.6244 7.475	709.90	30.79 0.9396 7.366	30.82 0.4899 7.337	632.32											
	Proposed	<b>30.98</b> <b>0.9324</b> <b>7.219</b>	<b>30.91</b> <b>0.6607</b> <b>7.264</b>	<b>0.649</b>	<b>33.33</b> <b>0.9568</b> <b>5.495</b>	<b>33.32</b> <b>0.6539</b> <b>5.501</b>	<b>0.45</b>											
<b>tree</b>	15'ICCV	31.00 0.9081 7.186	30.84 0.5526 7.317	67.89	32.27 0.9554 6.213	32.25 0.4916 6.223	76.54											
	16'CVPR	30.87 0.8829 7.294	30.94 0.6227 7.237	1032.60	29.91 0.8851 8.153	29.96 0.4250 8.105	1131.30											
	Proposed	<b>32.41</b> <b>0.9358</b> <b>6.112</b>	<b>32.31</b> <b>0.6957</b> <b>6.183</b>	<b>0.771</b>	<b>35.55</b> <b>0.9673</b> <b>4.258</b>	<b>35.42</b> <b>0.7296</b> <b>4.319</b>	<b>0.38</b>											
<b>roof</b>	15'ICCV	29.96 0.9478 8.103	29.95 0.5658 8.113	80.36	28.84 0.9332 9.212	28.81 0.4685 9.243	93.65											
	16'CVPR	32.77 0.9638 5.861	32.83 0.5914 5.819	1511.22	30.91 0.9329 7.257	30.98 0.5158 7.198	1406.20											
	Proposed	<b>36.46</b> <b>0.9741</b> <b>3.834</b>	<b>36.49</b> <b>0.7889</b> <b>3.820</b>	<b>0.921</b>	<b>35.32</b> <b>0.9681</b> <b>4.373</b>	<b>35.24</b> <b>0.7348</b> <b>4.408</b>	<b>0.952</b>											
<b>building</b>	15'ICCV	26.66 0.8782 11.844	26.66 0.4335 11.845	114.70	26.35 0.9273 12.267	26.36 0.3465 12.252	110.47											
	16'CVPR	30.95 0.8849 7.228	31.15 0.6547 7.076	1072.1	30.30 0.8910 7.786	30.40 0.4582 7.694	1292.50											
	Proposed	<b>31.29</b> <b>0.9354</b> <b>6.945</b>	<b>31.29</b> <b>0.6298</b> <b>6.953</b>	<b>1.236</b>	<b>34.41</b> <b>0.9635</b> <b>4.848</b>	<b>34.43</b> <b>0.7076</b> <b>4.840</b>	<b>0.97</b>											
<b>river</b>	15'ICCV	31.56 0.8618 6.737	30.97 0.4073 7.213	99.67	33.61 0.9152 5.310	33.01 0.5203 5.700	98.33											
	16'CVPR	<b>35.62</b> <b>0.9320</b> <b>4.222</b>	<b>34.74</b> 0.6568 <b>4.674</b>	902.42	<b>36.29</b> <b>0.9370</b> <b>3.905</b>	<b>35.75</b> 0.7078 <b>4.155</b>	862.40											
	Proposed	33.66 0.9103 5.290	34.59 0.7053 4.751	<b>1.175</b>	34.69 0.9166 4.915	34.02 0.7315 4.371	<b>0.573</b>											

Table 2: Average quantitative performance (with corresponding standard deviation) of different methods on 38 test images of UCID dataset with different simulated streak types (Bold: the best one).

Rain type	Method	Background (T)			Streak (S)			Time (s)
		PSNR	SSIM	RMSE	PSNR	SSIM	RMSE	
<i>den=0.04</i>	15'ICCV	26.01±1.23	0.8289±0.0524	12.882±1.755	26.38±1.02	0.3576±0.0703	12.315±1.460	114.95
	16'CVPR	30.01±1.67	0.8958±0.0358	8.202±1.556	30.31±1.39	0.6416±0.0932	7.874±1.228	1386.38
	Proposed	<b>30.44±0.89</b>	<b>0.8991±0.0217</b>	<b>7.707±0.796</b>	<b>30.67±0.78</b>	<b>0.6805±0.0586</b>	<b>7.490±0.657</b>	<b>1.32</b>
<i>den=[0.02, 0.1]</i>	15'ICCV	25.73±1.44	0.7620±0.0771	13.357±2.088	25.85±1.47	0.3486±0.0747	13.178±2.128	78.73
	16'CVPR	29.13±1.71	0.8738±0.0426	9.072±1.687	29.14±1.73	0.6347±0.1035	9.062±1.706	1489.45
	Proposed	<b>29.32±1.89</b>	<b>0.8857±0.0274</b>	<b>8.917±1.851</b>	<b>29.32±1.90</b>	<b>0.6731±0.0739</b>	<b>8.916±1.860</b>	<b>1.41</b>
<i>len=[10, 60]</i>	15'ICCV	28.94±1.88	0.9020±0.0378	9.318±1.991	29.47±1.90	0.5853±0.0745	8.767±1.903	73.24
	16'CVPR	32.09±1.81	0.9298±0.0273	6.472±1.366	32.64±1.62	<b>0.6925±0.0808</b>	6.051±1.154	1452.37
	Proposed	<b>32.55±0.88</b>	<b>0.9428±0.0212</b>	<b>6.038±0.624</b>	<b>33.16±0.82</b>	0.6229±0.0640	<b>5.626±0.545</b>	1.09

Similar to the results on synthetic data, the defects of “15’ICCV” and “16’CVPR” still exist. The method “15’ICCV” leaves too many rain streaks in the background images. Furthermore, the method “16’CVPR” removes rain streaks well, however, it will significantly over-smooth the details of background images (see boxes in Fig. 8).

### 4.3 Computation

Our algorithm is a quite efficient method for image rain removal problem. In Algorithm 1, the four subproblems all have closed-form solutions which can be computed directly or by the efficient FFT algorithm. The computation complexity of our algorithm is about  $\mathcal{O}(n \cdot \log n)$  where  $n = MN$ . From Tab. 1 and Tab. 2, we know that the proposed method outperforms the other two compared methods significantly from the aspect of computation time. For instance, for the “panda” image in Fig. 6 (size  $229 \times 305 \times 3$ ), the proposed method only costs about 0.45 second while “15’ICCV” and “16’CVPR” take about 44 seconds and 632 seconds, respectively. Note that, the code of “16’CVPR” utilizes a strategy of downsampling to reduce computation, however, here we do not take this strategy and use full-size image

Table 3: Average quantitative performance on a new dataset that synthesizes the rainy images by the photorealistic rendering techniques [40] (Bold: the best one).

Method	PSNR	SSIM	RMSE	Time (s)
15'ICCV	$30.10 \pm 2.72$	$0.8671 \pm 0.0647$	$8.310 \pm 2.404$	108.81
16'CVPR	$32.15 \pm 1.62$	$0.9152 \pm 0.0264$	$6.400 \pm 1.202$	1646.30
Proposed	<b><math>33.18 \pm 3.58</math></b>	<b><math>0.9365 \pm 0.0267</math></b>	<b><math>6.033 \pm 2.497</math></b>	<b>1.22</b>



Figure 8: Visual results on real images. Our method preserves more image details and removes rain streaks better.

for experiments, which results in more computation time but gets better quantitative performance.

#### 4.4 More discussions

**The performance of each image (38 images) in UCID dataset:** In the synthetic experiments, 38 images from UCID dataset are utilized to evaluate the ability of rain streaks removal of all compared methods<sup>14</sup>. In Fig. 9, we present PSNR, SSIM and RMSE of each image for the three compared methods. According to this figure, our method performs best for almost all images. The number of best performance (NBP) for PSNR are 28, 10, 0 for the proposed method, “16’CVPR” and “15’ICCV”, respectively, which means that the proposed method reaches the best PSNR on 28 test images within total 38 images and the method “16’CVPR” gets the best PSNR on 10 test images. Similarly, the NBP for SSIM and RMSE are 31, 7, 0 and 28, 10, 0, respectively. It is clear that our method gets the largest NBP. In particular, from Fig. 9, we know that the blue points of our method oscillate smoothly, which

<sup>14</sup>For convenience, the 38 rainy images employed for comparisons are the same with the third case of Tab. 2.

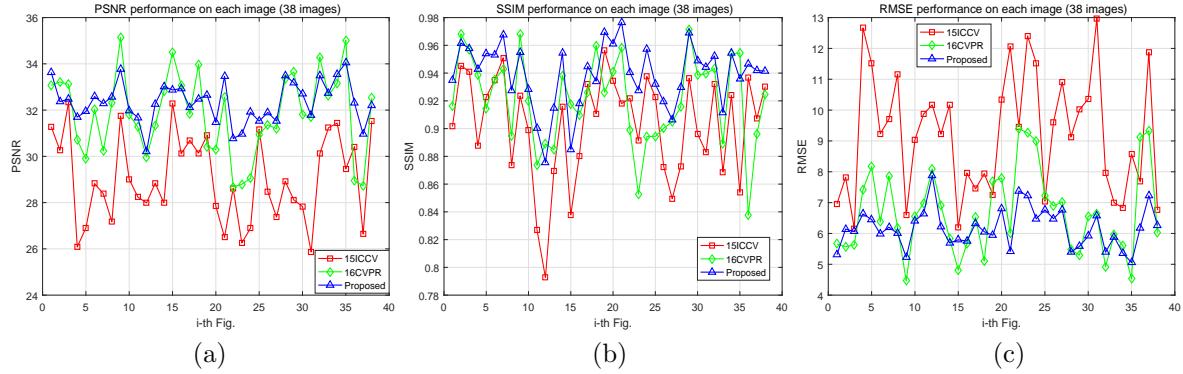


Figure 9: (a) The PSNR performance on each image (38 images from UCID dataset), and the number of best performance (NBP) is **28, 10, 0** for the proposed method, 16CVPR and 15ICCV, respectively; (b) The SSIM performance on each image, and NBP is **31, 7, 0**, respectively; (c) The RMSE performance on each image, and NBP is **28, 10, 0**, respectively.

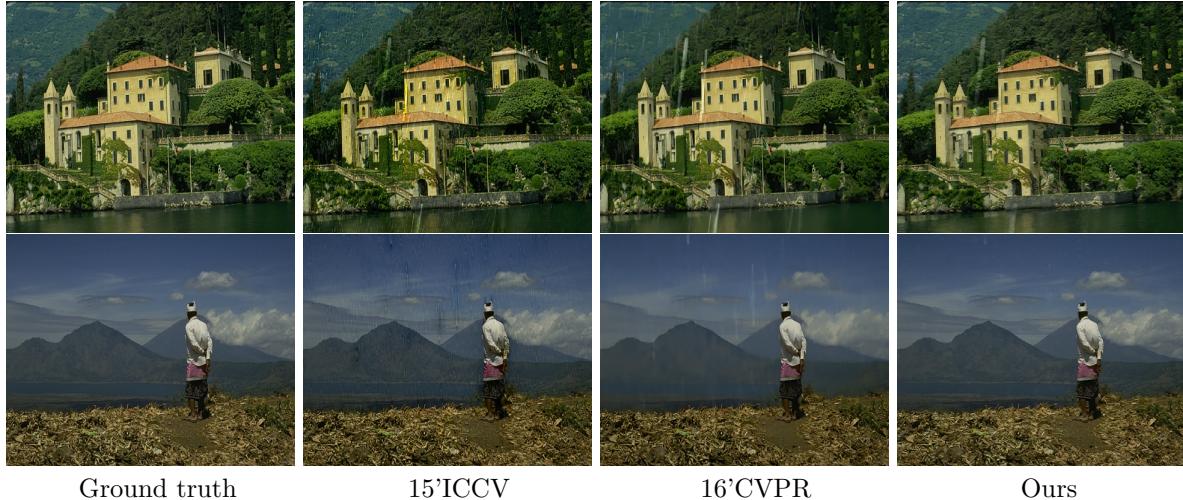


Figure 10: The visual comparisons on a new dataset that synthesizes the rainy images by the photorealistic rendering techniques [40]. To save space, here we only exhibit two of all 12 results.

also demonstrates that the proposed method is more stable (see the standard derivations in Tab. 2).

**The performance on a new dataset (12 images):** Here, we also show more comparisons on a new dataset<sup>15</sup> that synthesizes the rainy images by the photorealistic rendering techniques proposed by Grag and Shree [40]. From Fig. 10 and Tab. 3, it is easy to know that the proposed method obtains the better results once again than the other two state-of-the-art rain removal methods, which demonstrates the removing capability of the given method for different rain streaks. Note that, for this dataset, we only need to change  $\beta_i, i = 1, 2, 3, 4$  to 70, and in the meanwhile keep other parameters unchanged.

<sup>15</sup>[http://yu-li.github.io/paper/li\\_cvpr16\\_rain.zip](http://yu-li.github.io/paper/li_cvpr16_rain.zip)

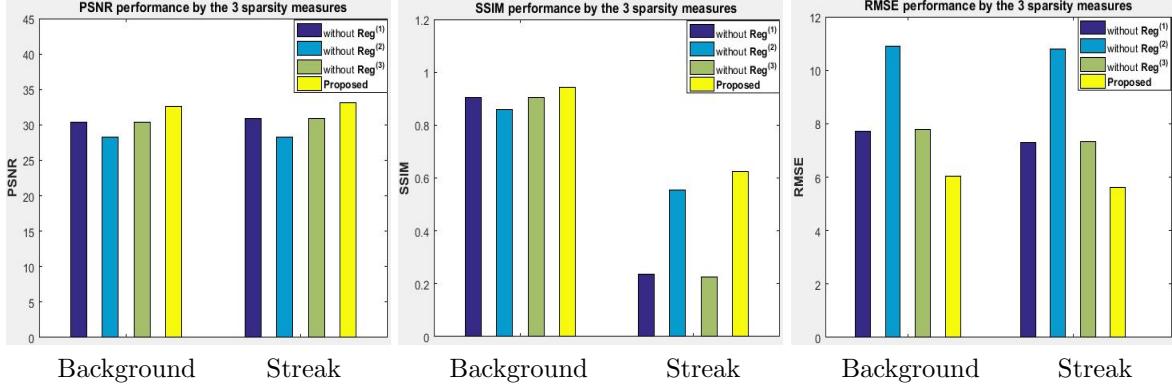


Figure 11: PSNR, SSIM and RMSE of background image and rain streaks obtained by discarding one of the 3 sparse terms (*i.e.*, the average performance on the 38 images of UCID dataset).



Figure 12: The visual results on the effect of three terms.

**The effect of different terms in our model:** The proposed model (15) involves three sparse terms which can depict the directional latent and intrinsic properties of rainy images. To reveal the effect of the three terms, we give some experimental analysis to learn which term contributes to rain removal to the most extent. Fig. 11 shows the rain removal results, in terms of PSNR, SSIM and RMSE, on 38 images of UCID dataset after discarding the sparse term  $\mathbf{Reg}^{(1)}$ ,  $\mathbf{Reg}^{(2)}$  and  $\mathbf{Reg}^{(3)}$ , respectively. From the figure, the proposed model that combines all three terms performs best (see yellow bars in Fig. 11), while other models that discard one sparse term all get worse quantitative results. In particular, the second sparse term  $\mathbf{Reg}^{(2)}$  affects the results to the most extent (see purple bars), while the first term  $\mathbf{Reg}^{(1)}$  and the third term  $\mathbf{Reg}^{(3)}$  affect the final quantitative results almost equivalently, see blue and green bars in Fig. 11. Note that, after discarding the first term  $\mathbf{Reg}^{(1)}$ , the model actually changes to the UTV model. In addition, Fig. 12 shows the visual results about the effect of three terms.

**The issue of parameters:** The proposed method mainly involves five parameters, *i.e.*,  $\lambda_1$ ,  $\lambda_2$ ,  $\beta_1$ ,  $\beta_2$  and  $\beta_3$ . Since we set the same  $\beta_1$ ,  $\beta_2$  and  $\beta_3$  in our experiments, here we only discuss the selection of  $\lambda_1$ ,  $\lambda_2$  and  $\beta_1$ . Fig. 13 presents the quantitative results of one image selected from the 38 test images with varying  $\lambda_1$ ,  $\lambda_2$  and  $\beta_1$ <sup>16</sup>. From the results, it is clear that the PSNR values of background images

<sup>16</sup>When one parameter varies, the other parameters are fixed to the default setting  $\lambda_1 = 0.95$ ,  $\lambda_2 = 0.08$  and  $\beta_1 = \beta_2 = \beta_3 = 200$ .

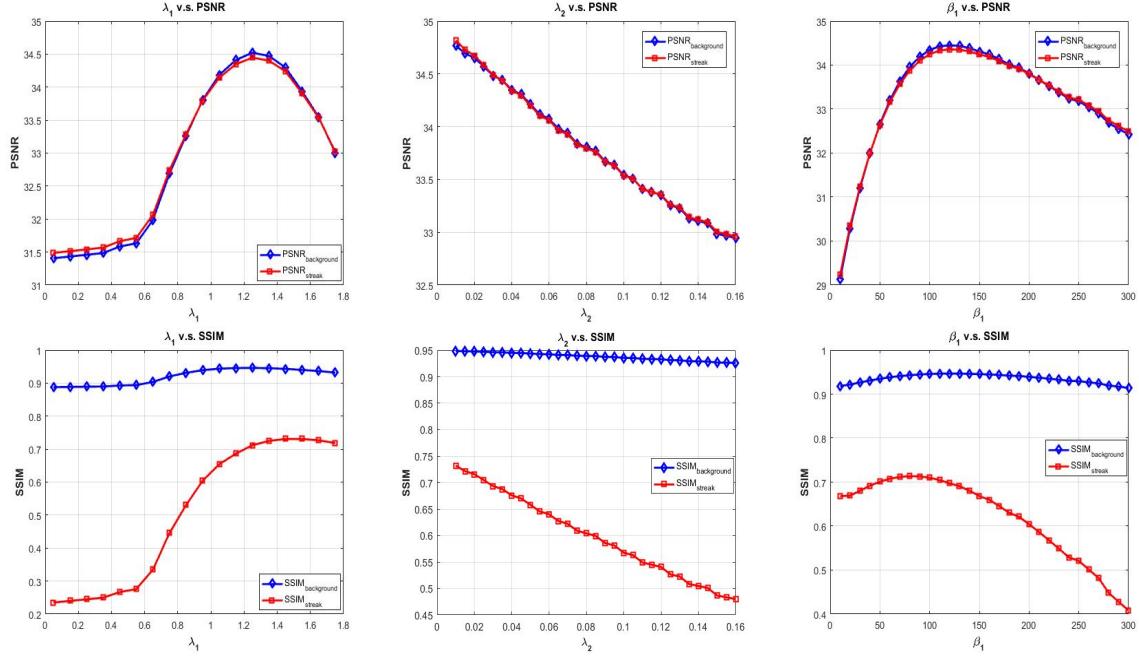


Figure 13: First row: PSNR curves as functions of the parameters  $\lambda_1$ ,  $\lambda_2$  and  $\beta_1$ ; Second row: SSIM curves as functions of the parameters  $\lambda_1$ ,  $\lambda_2$  and  $\beta_1$ . Note that, due to setting  $\beta_1 = \beta_2 = \beta_3$  in the experiments, thus the last column also reflects the relation  $\beta_2$  v.s. PSNR, SSIM, and  $\beta_3$  v.s. PSNR, SSIM.

are almost consistent with those of rain streaks with varying parameters. Moreover, although the SSIM values of background images are generally higher than those of rain streaks, the variation trend of SSIM values are almost the same, which indicates the consistency of the influence of parameters.

Specifically, the best parameter selection of our method for the related image in Fig. 13 is about  $\lambda_1 = 1.2$ ,  $\lambda_2 = 0.01$  and  $\beta_1 = \beta_2 = \beta_3 = 120$ , while in this paper we set  $\lambda_1 = 0.95$ ,  $\lambda_2 = 0.08$  and  $\beta_1 = \beta_2 = \beta_3 = 200$  for all experiments. Since although tuning parameters finely for different images may get better results, we tend to unify the parameters of all experiments to exhibit the stability of the proposed method. Therefore, here we select  $\lambda_1 = 0.95$ ,  $\lambda_2 = 0.08$  and  $\beta_1 = \beta_2 = \beta_3 = 200$ , which can get relatively good results for most of test images. In addition, although the smaller  $\lambda_2$  may result in higher PSNR and SSIM, the visual results stay poor performance. Empirically, here we set  $\lambda_2$  as 0.08.

**The proposed model v.s. TV-based model:** The proposed model (15) is similar to the anisotropic TV-based model if we add one sparse term  $\lambda_3 \|\nabla_y(\mathcal{D}_\theta(\mathbf{r} - \mathbf{s}))\|_1$ . Thus it is necessary to illustrate the difference between the TV-based model and the proposed model. Fig. 14 presents the average quantitative performance of our model and TV-based model on the 38 test images. We can see that the proposed model performs slightly better than TV-based model when  $\lambda_3$  is relatively large, in the meanwhile, when  $\lambda_3$  tends to be small, the TV-based model performs almost as well as the proposed model does. In summary, TV-based model leads to one more parameter  $\lambda_3$  but can not get better results than our directional model, therefore, our model is a better choice than TV-based model for image rain streaks removal task.

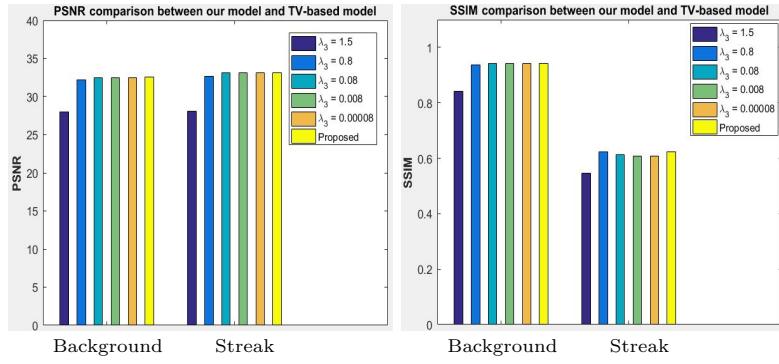


Figure 14: Average quantitative performance of our model and TV-based model (with different  $\lambda_3$ ) on the 38 images from UCID dataset. Note that our model performs slightly better than TV-based model when  $\lambda_3$  is relatively large, while TV-based model performs approximately as well as our model does when  $\lambda_3$  tends to be small.

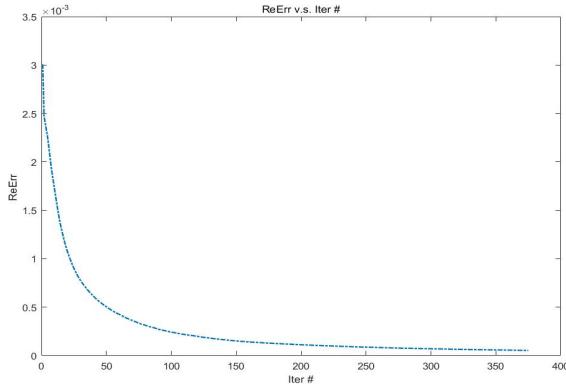


Figure 15: The convergence curve of the proposed algorithm.

**The convergence curve:** To illustrate the convergence property of the given algorithm, in Fig. 15, we show the curve between the iteration number and the relative error that is defined as  $\frac{\|\mathbf{t}^{k+1} - \mathbf{t}^k\|_2}{\|\mathbf{t}^{k+1}\|_2}$  where  $\mathbf{t}^{k+1}$  and  $\mathbf{t}^k$  are the last and the previous outcome, respectively. From this figure, it is easy to see that the given algorithm can converge with the iteration number increasing.

**Applying the dehaze procedure to the derain outcome:** For the case of heavy rain, the obtained de-rain images sometimes contain the high intensity appearance which makes the results holding undesired contrast. Like the previous strategies, see for instance [4], we utilize the dehaze method [41] as the post-processing to get the enhanced de-rain output. Two examples are shown in Fig. 16, in which we apply the dehaze strategy to the first two de-rain images addressed by our approach in Fig. 8. It demonstrates the effectiveness of the dehaze strategy for the de-rain image.

**The capability of removing horizontal and vertical line structures:** In Fig. 17, we simulate a toy example to exhibit the rain removal property of the given model. In this figure, the added horizontal (yellow circle) and vertical (green circle) lines in the rainy image can be viewed as a part of the rain-free



Figure 16: Two examples of taking the dehaze method as the post-processing.



Figure 17: (a) A toy example with simulated horizontal (yellow circle) and vertical (green circle) lines; (b) The recovered results by the proposed method, which shows that our model does not affect the horizontal lines, but removes the vertical lines effectively.

image. Here we use these added lines to test the capability of the proposed method for the removal of the horizontal and vertical structures. Fig. 17(b) shows that the proposed method does not affect the horizontal lines, but removes the vertical lines. In summary, the given model sometimes removes the vertical structures of rain-free image wrongly, which can be viewed as a drawback of this method. However, these vertical and independent line structures do not usually appear in a natural image. Thus in general our method will not affect the image quality seriously.

## 5 Conclusion

We have proposed a simple but efficient method based on a unidirectional global sparse model for rain streaks removal problem. For the proposed convex model involving three sparse priors, we designed a ADMM-based algorithm which guarantees the global optimal to solve it. Results on synthetic and real data demonstrated that the proposed method not only removed rain streaks well, but also preserved more details of background images. It outperformed two recent state-of-the-art rain removal methods

for almost all examples, both visually and quantitatively. Moreover, the computation of the proposed method was much less than the two compared methods, because the computational complexity of our method is only  $\mathcal{O}(n \cdot \log n)$ .

However, our method also has certain limitations. In the case where rain streaks are far away from the vertical direction, rainy images must be rotated by a user's estimation to make the rain streaks to be roughly in the vertical direction. In addition, the proposed method may remove the vertical lines of the background image, despite that generally there are few vertical lines in the rainy image. Moreover, in the case of heavy rain, which normally incorporates haze and rain streaks, our method can not remove them simultaneously to recover a clear de-haze and de-rain image. In the future, we intend to incorporate an orientation estimation technique into the proposed method to automatically remove rain streaks far away from the vertical direction. Furthermore, we will incorporate de-haze and de-rain processes into a uniform model for the case of heavy rain.

## Acknowledgment

The authors would like to thank Dr. Li who is the first author of the method “16’CVPR” for providing their image de-rain code, so that we can present their excellent results in this paper. Moreover, thank Mr. Zi-Yao Zhang to revise the language of this paper. In addition, the authors also thank the support by NSFC (61702083, 61772003) and Fundamental Research Funds for the Central Universities (ZYGX2016KYQD142, ZYGX2016J132, ZYGX2016J129).

## References

- [1] K. Garg and S. K. Nayar, “Detection and removal of rain from videos,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [2] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” *International Conference on Computer Vision (ICCV)*, 2015.
- [3] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2341–2353, 2011.
- [4] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain Streak Removal Using Layer Priors,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [5] K. Garg and S. K. Nayar, “When does a camera see rain?,” *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 1067–1074, 2005.
- [6] K. Garg and S. K. Nayar, “Vision and rain,” *International Journal of Computer Vision*, vol. 75, pp. 3–27, 2007.
- [7] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, “Rain removal in video by combining temporal and chromatic properties,” *IEEE International Conference on Multimedia and Expo (ICME)*, 2006.
- [8] P. C. Barnum, S. Narasimhan, and T. Kanade, “Analysis of rain and snow in frequency space,” *International Journal of Computer Vision*, vol. 86, pp. 256–274, 2010.
- [9] 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*, vol. 93, pp. 348–367, 2011.

- [10] Y. L. Chen and C. T. Hsu, “A generalized low-rank appearance model for spatio-temporally correlated rain streaks,” *International Conference on Computer Vision (ICCV)*, pp. 1968–1975, 2013.
- [11] V. Santhaseelan and V. K. Asari, “Utilizing local phase information to remove rain from video,” *International Journal of Computer Vision*, pp. 1–19, 2014.
- [12] A. E. Abdel-Hakim, “A novel approach for rain removal from videos using low-rank recovery,” *IEEE International Conference on Intelligent Systems, Modelling and Simulation*, pp. 351–356, 2014.
- [13] A. K. Tripathi and S. Mukhopadhyay, “Removal of rain from videos: a review,” *Signal, Image and Video Processing*, vol. 8, pp. 1421–1430, 2014.
- [14] J. H. Kim, J. Y. Sim, and C. S. Kim, “Video deraining and desnowing using temporal correlation and low-rank matrix completion,” *IEEE Transactions on Image Processing*, vol. 24, pp. 2658–2670, 2015.
- [15] 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,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [16] L. W. Kang, C. W. Lin, and Y. H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” *IEEE Transactions on Image Processing*, vol. 21, pp. 1742–1755, 2012.
- [17] J. H. Kim, C. Lee, J. Y. Sim, and C. S. Kim, “Single-image deraining using an adaptive nonlocal means filter,” *IEEE International Conference on Image Processing (ICIP)*, 2013.
- [18] S. H. Sun, S. P. Fan, and Y. C. F. Wang, “Exploiting image structural similarity for single image rain removal,” *IEEE International Conference on Image Processing (ICIP)*, 2014.
- [19] S. C. Pei, Y. T. Tsai, and C. Y. Lee, “Removing rain and snow in a single image using saturation and visibility features,” *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2014.
- [20] C. H. Son and X. P. Zhang, “Rain removal via shrinkage of sparse codes and learned rain dictionary,” *IEEE International Conference on Multimedia Expo Workshops*, pp. 1–6, 2016.
- [21] Y. Wang, S. Liu, C. Chen, and B. Zeng, “A Hierarchical Approach for Rain or Snow Removing in a Single Color Image,” *IEEE Transactions on Image Processing*, vol. 26, pp. 3936–3950, 2017.
- [22] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng, “Joint Bi-layer Optimization for Single-image Rain Streak Removal,” *IEEE International Conference on Computer Vision (ICCV)*, pp. 2526–2534, 2017.
- [23] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, “Should We Encode Rain Streaks in Video as Deterministic or Stochastic?,” *IEEE International Conference on Computer Vision (ICCV)*, pp. 2516–2525, 2017.
- [24] M. Bouali and S. Ladjal, “Toward optimal destriping of MODIS data using a unidirectional variational model,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, pp. 2924–2935, 2011.
- [25] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep Joint Rain Detection and Removal from a Single Image,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [26] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [27] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal,” *IEEE Transactions on Image Processing*, vol. 26, pp. 2944–2956, 2017.
- [28] H. Zhang, V. Sindagi, and V. M. Patel, “Image De-raining Using a Conditional Generative Adversarial Network,” *arXiv:1701.05957*, 2017.
- [29] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, pp. 259–268, 1992.
- [30] L. J. Deng, H. Guo, and T. Z. Huang, “A fast image recovery algorithm based on splitting deblurring and denoising,” *Journal of Computational and Applied Mathematics*, vol. 287, pp. 88–97, 2015.
- [31] J. Liu, I. W. Selesnick, X. G. Lv, and P. Y. Chen, “Image restoration using total variation with overlapping group sparsity,” *Information Sciences*, vol. 295, pp. 232–246, 2015.
- [32] X. L. Zhao, F. Wang, and M. Ng, “A new convex optimization model for multiplicative noise and blur removal,” *SIAM Journal on Imaging Sciences*, vol. 7, pp. 456–475, 2014.
- [33] L. J. Deng, W. Guo, and T. Z. Huang, “Single image super-resolution via an iterative reproducing kernel Hilbert space method,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 2001–2014, 2016.
- [34] L. J. Deng, W. Guo, and T. Z. Huang, “Single image super-resolution by approximate Heaviside functions,” *Information Sciences*, vol. 348, pp. 107–123, 2016.
- [35] X. Liu, X. Lu, H. Shen, Q. Yuan, Y. Jiao, and L. Zhang, “Stripe Noise Separation and Removal in Remote Sensing Images by Consideration of the Global Sparsity and Local Variational Properties,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 3049–3060, 2016.
- [36] T. Y. Ji, T. Z. Huang, X. L. Zhao, T. H. Ma, and L. J. Deng, “A non-convex tensor rank approximation for tensor completion,” *Applied Mathematical Modelling*, vol. 48, pp. 410–422, 2017.
- [37] D. L. Donoho, “De-noising by soft-thresholding,” *IEEE Transactions on Information Theory*, vol. 41, pp. 613–627, 1995.
- [38] X. Zhao, F. Wang, T. Z. Huang, M. Ng, and R. Plemmons, “Deblurring and sparse unmixing for hyperspectral images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, pp. 4045–4058, 2013.
- [39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.
- [40] K. Garg and S. K. Nayar, “Photorealistic rendering of rain streaks,” *ACM Transactions on Graphics*, vol. 25, pp. 996–1002, 2006.
- [41] D. Berman, T. Treibitz, and S. Avidan, “Non-Local Image Dehazing,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.